Drifts Slips and Misses

Input Accuracy for Touch Surfaces

Chat Wacharamanotham

Drifts, Slips, and Misses
Input Accuracy for Touch Surfaces

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Chatchavan Wacharamanotham, M.Sc.

aus Bangkok, Thailand

Berichter: Prof. Dr. Jan Borchers
Pierre Dracigevic, Ph.D.

Tag der mündlichen Prüfung: 5 Februar, 2016

Diese Dissertation ist auf den Internetseiten der Hochschulbibliothek online verfügbar.

Contents

List of Figures

List of Tables

Abstract

Touch screens allow users to interact with virtual objects directly below their fingertips. The proximity of input and output blurs the line between the physical and the virtual world, allowing the interactions to feel natural. However, direct finger input has several limitations. Compared to the tip of a graphics stylus, a fingertip is bigger and softer, making it more likely to occlude the screen and generate ambiguous touch signals. Furthermore, touch screens register any contact, making it more likely that they will respond to unintentional input, such as pressing a button when a finger just brushes pass it. These two problems exemplify issues in two types of input accuracy: space accuracy ("where it is being touched") and state accuracy ("whether it is being touched").

In this thesis, we investigate space and state accuracy in four usage scenarios.

First, we focused on users with hand tremors, whose involuntary finger oscillation causes them to miss targets and creates spurious touches and releases. To improve touch screen accessibility, we investigated how tremors influence touch input. Then, we designed and evaluated an alternative interaction technique that leverages the tremor movement characteristics for more accurate input.

Second, we addressed a state accuracy problem in indirect multi-touch systems, in which a horizontal multi-touch screen is used to control cursors on a vertical display for ergonomic usage. We operationalized measures for state slips and compared four techniques for controlling the state of cursors.

Third, we augmented touch screens with near-surface interaction by sensing fingers hovering in a thin layer above the screen surface. We determined appropriate layer thickness to minimize the likelihood that the fingers will slip out of the layer.

Finally, we tackled the problem where touch contacts drift away from buttons when users employ touch screens without looking at them. Here, we assessed how magnetic forces might substitute for vision by guiding the fingertips towards the button in these scenarios.

While the findings contribute to the body of scientific knowledge in each specific usage scenario, the insights derived from all four scenarios in combination suggest strategies for designing touch interaction techniques to maximize space and state accuracy.

Überblick

Mittels Touchscreens können Nutzer direkt mit virtuellen Objekten unter ihren Fingern interagieren. Da Eingabe und Ausgabe von Daten so eng zusammen liegen, schwindet die Grenze von der physischen zur virtuellen Welt, wodurch sich Interaktionen 'natürlich' anfühlen. Dennoch ist die direkte Eingabe mit dem Finger mit einigen Einschränkungen verbunden: Wenn man die Spitze eines Eingabestifts mit einem Finger vergleicht, ist letzterer grösser und weicher, was die präzise Eingabe auf Touchscreens erschwert und leicht zu unbeabsichtigten Signalen führt. Hier zeigen sich Probleme in zwei Arten der Eingabegenauigkeit: Treffgenauigkeit („wo wurde die Eingabe gemacht") und Zustandsgenauigkeit („wurde eine Eingabe gemacht").

In dieser Arbeit untersuchen wir Treffgenauigkeit und Zustandsgenauigkeit in vier Nutzungsszenarios:

Zunächst stellen wir fest, dass Nutzer, die unter einem Handtremor leiden, durch unbeabsichtigtes Zittern Ziele verfehlen und ungewollte Eingaben verursachen. Um die Treffgenauigkeit zu verbessern, haben wir untersucht wie Muskelzittern die Eingabe auf Touchscreens beeinflusst, eine Alternative zur Eingabe entwickelt, die o. g. Schwierigkeiten umgeht, und diese ausgewertet.

Des Weiteren, sprechen wir ein ergonomisches Problem mit der Zustandsgenauigkeit bei multi-touch Systemen an, bei denen horizontale Bildschirme zur Eingabe genutzt werden, während die Ausgabe auf vertikalen Bildschirmen erfolgt. Wir haben Massnahmen gegen ungewollte Zustandsänderungen entwickelt und vier Methoden zur besseren Kontrolle des Cursors verglichen.

Ferner, haben wir eine Verbesserung an Touchscreens mit oberflächennahen Interaktionsfunktion vorgenommen, wobei mittels einer zusätzlichen, hauchdünnen Schicht auf dem Bildschirm Fingeraktivitäten besser wahrgenommen werden können. Hierzu haben wir eine geeignete Stärke festgelegt, um ein Abrutschen des Fingers bei der Eingabe zu minimieren.

Zum Schluss sprechen wir ein weiteres Problem an, das sich mit dem Abdriften des Kontaktpunktes beim Drücken von Schaltern befasst, wenn der Nutzer Touchscreens blind benutzt. Hierbei bewerten wir magnetische Hilfsmittel, die den Nutzer bei der blinden Anwendung führen.

Haben wir uns während dieser Arbeit zum Beitrag des Wissensbestandes mit jedem Szenario individuell befasst, dienen alle vier zusammen als Anregung für die weitere Entwicklung von Touchscreens, um Treff- und Zustandsgenauigkeit zu maximieren.

Acknowledgements

I owe a debt of gratitude to my advisor, Prof. Jan Borchers. His tolerance to my rebellious attitude is impeccable. I am grateful for the intellectual freedom and impressive working environment that he curated. I would like to express my gratitude to Dr. Pierre Dragicevic for insightful conversations that inspired my work in understanding statistics and for being an evaluator for this thesis.

My doctoral study and research were, in part, sponsored by the B-IT foundation. I am fortunate to have collaborated with and learned from Alexander Mertens and Sukeshini Grandhi.

I owe my intellectual growth to the folks at the Media Computing Group, especially the following: Thorsten Karrer and Malte Weiß, I learned a lot from both of you, probably more than you are aware of. Thanks for your advice and all the kicks in the behind that made me strive to be better. Simon Völker and Florian Heller: It's a joy to work with you guys. Thanks for being such good friends. This time, no more late-night CHI sprints for real! Leonhard Lichtschlag and Moritz Wittenhagen were always there when I needed practical advice. A major part of my learning experience came from giving advice to junior colleagues. Christian Leonhard Corsten, Nur Al-huda Hamdan, and Krishna Subramanian, thank you for bearing with this old fool throughout his guidances and misguidances. Special thanks to Nur and Simon who gave feedback on several drafts of this thesis. Many of our works would have been impossible without our unsung hero, Clarissa de Gavarelli and her bureaucratic magic.

I am fortunate to have worked with many extraordinary students: Marty Pye, Philipp Wacker, and Kashyap Todi, it was my privilege to work with you and to see you mature intellectually. I hope that my contribution bettered your life, at least in a small way. I am also grateful of helps in teaching and software development from Can Liu, Zhao He, Jan Schnitzler, Aaron Krämer, and many other wonderful i10 student assistants.

Thanks to Nichamon Naksinehaboon and Ignacio Avellino for moral support in tough times. This thesis would not have been possible without Roland Engelmann, who helped in proofreading and provided moral support, delicious food, and laughters.

Lastly, Mom and Dad, thank you for giving me freedom and encouraging me to pursue my aspiration: ขอบคุณป๊ากับม๊าที่ให้อิสระและสนับสนุนให้ตี๋ได้ไปไกลตามที่ตี๋ต้องการ

Chapter 1

Introduction

> *"The Golden Touch !" exclaimed he. "You certainly deserve credit, friend Midas, for striking out so brilliant a conception. But are you quite sure that this will satisfy you?"*
>
> *"How could it fail?" said Midas.*
>
> —Hawthorne and Crane [1892]

Touch screens are prevalent and versatile.

A touch screen registers the location of fingers pressing on its surface for computer input, and visual output is displayed on the same surface [Buxton et al., 1985, Hinckley, 2012]. It is expected that more than two billion touch screen units will be shipped in 2016 [Hsieh, 2015]. They are used in a variety of sizes, from small wearables, mobile phones, and laptop computers to large interactive public displays. This versatility is partially due to the fact that many interaction techniques on touch screens allow users to behave and feel natural [Wigdor and Wixon, 2011]. Touch screens have several properties that can be exploited for natural interaction: the proximity between the fingertip input and the visual output, the high degrees of freedom from multiple touches, and the spatial relationships of the virtual world and the screen frame [Wigdor and Wixon, 2011].

Some characteristics of touch screens benefit interaction design

Other characteristics make interaction design challenging.

Meanwhile, other characteristics of touch screens make interaction design challenging. The flat surface of touch screens provides limited tactile cues. It is hard for users to ascertain the location that they touched without looking at the screen or receiving other forms of feedback [Findlater et al., 2011]. In other situations, when users look at their touch, the point registered by the screen may still deviate from where they intended. Lastly, unintended contacts from the fingers or other body parts can be registered as input [Hinckley et al., 2010]. These problems reduce the accuracy of touch input. But what do we mean by accuracy?

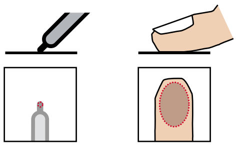
Definition: linebreakemphaccuracy

Accuracy is a quality indicating "the states in which the system or the user makes errors" [Lazar et al., 2010]. High accuracy is one of the main goals of user interface design [Shneiderman et al., 2009]. The accuracy of user interfaces is commonly evaluated with other aspects such as speed or subjective satisfaction [Lazar et al., 2010]. There is usually a compromise between accuracy and speed: achieving tasks at higher speed usually yields lower accuracy [Lazar et al., 2010]. To evaluate a user interface, specific aspects of accuracy are quantified in *accuracy measures*. The generic accuracy measure is the error rate, which is the ratio between the number of successful attempts to the total number of attempts. However, it is generally more insightful to consider the aspect of accuracy that is specific to the task or user interface. For input on touch screens, there are two well-known accuracy problems: the fat finger problem and the Midas touch problem.

Definition: linebreakemphaccuracy measures

1.1 Two Accuracy Problems on Touch Screens

The fat finger problem: fingertip softness and occlusion causes the mismatch of registered vs. intended contact point



Physically, each touch on a touch screen creates a two-dimensional *contact area*. However, touch screens generally represent each contact area as a *contact point* for input processing. But which point should represent the contact area? On graphics tablets, the solid tip of a stylus yields a small contact area. Using any of the candidate points results in an infinitesimal deviation from the user's expectation. On touch screens, however, the finger touch has a much larger contact area because human skin is compressible [Hinckley et al., 2010]. When an incorrect point is used to represent a contact area, it is more noticeable. Moreover, the finger itself occludes the user from seeing the exact position, resulting in additional inaccuracy [Vogel and Baudisch, 2007]. The inaccuracy in touch contact caused by the fat contact area and the occlusion of the fingertip is dubbed *the fat finger problem*.

The Midas touch problem: incorrectly registering touch

Immediate activation may result in accidental input when fingers brush or rest on the screen; this is known as *the Midas touch problem* [Hinckley et al., 2010]. In Greek mythology, King Midas was blessed with the ability to turn what he touched into gold. His initial satisfaction with the wealth he brought with his touches, he was devastated after accidentally turned his daughter into gold. The first use of the term "Midas touch" in HCI referred to a problem in gaze-based input by Jacob [1991]. He discussed a naïve interaction technique that activated any point that the user looked at. Since the eye movements are caused by a mixture of voluntary and involuntary movements [Zhai et al., 1999], this interaction technique is prone to activating locations that the user does not intend. Hinckley et al. [2010], later, pointed out that this problem applies to touch screens as well since every brushing on the screen (e.g., by users' palms) activates the screen.

Midas touch (Image adopted from [Hawthorne and Crane, 1892])

1.2 Input Accuracy on Touch Screens

The fat finger problem and the Midas touch problem are examples of touch input accuracy problems. Fundamentally, from the input capability, a touch screen "can sense that it is being touched, and where it is being touched" [Buxton et al., 1985]. Based on this definition, we derive two types of touch input accuracy: the accuracy of the input state ("whether it is being touched") and the accuracy of the input position ("where it is being touched"). Below, we explain how the fat finger problem and the Midas touch problem relate to these two types. For common understanding, we use the basic terms in Box 1.

Two types of the accuracy problems are derived from Buxton's definition of touch screens.

Box 1: Basic Terms (An excerpt from [Hinckley, 2012])

Input device An input device is a transducer that senses physical properties of people, places, or things.

Conceptual model A conceptual model is a coherent model that users visualize about the function of a system—what the system is, how it works, and how it will respond to users' input. Thus users can determine what input they need to give to the system to achieve desired result.

Interaction technique An interaction technique is the fusion of input and output, consisting of all hardware and software elements, that provides a way for the user to accomplish a task for a particular conceptual model.

User interface A user interface is the representation of a system—the summation of all its input devices, conceptual models, and interaction techniques—with which the user interacts.

1.2.1 Space Accuracy

Margin of error in touch contact depends on the task.

The fat finger problem, too, depends on the task.

As described above, touch screens may register contact points in a position that deviates from where the user intended. However, how the user experiences a touch system may not necessarily be altered if the deviation is sufficiently small. In other words, there is a *margin of error* that is tolerable for positional deviation for each task. For example, in a handwriting application, the position of each stroke in a character influences the character's legibility as a whole. Such an application would demand a smaller margin of error than a calculator application, which typically has only twenty buttons on the entire screen. On the same touch screen, users are more likely to experience the fat finger problem in the handwriting application than in the calculator application. With the concept of acceptable margin of error, we define *space accuracy* as follow.

Definition: *Space accuracy*

SPACE ACCURACY:
Space accuracy is the quality of the user interface that allows the user to specify the position of input as intended within a desirable margin of error.

Another example: drifts in eyes-free input

Besides the fat finger problem, there are other situations where the space accuracy is reduced. For example, in a touch D-pad controller, there are several buttons that the user repeatedly taps with the thumbs (Figure 1.1). In eyes-free input, which is when the visual attention is elsewhere on the screen, the thumb may drift away from the buttons because of the absence of tactile cues.

Space accuracy is influenced by input device, conceptual model, interaction technique

These space accuracy problems stem from different parts of the user interface: the input device, the conceptual model, or the interaction technique. As discussed above, the fat finger problem is inherent to the input device and the conceptual model of touch. It is possible to alleviate the fat finger problem by improving the contact point recognition algorithm to match users' conceptual model of touch [Holz and Baudisch, 2011]. For eyes-free drifting and motor disability, however, new interaction techniques are needed. In the D-pad controller example, drifting is irrelevant if the controller widget is changed to a virtual joystick (Figure 1.2): the user slides the thumb in the desired direction instead of tapping on directional buttons without lifting the thumb from the screen.

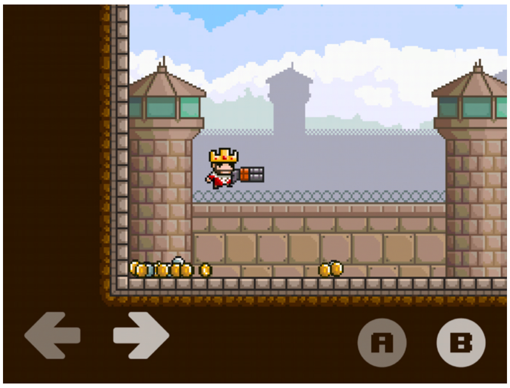


Figure 1.1: Directional controller (D-pad) in a game. The user's finger has to home onto the directional buttons. Sometimes, the finger drifts away from the buttons due to the absence of tactile cues, reducing space accuracy. (Image from Random Heroes, Ravenous Games Inc.)

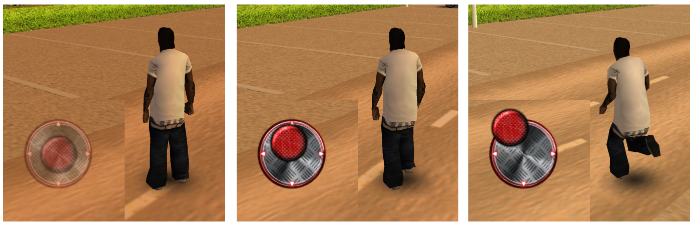


Figure 1.2: Touch screen virtual analog joystick can be controlled without lifting the thumb from the screen, hence, suffer less from space inaccuracy. (Image from Urban Crimes, Gameloft)

1.2.2 State Accuracy

In the Midas touch problem, the screen registers the contact points that were produced unintentionally. The inaccuracy occurs not in the *space* of the screen but in the *state* of each touch. In general, each transducer on the touch screen is capable of distinguishing two states: whether it is being touched, or there is no contact in range. The presence or absence of physical contact changes between the two states. To formulate this input capability, Buxton [1990] used a finite state machine model as described in Box 2.

The Midas touch problem concerns the accuracy of registering input state.

Box 2: The State Model of Graphical Input

Buxton [1990] modelled each input device using a finite state machine with three possible types of states:

State 0: when user movements are not interpreted as action (out of range)
State 1: when user movements control cursors (tracking)
State 2: when user movements control virtual objects (dragging)

Interaction techniques are needed to switch among these input states. Such techniques can be implemented in hardware or software, or they can be simple physical actions taken by the user. Reliable state transitions are essential for designing interaction techniques [Hinckley, 2012].

For example, a graphics tablet with a stylus is a device that can register all three input states (Figure 1.3, left). When the stylus is far from the tablet, it is out of the range of the sensors on the tablet surface. Any movements of the stylus do not influence the system (state 0). When the stylus is in range of the tablet sensor, moving the stylus causes the cursor to move on the screen (state 1). Depressing the tip of the stylus activates virtual objects (e.g., depressing a button) below the cursor (state 2).

A computer mouse registers state 1 by default and state 2 when a button is depressed (Figure 1.3, middle). (A mouse with multiple buttons can have multiple instances of state 2.) Similar to the case of graphics tablet, state 1 is used to track cursors on the screen to preview the location, the action (e.g., showing tooltip), or properties of the action (e.g., the brush size in graphics software). When switching between state 1 and 2, users receive distinct tactile feedback from the mouse buttons.

In contrast, a majority of touch screens only differentiate between state 0 and state 2, on the presence and absence of touch contact, respectively [Hinckley, 2012] (Figure 1.3, right). Compared with mouse and stylus tablets, the absence of state 1 makes it hard for users to specify the precise location on the screen. For example, in touch painting software, it is hard to draw small details without resorting to zooming interfaces. The lack of distinct tactile feedback also makes touch screens prone to accidentally switching input states. For example, a virtual button may be accidentally activated when a finger brushes over it.

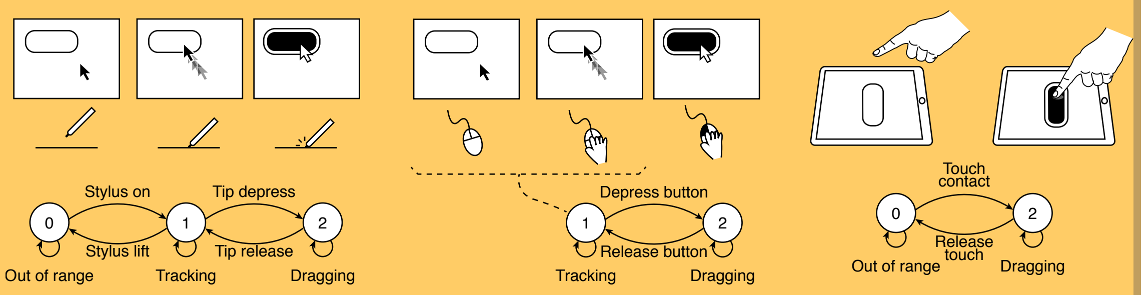


Figure 1.3: Input states of graphics tablet with a stylus, mouse, and touch screen

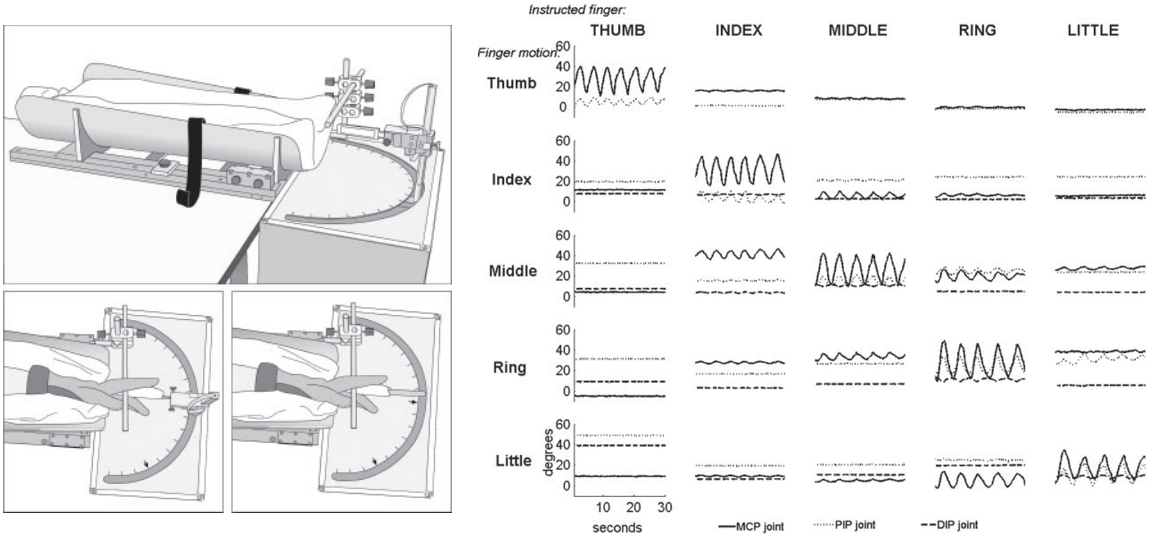


Figure 1.4: An experiment on mechanical coupling among fingers. Left: a schematic of the device used to measure the coupling. Right: A trace of finger movement. The presence of movement amplitude apart from the cells on the main diagonal indicates coupling. (Image adopted from Lang and Schieber [2004])

According to the input state model, the Midas touch problem is a mismatch between the input state in the user's conceptual model (dwelling in state 0) and the state that was registered (transitioned to state 1). The Midas touch problem is the lack of state accuracy.

STATE ACCURACY:
State accuracy is the quality of the user interface that allows the user to transition among input states as intended.

Definition:
State accuracy

Problems in state accuracy may result from anatomical limitation. In particular, fingers on the same hand, especially the index, middle, and ring fingers, are mechanically coupled [Lang and Schieber, 2004] , as shown in Figure 1.4. In the history of music, this coupling is known as the *Schumann slip* [Jaynes, 1996] , named for a famous pianist who tried to force his ring finger to be independent via a mechanical contraption and ended up ruining his right ring finger.

Anatomical limitations may threaten state accuracy.

Space accuracy and state accuracy may be threatened by the same source. For example, a motor disability such as hand tremor, for example, causes involuntary movements in the hands and fingers. These movements may deviate the user's touch from the intended position

Both space accuracy and state accuracy may be threatened by the same phenomenon.

(reducing space accuracy) as well as oscillate the fingertip on and off the screen (reducing state accuracy).

These low-level space and state accuracy influences higher-level accuracy

Beyond these two types of touch input accuracy are other task-specific accuracies such as text entry, pointing, or strokes. The state accuracy and space accuracy can influence many of these high-level accuracy measures. For example, the text entry error rate can be lowered if the users are unable to position a finger on a button (low space accuracy) or unable to prevent a finger from accidentally tapping the button (low state accuracy). In the following chapters, we use some of these high-level measures to reflect the state and the space accuracy.

1.3 Usage Scenarios and Contributions

Four usage scenarios represent four components of interactive systems.

The problems of space accuracy, state accuracy, or both can stem from different parts of the user interface. According to Dix et al. [2003], interactive systems comprise four major components: the user, system, input, and output (Figure 1.5). In this thesis, we take a look at four usage scenarios; each represents one component in this framework.

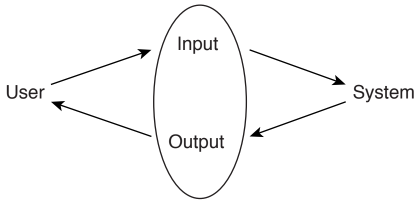


Figure 1.5: Dix et al.'s framework of interactive systems. (Diagram created by the author based on Dix et al. [2003])

Chapter 2 concerns state and space inaccuracy caused by hand tremor.

In the first scenario, we show how a motor disability threatens to reduce space and state accuracy. In particular, we focus on *tremors*, which are involuntary oscillating movements. When using a touch screen, users with hand tremor make spurious and deviating contact points, resulting in input inaccuracy. In Chapter 2, we quantify how tremor symptoms influence touch screen usage. From this knowledge, we designed and evaluated *Swabbing*, an interaction technique that improves the state and space accuracy for this user group.

The second scenario concerns the accuracy issue in the *indirect multi-touch form factor*. Such systems combine two touch screens: a horizontal touch screen receives multi-touch input, and a vertical screen provides visual output. This combination allows the user to sit upright during usage, which is considered ergonomic [Marras, 2012]. However, the separation of the input and output surface makes it hard to land the finger accurately on intended targets [Schmidt et al., 2009], resulting in space inaccuracy. Adding the cursor tracking state (state 1) is one solution for the space inaccuracy. However, since touch screen hardware inherently senses state 0 and 1, an interaction technique is needed. In Chapter 3, we operationalize state inaccuracy in the indirect touch system and describe an evaluation of four candidate interaction techniques.

Chapter 3 presents state inaccuracy in indirect multi-touch form factor.

In the third scenario, we consider an expansion of touch screen sensing capability that allows tracking of the position of the fingertip near the surface. Near-surface input can be treated differently from on-surface touches and thus increase the possible interaction vocabulary of the user interface. However, this extension creates a continuum of space on-surface, near-surface, and out-of-range space, without clear visible or tactile boundaries. As a result, state inaccuracy can occur when the finger drifts from one area to another. To minimize state inaccuracy an appropriate layer thickness is required. We explored these parameters through user studies described in Chapter 4.

Chapter 4 describes state inaccuracy in near-surface input.

Besides expanding possible interaction designs, we consider how near-surface tracking can alleviate space inaccuracy in typical touch screen input. In the fourth scenario, we envisioned a system that—in addition to near-surface tracking—provides haptic cues to the finger before touching the surface. Chapter 5 describes *FingerFlux*, a prototype of such a system that creates haptic cues through electromagnetic vibration. We evaluated how such a system prevents the finger from drifting from the target on the screen.

Chapter 5 concerns alleviation of space inaccuracy using near-surface haptic cue.

In summary, this thesis makes the following contributions:

1. To improve accessibility of touch screens for users with hand tremor, we designed and evaluated the Swabbing input technique which improves both space accuracy and state accuracy.

2. To enable indirect multi-touch form factors, we operationalized measures for state accuracy specific to this form factor, and empirically evaluated four state-switching techniques.

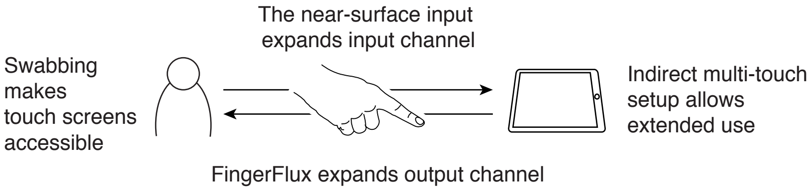

The near-surface input expands input channel
Swabbing makes touch screens accessible
Indirect multi-touch setup allows extended use
FingerFlux expands output channel

Figure 1.6: Contribution summary according to Dix et al.'s framework

3. To ensure state accuracy in the input techniques that use signals from finger hovering near-surface, we empirically determined a thickness for the near-surface input layer.
4. To improve space accuracy, we developed the FingerFlux, a system that guides the fingertip on- and near-surface with electromagnetic actuation.

These contributions are summarized in Figure 1.6, according to Dix et al.'s framework.

1.4 What's New?

Works described in this thesis are a subset of the findings that were previously published. These publications ranged from co-authored peer-reviewed articles, to several bachelor's and master's theses that were supervised by the author.

1. Organization under the concepts of space and state accuracy.

This thesis differs from the previous publications in two major aspects. The first aspect is the organization of these works under the umbrella of state accuracy and space accuracy. Conceptually, the two types of accuracy can be viewed as two dimensions of the input accuracy problem space. Each of the four chapters provides a data point in this problem space.

2. Data analysis with the statistical estimation paradigm

Secondly, this manuscript provides a revised data analysis. In our previous publications, we analyzed data collected from user studies using null-hypothesis significance testing (NHST) paradigm, which was the best process to my knowledge at the time of each publication. However, NHST has several flaws. In particular, it biases the interpretation of results towards dichotomous thinking [Dragicevic, 2015].

In this manuscript, the author re-analyzed data using statistical estimation paradigm, which emphasizes on estimating effect sizes and their uncertainty. As a result, the findings and discussion in this thesis may differ from previous publications. When a discrepancy occurs, the differences are discussed.

For additional materials such as R code for data analysis and errata, see http://chat.info/driftsslipsmisses.

> *"...the scientists who embraces a new paradigm is like the man wearing inverting lenses. Confronting the same constellation of objects as before and knowing that he does so, he nevertheless finds them transformed through and through in many of their details."* —Kuhn [2012]

Chapter 2

Improving Touch Screen Accessibility for Users with Hand Tremors

Hand tremor degrades space and state accuracy.

Tremors in the upper limb influence space and state accuracy of touch input. The oscillation parallel to the screen plane may cause the contact point to deviate from the intended position. The oscillation orthogonal to the screen plane may cause the finger to repeatedly bounce on and off the screen.

Current assistive features are inadequate

Despite these difficulties, touch screens are still appealing for people with hand tremor. Anthony et al. [2013] surveyed YouTube videos of touch screen users with hand tremor. The results indicate that assistive features implemented in current touch screen operating systems

Publications: The work in this chapter is a collaboration with Alexander Mertens. The author is responsible in creating tremor measurement hardware, designed swabbing gesture recognizer, designing the experiments, and analyzing data from the experiments. Part of this work was first published as a short paper at the CHI 2011 conference [Wacharamanotham et al., 2011], then as a poster at DGN 2011 [Mertens et al., 2011], a book chapter [Mertens et al., 2012], and a workshop paper at CHI 2013 [Wacharamanotham et al., 2013]. Early iterations of Swabbing was implemented and evaluated in several theses supervised by the author: Hurtmanns [2011] implemented and evaluated the first version of swabbing and collected data for tremor characterization. Huck [2012] improved the implementation and conducted the study evaluating different opening angles. Kehrig [2013] re-implemented swabbing in multi-touch tablets and dealt with edge cases in the vertical swabbing as well as multi-touch handling. He also implemented the swabbing browser and conducted the longitudinal study.

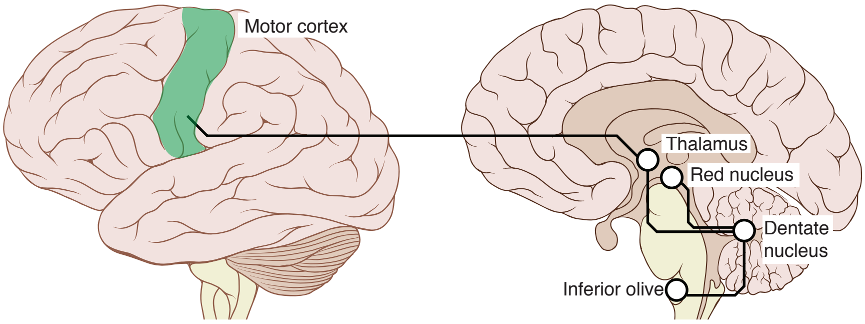


Figure 2.1: Tremor network as identified by [Sharifi et al., 2014][1]. The complex relationship among different parts of the brain still make curing tremor out of reach.

Users modify their touch screen with home-made hardware.

are still inadequate for this user group. However, many of the users came up with ad hoc modification to improve accessibility, such as adding cardboard barriers or putting a plastic bag onto the screen.

In this chapter, we investigate how the space and state accuracy is influenced by tremor, and we propose and evaluate an interaction technique to improve the accuracy. But what exactly is a tremor?

2.1 Hand Tremor

Definition: linebreakemphtremor

The Movement Disorder Society defines *tremor* as "rhythmical, involuntary oscillatory movement of a body part" [Deuschl et al., 1998]. They distinguish between temporary physiologic tremors (caused by, e.g., sore muscles or alcohol consumption), and chronic pathologic tremors, which are found in people with neurological disorders, such as multiple sclerosis, strokes, or brain injuries. These disorders damage brain regions involved in motor control, especially motor cortex, thalamus, and dentate nucleus [Sharifi et al., 2014] (Figure 2.1).

Causes of pathologic tremors

Abnormal neurological firings in these regions cause muscles to involuntarily rapidly contract and relax, resulting in a tremor. Because of the complexity of the brain regions involved, tremor oscillation in each patient is variable by numerous factors such as mood, movement conditions, and medication.

[1] Image created by the author based on illustrations by Patrick J. Lynch, medical illustrator; C. Carl Jaffe, MD, cardiologist (Creative Commons Attribution)

2.1.1 Treatments

Medical science is still grappling with diseases that cause tremors. However, in severe cases, the tremor can be suppressed by implanting electrodes that generate high frequency impulses in thalamus to block abnormal signals from activating muscles. This procedure is called *deep brain stimulation* (DBS). Implanting a battery-powered DBS typically requires three to six hours of awake brain surgery. When the stimulation device is turned on, there are possible side effects such as speech difficulties, involuntary facial contraction, and confusion Beric et al. [2001]. When the device is turned off the tremor resurfaces.

Deep brain stimulation is a temporary solution with several drawbacks.

Recently, Elias et al. [2013] useed MRI-guided ultrasound to stimulate thalamus. The operation is done with the patient's skull intact, and the effect lasts for several years. Nevertheless, this technology is still nascent. As of 2013, only 15 patients were treated, and there are several side effects such as motor and speech problems.

A less invasive ultrasound treatment is still under development.

2.1.2 Classification

There are many classification schemes for pathologic tremors [Deuschl et al., 1998]. They can be classified by syndrome such as *essential tremor* (which is the most common type of tremor) and *parkinsonian tremor* (which is associated to Parkinson's disease). The syndromic classification is useful for choosing the right medical treatment.

Two common syndromes: essential tremor and parkinsonian tremor.

However, a patient diagnosed with one syndrome may exhibit the tremor in different movement conditions. The phenomenological classification concerns the following two conditions that may activate tremors: *Rest tremor* occurs when the affected body part is idle and supported. *Action tremor* occurs with voluntary muscle contraction. One subtype of the action tremor is *intention tremor* that occurs in goal-directed movement such as aiming to press a button on a touch screen. In this thesis, we focus on users with intention tremor.

Intention tremor occurs in goal-directed movements.

In a door-to-door survey conducted in a province in Turkey, Dogu et al. [2003] estimated that essential tremor affects 4% of the population older than 40 years. Louis et al. [2009] found that 44.3% of people with essential tremor have the intention tremor. Aggregating these results together, we can estimate that intention tremor is present in around 2% of the world population older than 40 years.

Intention tremor affects 2% of people aged 40 years or older.

Tremors can also be classified by the oscillation frequency.

Lastly, tremors can be classified by oscillation frequency into low (<4 Hz), medium (4–7 Hz), and high (>7 Hz). The relationship between syndromic, phenomenological, and frequency classification is shown in (Figure 2.2).

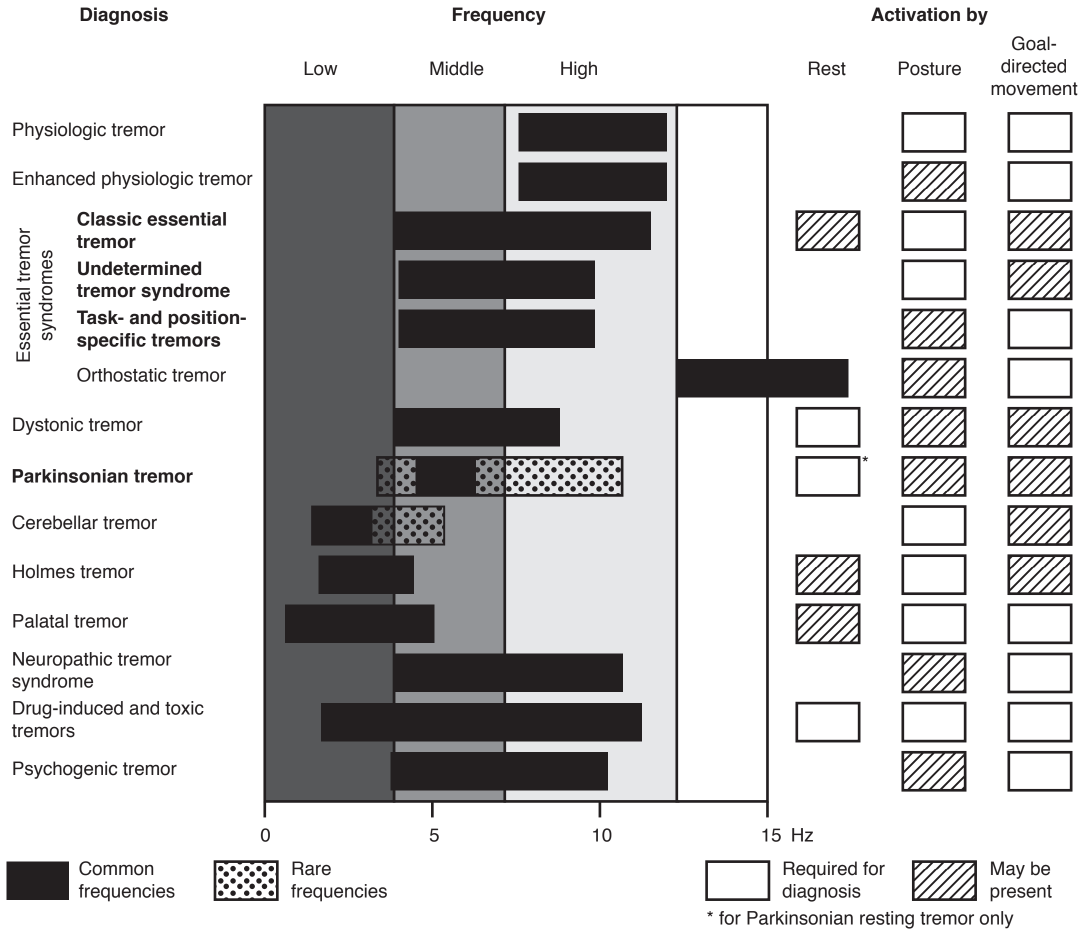


Figure 2.2: Tremor syndrome, oscillation frequency, and activation conditions. Syndromes found in our participants are emphasized in bold. (Image created by the author based on [Deuschl et al., 1998].)

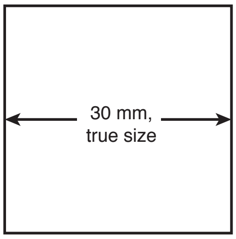
2.2 Related Work

We first describe investigations of tremor effect on touch screen usage and guidelines. Then, we describe alternative interaction techniques.

2.2.1 Touch Screen Tapping for Users with Hand Tremor

Motor disabilities have been found to degrade user experience with touch screens. Previous works investigated tapping, which is a common interaction technique. Chen et al. [2013] compared button sizes (10–30 mm square) for users with and without motor disabilities (12 users with tremor) in a four-digit number entering task. In terms of error rate, their able-bodied participants reached plateau at the button size 20 mm. However, the error rate from the participants with disabilities still improved from the enlargement up to 30 mm. Even in the largest button, the error rate was more than twice larger in the disabled group than in the able-bodied group. Therefore, space accuracy of touch screen tapping is deteriorated because of hand tremors.

Button size for tremor:



State accuracy is also influenced by tremors. Irwin and Sesto [2012] found that users with motor disability generate 1.5 times more taps than able-bodied users in a reciprocal tapping experiment. In a study on mobile phone text entry for elderly users (eight of them have hand tremor), Nicolau and Jorge [2012] found a strong correlation between the strength of hand tremor and the number of times their users unintentionally repeated tapping on the same key.

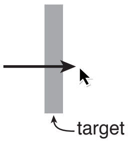
State inaccuracy occurred from repetitive touches

Improving touch screen tapping is challenging. Montague et al. [2014] used a sudoku puzzle application on an iPod Touch (3.5 inch touch screen) to collect tapping input from users with motor disabilities (9 participants, 7 with tremors) in four weeks. These input data were used to train two probabilistic gesture recognizers: *user-specific* and *session-specific*. They were compared to the default iPod Touch tap recognizer, yielded 85% cross-validation accuracy. The results show that, to accommodate motor disabilities, training gesture recognizers per user is not enough (79.7% accuracy); they need to be trained per usage session (95.1%). However, this training would be impractical in real use (200 training taps, in Montague et al. [2014]'s work).

Tap gesture recognizer needs to be session-specific, but the required training is impractical for real use.

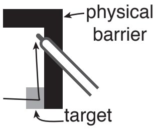
2.2.2 Alternative Interaction Techniques on Touch Screens

Move the cursor crossing over a target to select.



Instead of improving tapping, several works proposed using alternative input methods, one of such is *goal crossing*: swiping the finger (or using a computer mouse to move the cursor) across a target to select it. Wobbrock and Gajos [2008] evaluated goal crossing with trackball and mouse input. They found that goal crossing improves stability of cursor motion. However, Nicolau et al. [2014] found that goal crossing on touch screens yielded similar error rate as tapping in users with tetraplegic. To date, there was no study on touch screen goal crossing specifically in users with hand tremors. We speculate that tremors may cause sporadic lifts when the finger is about to cross the targets, resulting in a miss.

Physical barriers are used to stabilize stylus motion.



Barrier Pointing used physical edges of the screen to help stabilizing stylus on touch displays [Froehlich et al., 2007]. In this technique, selection targets are placed near the edge of the screen. To select a target, users move the stylus to the edge next to the target and perform a gesture, such as, swiping the stylus to a corner. When applying this technique to finger touch input, target occlusion could be problematic because fingers are much larger than the tip of styluses. Nevertheless, Anthony et al. [2013] found that some users attached home-made physical barriers to their touch screens in order to use them with specific applications.

Informal observation: sliding on the screen stabilizes the finger

In an informal observation Mertens et al. [2010] found that users with hand tremor can use the surface of the screen to stabilize their finger while making swiping gesture across the screen without breaking. They speculate two causes of the increased stabilization: (1) The friction between the finger pad and the screen surface limits the oscillation. (2) The lack of breaking allows the latter part of the swiping movement to be less intentional, hence reducing *intention* tremor symptom. They proposed TRABING design pattern that employs directional swiping to select targets aligned on the edge of the screen. In a preliminary Wizard-of-Oz study, they found that the technique was easy to learn and the participants had positive attitude towards the technique. In this chapter, we further develop and evaluate this interaction design concept.

2.3 Characterizing Tremors during Touch Input

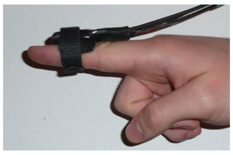
The first principle of ability-based design stated that "designers will focus on ability not *dis*-ability, striving to leverage all that users can do." [Wobbrock et al., 2011]. Hence, to design a touch screen interaction technique for users with hand tremor, we began by investigating how tremors influence touch screen usage. In particular, we investigated how tremor symptoms manifest during tapping and swiping.

We designed the interaction technique based on what users *can* do.

2.3.1 Apparatus

Previous works that quantify tremor in stationary or goal-directed movements (e.g., [Roels et al., 1983, Giuffrida et al., 2009]) attached accelerometers to users' limbs. Similarly, we attached one GForce3D-33 accelerometer GForce3D-33 accelerometer[2] to the back of the extreme joint of the index finger (dominant hand) with a velcro ring. This ring left the entire fingertip uncovered as shown in Figure 2.3.1. The X-axis and Y-axis of the accelerometer pointed along the width and height of the screen, respectively. The Z-axis of the accelerometer was parallel to the normal vector of the screen plane. The accelerometer was connected with an Arduino, which streamed the acceleration data to a computer (effective sampling frequency 20 Hz). The test was performed on a horizontal touch screen. A static crosshair was shown on the screen. No visual feedback was provided for any actions.

An accelerometer was attached to the index finger with a velcro ring.



2.3.2 Procedure

We asked our users to use their index finger in four conditions on a touch screen: (1) hold the finger still in *midair*, (2) *resting* the finger on the screen, (3) repeatedly *tapping* on the screen, and (4) *swiping* the finger to the left and right, along the width of the screen. The *midair* and the *resting* conditions were used to provide a frame of reference. Since we focused on users with intention tremor, having to hold their fingers still in midair was expected to result in highest oscillation. The resting gesture should yield less oscillation than others. For tapping and swiping, we asked the users to pace each action once per second. Each action was recorded for 10 seconds.

Independent variable: four conditions (midair, resting, tapping, and swiping)

[2] http://infusionsystems.com/catalog/product_info.php/products_id/157

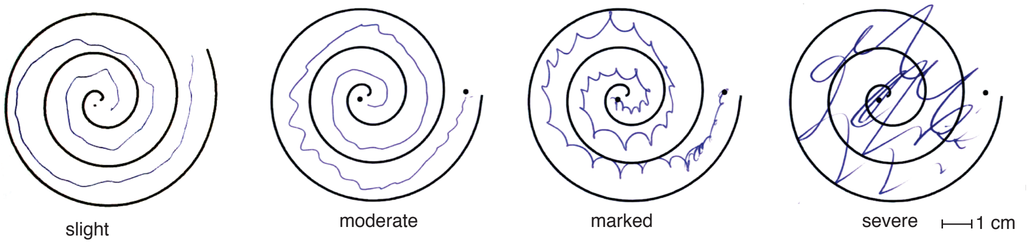


Figure 2.3: Spiralometry test: Participants were asked to draw a spiral inside the track. The distance of the trace from a smoothed path was used to determine the degree of tremor. (Image modified from [Wacharamanotham et al., 2011].)

2.3.3 Participants

Nine participants with various intention tremor strength

We recorded data from nine participants (3 females, average age 74), all with intention tremor. According to the spiralometry test (Figure 2.3), we had one participant with a slight tremor (deviation from spiral <0.5 cm), three moderate (0.5 – 1 cm), two marked (1–2 cm), and three severe (>2 cm).

2.3.4 Data Analysis

Acceleration data is analyzed in the frequency domain.

We trimmed the first and the last second of each action to prevent extreme movements corresponding to the onset and the ending of the task. Then, we converted the accelerometer data from each axis into the frequency domain by applying a Fourier transformation. Figure 2.4 shows the transformation of data from one user.

We used 3–9 Hz frequency range.

Although the center frequency of tremor had outsanding amplitude in Fourier spectrum, a modulation in frequency, amplitude, or both may occur and result in peaks in several frequencies [Gresty and Buckwell, 1990]. Therefore, we used a range of frequencies in our analyis, similar to Veluvolu and Ang [2011][3]. We used a band-pass filter in the 3–9 Hz range to remove the low frequency (1 Hz, from the assigned movement in the task, e.g., tapping) and the high frequency (from the accelerometer noise). The 3–9 Hz range covers the medium tremor frequency [Deuschl et al., 1998].

[3] In our previous publication, we used only the top frequency peaks when they were higher than 0.1 G [Wacharamanotham et al., 2011]. See the Discussion section below.

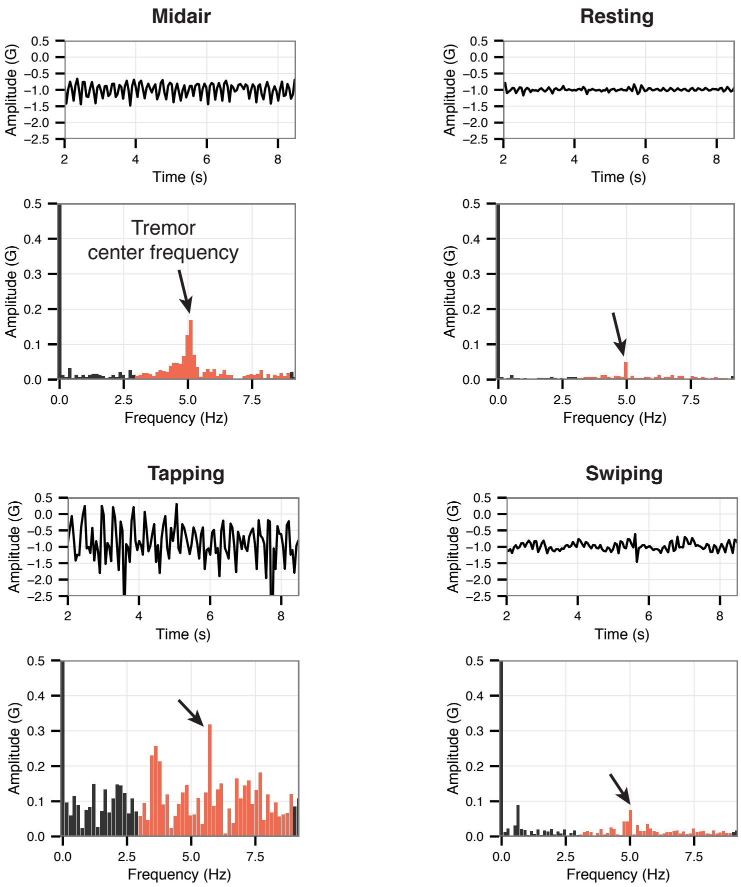


Figure 2.4: Examples of accelerometer data (top panels) and their Fourier transformation (bottom panels). The amplitudes in the tremor frequency range was increased by tapping, but swiping yields relatively low amplitudes. The frequency region we used in the analysis is highlighted. All data shown here were captured from the same participant (moderate tremor).

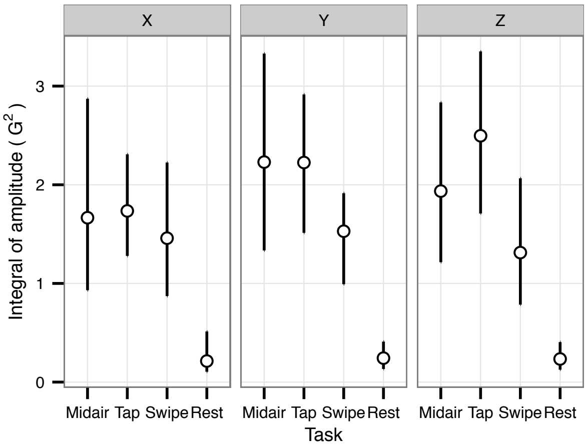
Bootstrapping mean and confidence intervals of amplitude integral over the frequency range

Each trial is represented by the integral of amplitudes in this frequency range. The more severe the tremor, the higher the integral value. In plots below, we show means and their 95% confidence intervalof the integral obtained from all users. To calculate means, we used ordinary non-parametric bootstrapping (10,000 replicates) on the data across all users. CIs were calculated with the bias-corrected and accelerated method (BCa). Some confidence intervals are asymmetric, which reflects the distribution of the data. As for the within-subjects differences, we used the data from tapping action as a baseline.

2.3.5 Results

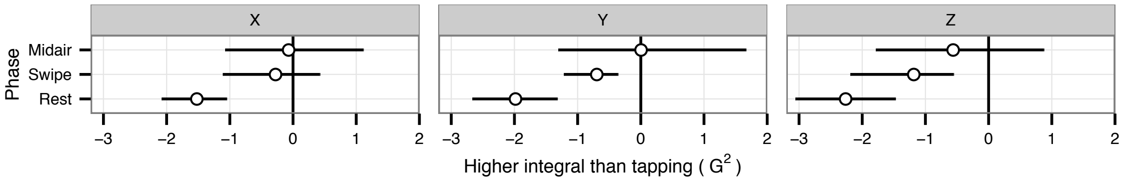
As expected, resting condition yielded the least tremor.

The chart shows the integral of amplitude in each of the axes from all participants. Since all of our users had intention tremor, the integral is relatively low in the resting condition than in others in all three axes.



Tapping yielded tremor similar to holding the finger in midair.

The charts below display the within-subjects differences with the tap condition as a baseline. For the conditions that are likely to show less tremor, CIs are left from zero. The result can be interpreted in context by comparing to the midair condition (in which tremors should be severest) and to the resting condition (in which tremors should be mildest). The CIs of the resting condition are left from zero in all three accelerometer axes, suggesting that the resting condition resulted in less oscillation than the tap condition, as expected for users with intention tremor. Zero is well within the CI of the midair condition in all three axes, indicating that the oscillation in tapping is comparable to holding the finger in midair.



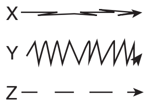
Swiping yielded less tremor in Y- and Z-axis.

The CIs of the swiping condition are left from zero on both the Y- and the Z-axes, suggesting that swiping yielded less oscillation than tapping in these axes. These CIs are shorter on the Y-axis than on the Z-axis, suggesting that the difference is likely to be more consistent on the Y-axis. On the X-axis, zero is well inside the CI, indicating that the oscillation on the X-axis is comparable to tapping.

2.3.6 Discussion

The results support earlier observations in [Mertens et al., 2010]. Compared with tapping, tremor oscillation is less severe in the swiping gesture. Nevertheless, the oscillation still occurred in the direction orthogonal to the screen plane (Z) and the direction of movement (X). Hence, using the swiping gesture directly for input would still suffer from the intermittent finger lifts from the screen and moving back and forth, left). The lower oscillation on the Y-axis indicates that tremors influence less in the orientation of the overall trajectory of movement.

Swiping trajectory seems to be influenced less from tremors.



In the previously published paper [Wacharamanotham et al., 2011], we used only the highest frequency peak that is higher than 0.1 G for analysis. We compared the amplitude of the peaks between the four gestures and compared whether the amplitude of the peak was either lower or higher in different gestures. The results indicated that swiping lowers the peak on the Z-axis in most of the participants. This agrees with the analysis described above. Nevertheless, in the previous analysis, we were unable to capture the oscillation of nearby frequencies that may result from the amplitude or frequency modulation. Besides, the analysis above allows us to quantify the magnitude of within-subjects difference, aggregated across participants.

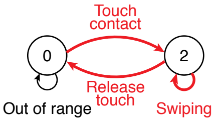
Current analysis captured richer information and agreed with previous analysis.

Since the trajectory of swiping movement is less perturbed by tremors, we designed a selection technique based on the swiping gesture.

2.4 Swabbing Interaction Design

From the related work and the result of the previous section, hand tremors degrade the user experience of tapping. The involuntary oscillation on screen plane (the X- and Y-axis) reduces space accuracy, resulting in missing the target. The oscillation orthogonal to the screen plane (the Z-axis) reduces state accuracy, resulting in spurious tapping. For the swiping gesture, the Z-axis oscillation results in unintentional lifts. In summary, according to the current state model of touch input, hand tremors reduce state and space accuracy at the transitions between state 0 to state 2, and in the dwelling in state 2.

Inaccuracy in touch input for users with hand tremors.



As an alternative input method, we leveraged the reduction of tremor oscillation during swiping on the Y- and the Z-axis to design the swab-

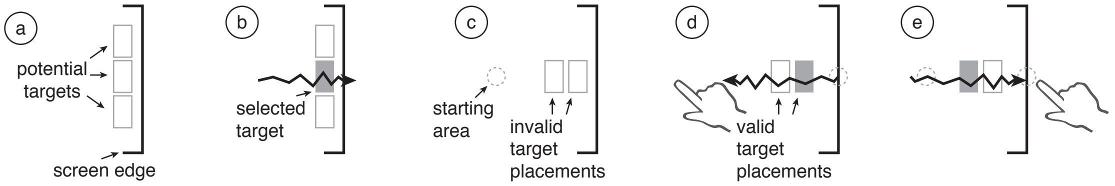


Figure 2.5: Swabbing target placement (a), selection (b), an invalid configuration of target placements (c), and an extension that allows swabbing from the edge of the screen (d, e).

Definition: linebreakemphswabbing

Target placement constraints

bing input technique. In a nutshell, to select a target with the *swabbing* technique, users swipe a finger towards the desired target (Figure 2.5a, b). The direction of the smoothed touch trace determines the selection target. Since trajectory is used to select a target, two targets should not be placed along the same movement trajectory (Figure 2.5c). To maximize the swiping area for better trajectory estimation, we placed targets on the edges of the screen, and we designated the middle of the screen as the starting area. In section 2.4.2, we will describe a modification that allows swabbing from the edge of the screen to allow stacking the targets (Figure 2.5d, e).

Superficially, swabbing is swiping with enhanced thresholding. The key difference, however, lies in the smoothing function in the gesture recognizer and the visual feedback.

2.4.1 Swabbing Gesture Recognizer

Swabbing recognizer for single-touch input

Since swabbing is a single-touch input technique, we first describe the swabbing gesture recognizer by assuming that there is, at maximum, one contact point present on the display at the same time. Later in this section, we describe how to handle multi-touch input.

Thresholds for state transitions are iteratively tested with three users.

The swabbing recognizer is a continuous gesture recognizer with four states as shown in Figure 2.6. Each state transition governs by several spatial or temporal thresholds. We determined these thresholds from an informal testing with three users with slight to moderate hand tremor. Below, we describe each transition and associated thresholds.

Touch events are segmented by 250 ms gaps.

Starting from the *Out-of-range* state, the input of swabbing gesture recognizer is a stream of touch m events $t_1, t_2, t_3, \ldots, t_m$. Each event comprises a screen coordinate $(x_i, y_i; i \in 1, 2, 3, \ldots, m)$, a type (land-on,

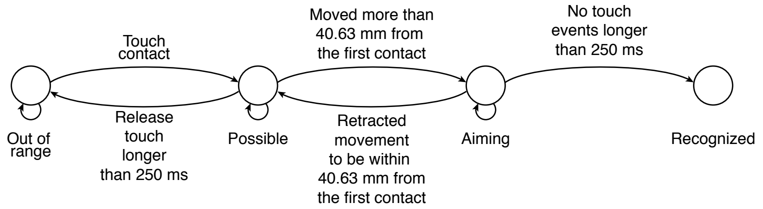
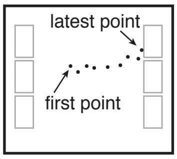

Touch contact
Release touch longer than 250 ms
Moved more than 40.63 mm from the first contact
Retracted movement to be within 40.63 mm from the first contact
No touch events longer than 250 ms
Out of range
Possible
Aiming
Recognized

Figure 2.6: The state model of the swabbing gesture recognizer.

move, or lift-off), and a timestamp ($Time(t_i)$). To tolerate the intermittent lifts (Z-axis oscillation), we ignored the distinction between land-on and lift-off types. Instead, the swabbing recognizer considers successive touch events occurred within 250 ms of each other to be in the same gesture: $Time(t_i) - Time(t_i - 1) < 250$ ms, for all $i \in 2, 3, 4, \ldots, m$. This set of m touch events are considered a possible swabbing gesture. Hence, the recognizer is now in the *Possible* state.



To shift from the *Possible* state to the *Aiming* state, the recognizer considers the 2D Euclidean distance between the latest touch event (t_i) and the first event (t_1): $d_i = \sqrt{(x_i - x_1)^2 + (y_i - y_1)^2}$. When d_i exceeds 40.62 mm, the recognizer shifts into the *Aiming* state.

Aiming state is triggered when the movement exceeds 40.62 mm

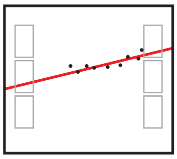
Here, the trajectory of the touches are calculated by fitting a linear regression $y = \alpha + \beta x$. We used the method of ordinary least squares (OLS) to calculate regression parameters α and β. Hence, at the i-th touch event, the parameters are determined as follow:

A linear regression of the touch points represents the trajectory

$$\overline{x} = \frac{\sum_{j=1}^{i} x_j}{i}; \overline{y} = \frac{\sum_{j=1}^{i} y_j}{i}$$

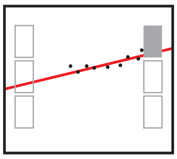
$$\beta = \frac{\sum_{j=1}^{i}(x_j - \overline{x})(y_j - \overline{y})}{\sum_{j=1}^{i}(x_j - \overline{x})^2}$$

$$\alpha = \overline{y} - \beta\overline{x}$$



From the linear regression, we are able to narrow down two directions (at the two ends of the regression line) of targets. Only the target that the touch trace moves towards is a potential selection. In other words, we chose the target k that has lower 2D Eucledian distance between the center of the target ($c_{k,x}, c_{k,y}$) and the latest coordinate (x_i, y_i). At this point, it is possible to provide intermediate visual feedback, which is described in section 2.4.3. The linear regression is up-

The trajectory determines the target.



dated when the new touch points are detected. This allows users to refine their selection when necessary. We tested with weighted regression (e.g., by giving more weight to the recent touch points), but we found that using all points yielded better numerical stability.

Finishing and cancelling the gesture

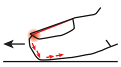
If users lift the finger longer than 250 ms in the *Aiming* state, the aimed target is selected. This changes the recognizer to the *Recognized* state and resets the recognizer. Cancelling swabbing is possible, in the *Aiming* state, the user can move the finger back to the starting point, reducing the distance d_i to be less than 40.62 mm.

2.4.2 Target Placement and Bi-directional Swabbing

To reduce the number of goal-directed movements, targets were placed on the edges of the screen.

Intention tremor aggravates in goal-directed movements. Therefore, we minimized the number of aiming required for swabbing. Once the swabbing gesture recognizer is in the Aiming state, users may lift the finger to select a target at anytime. To accommodate extreme tremors, we placed the targets on the edges of the screen. This allows users to continue the movement beyond the screen without having to aim where to stop. Some possible target layouts are shown in Figure 2.7a, b. We chose to use the radial layout over the linear layout to avoid relying on (x, y) coordinates of the initial contact point, which is likely to be unreliable from tremor oscillation.

We avoided swabbing in the upward direction because this movement pulls the nail bed.



In a later evaluation, we found that swabbing in the upward direction (away from the user's body) is difficult for many users. During the upward swipes, the friction between the screen surface and the finger pad pulls the the nail bed, which is more fragile than the skin in other directions of the finger pad. Therefore, we limited the possible opening angle to 270°, omitting the 90° in the upward direction, as shown in Figure 2.7c. (For more details, see [Huck, 2012].)

Bi-directional target placement allows doubling number of targets while retaining the same opening angles

Initially, we detected swabbing gestures that start around the center of the screen to simplify the conceptual model. In later studies, two of the participants proposed that swabbing should be allowed *bi-directional*: from the center of the screen outward, and from the edge of the screen inward. This extension requires additionally distinguishing whether the gesture originates from the center or the edge of the screen when choosing a target after linear regression. As a result, radial bi-directional swabbing allows doubling amount of the targets on the screen with the same opening angle for each target as shown in

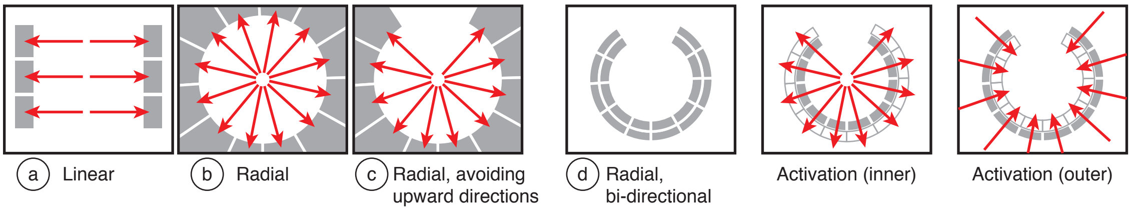


Figure 2.7: Target layouts for swabbing. Each of the filled grey shapes is a target. To activate each target, users perform swabbing gestures roughly in the direction pointed by the red arrow intersecting each target.

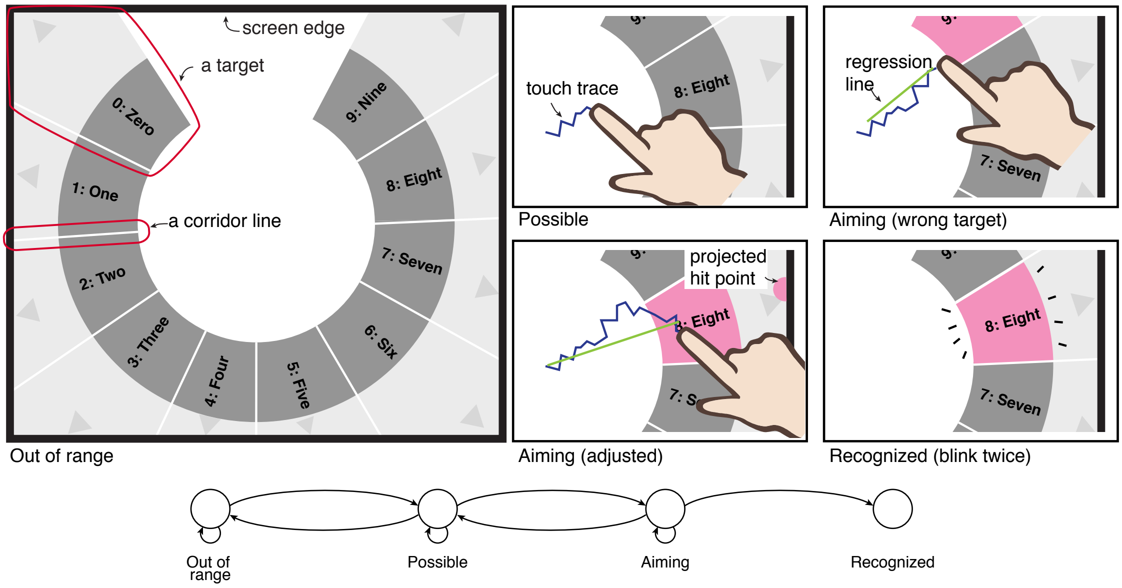


Figure 2.8: Visual guidance and feedback shown in each input state. Here, we illustrate a user intended to select "8:Eight", but initially aimed at "9:Nine".

Figure 2.7d. Nevertheless, we had an opportunity to test the radial bi-directional with only one user. For the rest of this chapter we used the radial uni-directional layout (Figure 2.7c), unless specifically mentioned otherwise.

2.4.3 Visual Guidance and Feedback

Figure 2.8 shows the visual design of guidance and feedback for swabbing at each gesture recognition state. In the *Out-of-range* state, adjacent swabbing targets are separated by a corridor line stretching to the

Visual corridor lines are added to signify the direction for swabbing.

edge of the screen to signify swiping action. At the edge of the screen, we placed a small arrowhead to indicate swabbing direction. To facilitate visual scanning, the target labels are placed towards the center of the screen and are aligned towards the outer circle of the target areas.

Swabbing gives two levels of visual feedback: users' action and system's interpretation.

When the user starts the swabbing gesture (*Possible* state), we draw a line based on touch points of the touch trace. This trace is the immediate feedback to users' action. After the trace exceeds the distance threshold (entering the *Aiming* state), three types visual feedback are shown: (1) a line segment in the direction of the linear regression is shown, (2) the projected hit point at the edge of the screen, and (3) target highlighting. By continuing the swabbing gesture, the screen shows how users' actions (represented by the touch trace) influence system's interpretation (line segment, projected hit point, highlighted target), allowing users to infer the causal link between the touch trace and the linear regression.

Visual feedback on selection

When the user lifts the finger from the screen, the swabbing gesture is recognized. The target blinks twice to acknowledge the user's action and the screen returns to the *Out-of-range* state.

2.4.4 Associating Swabbing Targets to the Origin

Swabbing aims to preserve the original visual layout of user interfaces.

To maximize the space users can use to stabilize their swabbing trajectory, the whole screen was used as an input area. Targets are positioned in a radial layout to minimize ambiguity among desirable swabbing trajectories. Nevertheless, many target selections in the real world require positional information to be useful. For example, in an email application (Figure 2.9a), there are many checkboxes that can be activated. Changing the layout of these checkboxes would take away the associated meaning with each of them. (In Figure 2.9a, each checkbox corresponds to an email message.) Users with hand tremor may still have a good vision. Thus, we aimed to preserve spatial information of the original user interface as much as possible.

Color-coded directional pointers are used to indicate corresponding swabbing targets.

We implemented a mode that allows aiming at small targets on the screen. Each clickable target on the screen is annotated with a pointer oriented at the direction corresponding to the swabbing targets (Figure 2.9b, c). To aid in distinguishing similar angles, the pointers and their corresponding swabbing targets are colored such that the the adjacent swabbing targets are distinguishable easily (Figure 2.9d).

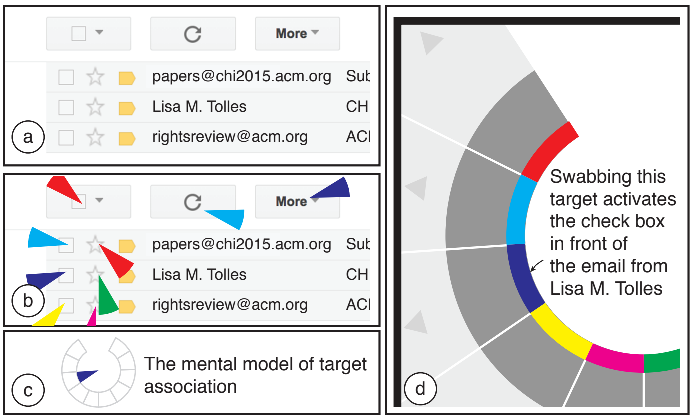

More
papers@chi2015.acm.org Sub
Lisa M. Tolles CH
rightsreview@acm.org ACI
a
b
c
d
The mental model of target association
Swabbing this target activates the check box in front of the email from Lisa M. Tolles

Figure 2.9: How swabbing visualize the association between swabbing targets and the original UI widgets. (a) A screen of an email application with several UI widgets of the same type. (b) Target pointers pointing to different directions. Adjacent directions are coded in different colors. (c) The mapping between the pointers and the swabbing target. (d) Swabbing user interface showing the corresponding swabbing targets.

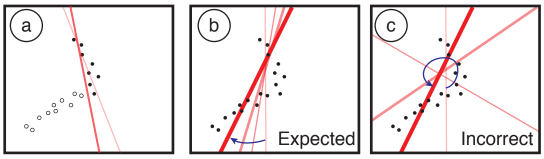


Figure 2.10: Swabbing crossing the Y-axis may result in unexpected visual feedback. (a) The touch trace of a swabbing from top to bottom. Filled dots are the touches that already occur. Blank dots are the touches to occur. The red lines show the regression. Recent regressions are thicker and more opaque. (b) The expected rotation. (c) The rotation in the opposite direction.

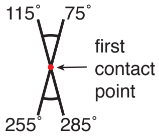
2.4.5 Edge Case: Crossing the Vertical Axis

Swabbing in the direction parallel to the Y-axis (henceforth *vertical axis*) of the screen causes the denominator of the β term $(\sum_{j=1}^{i}(x_j - \overline{x})^2)$ to converge to zero. (See section 2.4.1.) Such case can be trivially handled with a conditional statement in the implementation. However, trajectories crossing back-and-forth on the vertical axis can be problematic. This problem is especially remarkable when continuous visual feedback for swabbing is provided. We visualized the regression line continuously as the touch trajectory progress as

Visual feedback may be rotated in the wrong direction when touch trajectory crosses the vertical axis.

shown in Figure 2.10a. The slope β was used to specify a rotation angle for a keyframe animation. The expected animation should show the line gradually rotating clockwise (Figure 2.10b). However, when crossing the vertical axis, the rotation may be another way around, depending on the keyframe interpolation algorithm in each animation framework. These incorrect rotations dramatically change the visual feedback back and forth, causing users to be confused (Figure 2.10c).

Vertical movement detection threshold:



To address this problem, the swabbing recognizer detects if the touch trajectory heads close to the vertical direction. We found that simple thresholding works adequately: Vertical movements are detected when the latest contact point is between 75–115° or 255–285°. Upon detecting vertical movements, the X- and Y-axis are swapped during the calculation.

Alternative: different regression method

In the review of this thesis, Pierre Dracigevic suggested that this problem can be addressed by using an alternative regression method. The abovementioned problem occurs because the OLS defines the error function based on only one variable (the Y-axis in Figure 2.11a). This error function is algebraically optimized to a solution described in Section 2.4.1. Other regression methods, such as total regression (Figure 2.11b), define the error function from all variables. Replacing OLS with such method would be adequate to handle this edge case.

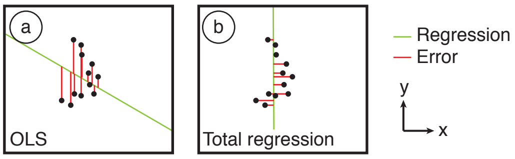


Figure 2.11: Error functions in OLS (a) and total regression (b). Optimizing the total regression would automatically handle the vertical movement.

2.4.6 Trace shifting for Multi-touch Screens

Subsequent touch traces are shifted to match the original trace by a fixed offset.

We observed that tremors cause multiple fingers to touch the screen simultaneously. The simultaneous touch traces from multiple fingers together with the jittering on- and off-surface destabilize the linear regression. Although touch traces from multiple fingers in a single hand are moving in the same direction, changing the trace association may dramatically alter the swabbing direction, as shown in Figure 2.12a–c. Therefore, once the original trace disappears, touch points from the adjacent traces that occur within 250 ms are shifted according to the offset of the initial location to the original trace. This combined trace is the input for linear regression as shown in Figure 2.12d–f.

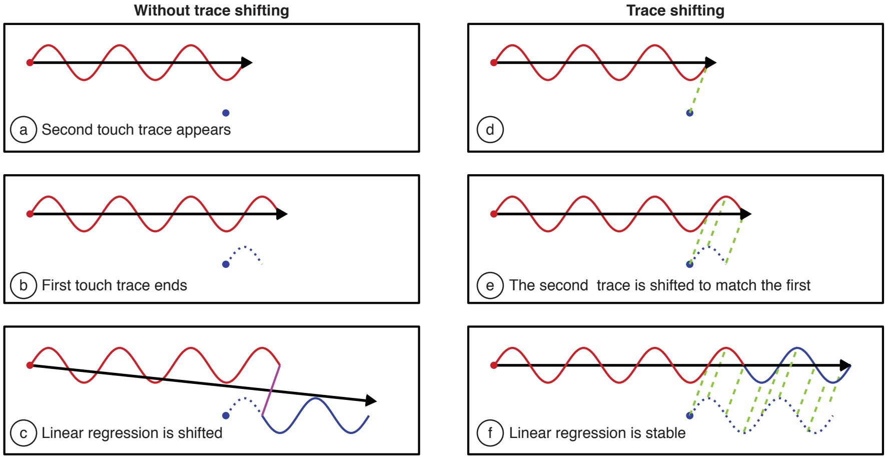
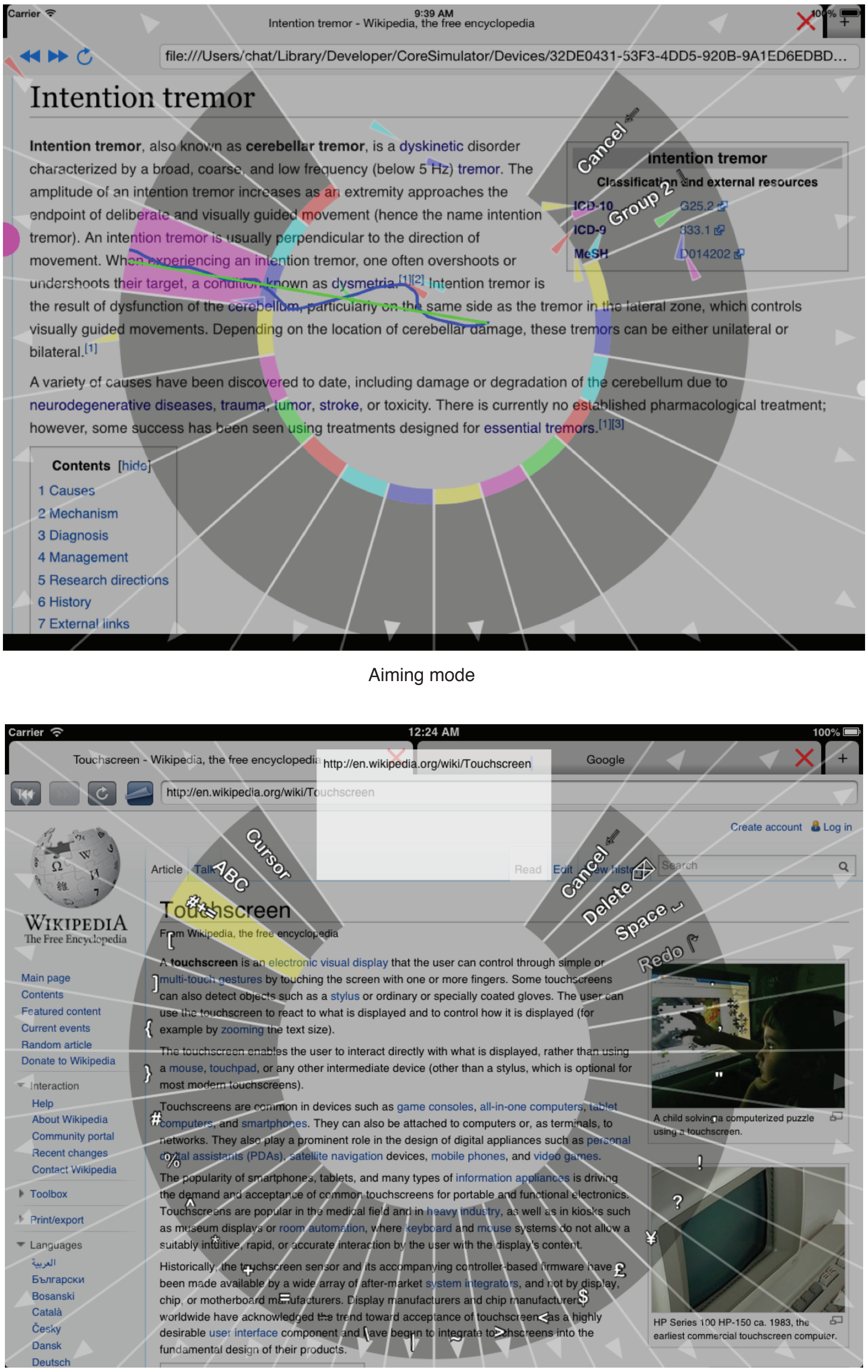

Without trace shifting
Trace shifting
a Second touch trace appears
d
b First touch trace ends
e The second trace is shifted to match the first
c Linear regression is shifted
f Linear regression is stable

Figure 2.12: In (a–c), the interference of multi-touch traces that destabilizes the trajectory of the linear regression. In (d–f), the subsequent trace is shifted back, based on the offset of the first touch point in the new trace. (Modified image courtesy of [Kehrig, 2013])

2.4.7 Prototype: Swabbing Web Browser

The latest version of swabbing input technique was implemented as a web browsing application for iPad (Figure 2.13). Users can toggle swabbing input by tapping five fingers on the screen (to accommodate tremors, touches within a one-second window are considered together). Swabbing input can be used for selecting targets (such as hyperlinks or HTML UI widgets), navigating between web pages (back, next) and for text entry.

Swabbing can be used for input necessary for web browsing.



Intention tremor - Wikipedia, the free encyclopedia

file:///Users/chat/Library/Developer/CoreSimulator/Devices/32DE0431-53F3-4DD5-920B-9A1ED6EDBD...

Intention tremor

Intention tremor, also known as **cerebellar tremor**, is a dyskinetic disorder characterized by a broad, coarse, and low frequency (below 5 Hz) tremor. The amplitude of an intention tremor increases as an extremity approaches the endpoint of deliberate and visually guided movement (hence the name intention tremor). An intention tremor is usually perpendicular to the direction of movement. When experiencing an intention tremor, one often overshoots or undershoots their target, a condition known as dysmetria.[1][2] Intention tremor is the result of dysfunction of the cerebellum, particularly on the same side as the tremor in the lateral zone, which controls visually guided movements. Depending on the location of cerebellar damage, these tremors can be either unilateral or bilateral.[1]

A variety of causes have been discovered to date, including damage or degradation of the cerebellum due to neurodegenerative diseases, trauma, tumor, stroke, or toxicity. There is currently no established pharmacological treatment; however, some success has been seen using treatments designed for essential tremors.[1][3]

Intention tremor	
Classification and external resources	
ICD-10	G25.2
ICD-9	333.1
MeSH	D014202

Contents [hide]

1 Causes
2 Mechanism
3 Diagnosis
4 Management
5 Research directions
6 History
7 External links

Aiming mode

Touchscreen - Wikipedia, the free encyclopedia | Google

http://en.wikipedia.org/wiki/Touchscreen

http://en.wikipedia.org/wiki/Touchscreen

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact Wikipedia

Toolbox

Print/export

Languages
العربية
Български
Bosanski
Català
Česky
Dansk
Deutsch

Create account Log in

Article Talk Read Edit View history Search

Touchscreen

From Wikipedia, the free encyclopedia

A **touchscreen** is an electronic visual display that the user can control through simple or multi-touch gestures by touching the screen with one or more fingers. Some touchscreens can also detect objects such as a stylus or ordinary or specially coated gloves. The user can use the touchscreen to react to what is displayed and to control how it is displayed (for example by zooming the text size).

The touchscreen enables the user to interact directly with what is displayed, rather than using a mouse, touchpad, or any other intermediate device (other than a stylus, which is optional for most modern touchscreens).

Touchscreens are common in devices such as game consoles, all-in-one computers, tablet computers, and smartphones. They can also be attached to computers or, as terminals, to networks. They also play a prominent role in the design of digital appliances such as personal digital assistants (PDAs), satellite navigation devices, mobile phones, and video games.

The popularity of smartphones, tablets, and many types of information appliances is driving the demand and acceptance of common touchscreens for portable and functional electronics. Touchscreens are popular in the medical field and in heavy industry, as well as in kiosks such as museum displays or room automation, where keyboard and mouse systems do not allow a suitably intuitive, rapid, or accurate interaction by the user with the display's content.

Historically, the touchscreen sensor and its accompanying controller-based firmware have been made available by a wide array of after-market system integrators, and not by display, chip, or motherboard manufacturers. Display manufacturers and chip manufacturers worldwide have acknowledged the trend toward acceptance of touchscreens as a highly desirable user interface component and have begun to integrate touchscreens into the fundamental design of their products.

A child solving a computerized puzzle using a touchscreen.

HP Series 100 HP-150 ca. 1983, the earliest commercial touchscreen computer.

Text entry mode

Figure 2.13: Swabbing web browser (Images courtesy of [Kehrig, 2013])

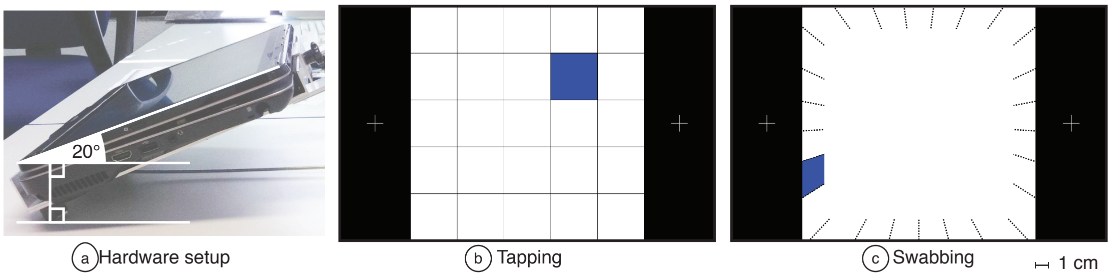


Figure 2.14: Hardware setup and task screens for study 1. (Image modified from [Wacharamanotham et al., 2011].)

2.5 User Studies

Swabbing was evaluated in two controlled experiments and a longitudinal case study. The extensive reports for these experiments are already published in Bachelors' and Diploma's theses, which are supervised by the author. Below, we summarize these studies and reanalyze a subset of results in the part that compare tapping with swabbing. We first describe the two controlled experiments (target selection study and typing study) combined because the hardware setup and the tasks are similar.

We conducted two controlled experiments and a longitudinal case study.

2.5.1 Apparatus for the Controlled Experiments

As shown in Figure 2.14a, we used HP TouchSmart tm2-1090eg[4] (a single-touch screen) mounted on a stand to incline the screen surface 20° from the desk surface. Participants sit on a chair that is adjusted such that the height of the desk is at the same level of the elbow when the arm points straight to the ground. The screen was placed in front of the user within the length of the forearm. Each participant used the index finger of the dominant hand to interact with the screen. In both studies, the full radial layout of swabbing was used (Figure 2.7 b).

[4] http://h10025.www1.hp.com/ewfrf/wc/product?product=4107117

2.5.2 Participants

We recruited participants from University Hospital Aachen and the University Hospitals of the Ruhr-University of Bochum. All participants have intention tremor. None of the participants use touch screens frequently. Demographic information of the participants is shown in Table 2.1. To assess tremor severity, we asked the participants to use the spiralometry (Figure 2.3) to measure their tremor.

	Study 1: Selection	**Study 2: Typing**
Number of Participants	7 (2 females)	12 (1 female)
Age Mean (SD)	75 (14.42)	72 (5.07)
Tremor severity:		
- Slight	1	0
- Moderate	1	4
- Marked	3	3
- Severe	2	5

Table 2.1: Demographic information of the two swabbing experiments

2.5.3 Tasks

Study 1: select the highlighted target on the screen.

In study 1, users were asked to select a target on the screen by either tapping or swabbing (Figure 2.14b, c). The task was shown at the center of the screen in a 163 x 163 mm square space. We used the entire square to render the target. For tapping, this space was divided into a grid of square in equal size without gaps in-between. In each trial, users first place the finger on a crosshair at the side. Then, a square is highlighted for the users to tap (land and lift). Then, the users home the finger back to the crosshair to complete the trial. We avoid giving visual feedback from swabbing (trajectory and selected target) in these two studies because intention tremor intensifies in aiming movement. Showing visual feedback in the swabbing condition could implicitly encourage users to aim, hence, increase tremor. To keep both conditions equal, visual feedback was also removed from the tapping condition. The highlighted square returns to normal only when the users land the finger in the home area.

No visual feedback was provided.

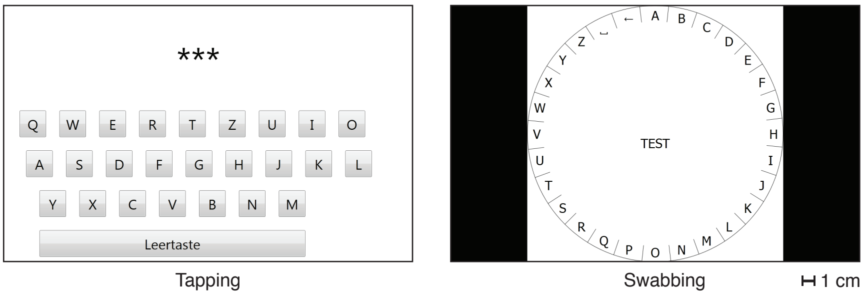


Figure 2.15: Task screens used in study 2. (Images modified from of Huck [2012].)

Study 2: transcribe words

In study 2, users were asked to transcribe short German words using either a touch screen keyboard or swabbing (Figure 2.15). The same words were used for all participants, but the order was randomized for each condition (27 characters in all). We implemented a touch screen keyboard with large buttons (16.5 mm width) and large gaps between the buttons (6.35 mm), as recommended for elderly users by [Jin et al., 2007]. To avoid a learning effect and to let the users finish the entire task without worrying about excessive mistakes caused by tremors, we showed an asterisk as feedback for each keystroke, similarly to Findlater et al. [2011]. There was no backspace button. Participants were asked to continue typing without corrections. Similarly, for swabbing, we only provided asterisks as feedback for each character selection.

Asterisks were provided as feedback.

2.5.4 Design of Experiments

Study 1: 2 input techniques × 4 target sizes

In study 1, we compared tapping and swabbing with 9, 16, 25, and 36 targets on the screen in a within-subjects study. This corresponded to the square button width of 5.43, 4.08, 3.26, and 2.71 cm, respectively. We counterbalanced the order of methods and the number of targets. Users selected 15 targets in each condition. Before testing each input technique, users selected 10 targets for practice. Each user used two techniques for three on-screen targets for 15 selections, giving 90 data points per user.

The full experimental design of study 2 involved six keyboard conditions in a within-subjects design, for more details, see [Huck, 2012]. Each user contributed one data point per technique as described next.

2.5.5 Dependent Variables and Data Analysis

Error rates and speed were measured

We measured errors and speed for both studies. In study 1, we counted the number of errors in each trial. We measured the target acquisition time, which was the length of time that the user took to acquire each target between making the last contact point in the home area to making the first contact point after performing the selection gesture. In study 2, the error rate was the ratio of correctly selected characters to the number of all characters. The speed is in seconds per character.

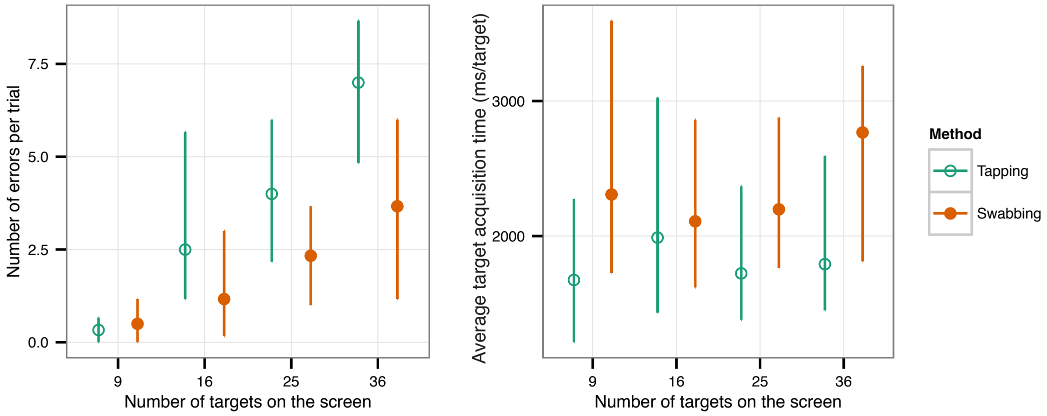
Bootstrapping was used for data analysis

To calculate the means, we used ordinary non-parametric bootstrapping (10,000 replicates). Confidence intervals (CIs) were calculated with the bias-corrected and accelerated method (BCa). Some confidence intervals are asymmetric, which reflects the distribution of the data. All error bars are 95% CIs. The within-subjects differences between the means were calculated using the tapping condition as a baseline with the same bootstrap resampling procedure. In both studies, we log-transformed the speed measurement before running the statistical analysis. The plots were inverse-transformed (anti-logged).

2.5.6 Results and Discussion

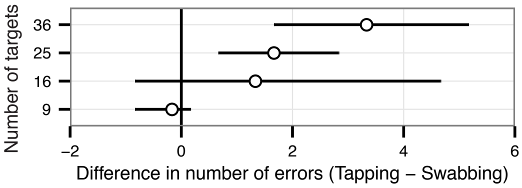
Tapping errors increased faster in smaller targets.

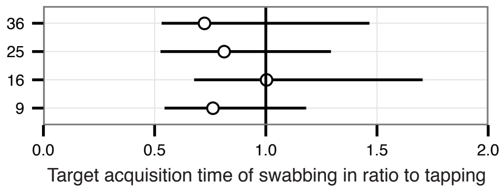
The charts below shows the descriptive statistics from study 1. The number of errors tended to increase with the number of targets, and the number of errors in the tapping condition seemed to increase faster than those in the swabbing condition. There is no clear evidence of a similar trend in the acquisition time in both conditions.



As for the within-subjects differences, the left chart below shows the bootstrap mean differences and their confidence intervals. The data points to the right of the zeros indicate that tapping yielded more errors than swabbing. This graph shows that swabbing performs similarly to tapping in 9 targets and 16 targets and yielded fewer errors in high numbers of targets. In terms of time, the right chart below shows that the acquisition times for both techniques were comparable in all the targets because the CIs captured 1. Due to log transformation, the mean and CIs for the target acquisition time shown here are in ratios relative to tapping, which is the baseline.

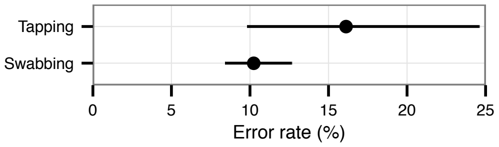
Swabbing was more accurate in smaller targets.

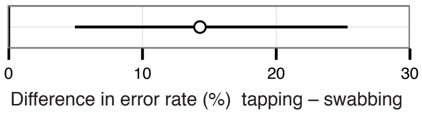




The error rate in study 2 is shown in the charts below, tapping seemed to yield a higher error rate. The wider CI indicates that the error rate varied greatly across users. Swabbing, on the other hand, had a much narrower CI. The within-subjects difference shows similar results as those in study 1. The CI is on the right of zero, indicating that tapping yielded around 5–25% more errors than swabbing.

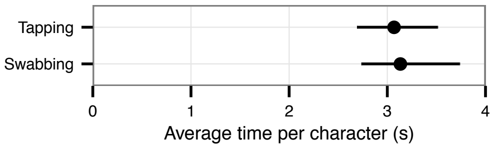
The results from study 2 supports those from study 1 in terms of error rate.

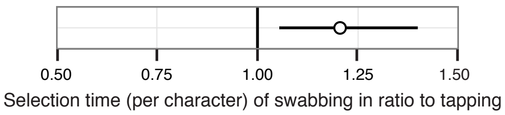




As for the speed, the charts below show that users were around 1.2 times slower in swabbing. Nevertheless, with an average duration of approximately 2.8–3.8 seconds per character, the additional duration may be acceptable because of the gains in accuracy.

In study 2, users took more time in swabbing than in tapping.





2.5.7 Longitudinal Case Study

To evaluate how the swabbing technique may be used in the long-term, we recruited a participant to use the swabbing web browser (section 2.4.7) on an iPad at home. Our participant was a 66-year-old man with a severe intention tremor. He used a desktop computer regularly, but he had no experience using touch screens.

Three periods (tapping, swabbing, swabbing bi-directional), six sessions each

We asked the participant to use the device for one month, once a day. We allowed the user to skip days on which he was busy. In all, he used the device for 18 sessions. These sessions were separated into three periods of six sessions each. During the first period, we asked the users to use the default iPad keyboard for input. During the second period, the user used the swabbing web browser with a uni-direction radial layout. After a few sessions during the second period, the user wanted to try the bi-directional layout. Therefore, during the third period, he used the swabbing keyboard with the bi-directional layout.

Task: text-entry (for quantitative results) and web-browsing (for qualitative results)

Each session included a text-entry task and a free web-browsing task. During the text-entry task, we aimed to capture how the user's performance evolved as he grew more familiar with each technique. As for the free web-browsing task, we aimed to capture qualitative feedback about the user's experience with swabbing. Below, we summarize a subset of results from the text-entry task to discuss the learning effect, for other results see [Kehrig, 2013].

Tapping was faster, but swabbing was more accurate.

As shown in Figure 2.16 (left), the user was roughly twice as fast with tapping; however, there was a large variation within each session. Although the user was slower in both tasks when swabbing, the performance was consistent. The data also suggest a slight improvement over sessions, but the time window is too narrow to assess whether the performance had already plateaued. Figure 2.16 (right) plots the corrected and uncorrected error rates. The low uncorrected error rate in all three conditions indicates that the user was equally careful when entering the text in all periods. He was careful to correct any mistakes that occurred before finishing each trial. The high corrected error rate in tapping indicates that the backspace key was pressed many times to correct errors. This indicates that although swabbing is slower, it allows users to be more accurate when entering each of the characters.

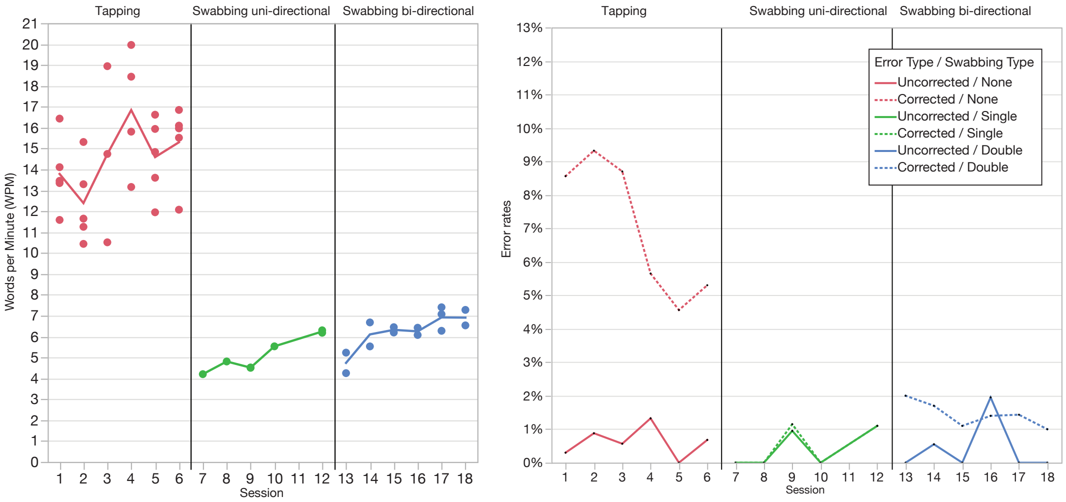


Figure 2.16: The quantitative results from the longitudinal case study, showing potential learning effect of swabbing. Text entry speed in WPM (left) and text input error rates (right) Note: There are more data points in the first six weeks because the user asked to shorten the sessions in later weeks. (Images courtesy of Kehrig [2013].)

2.6 Limitations and Future Work

More testing is needed for the bi-directional layout

The bi-directional swabbing layout was developed during the course of the longitudinal case study. Only one user had tested the technique in the last phase of the study, therefore, his experience with the swabbing technique might allow him to use the bi-directional layout more successfully. Further testing with users who have various tremor severity is recommended as a future work.

Swabbing for touch screen phones

Swabbing is developed for tablet-size screens. During the course of the evaluation, many participants expressed interest in having swabbing in a more ubiquitous touch screen phones. It is possible to put a the phone in an armband to prevent users from dropping it because of hand tremors. Nevertheless, there are two major challenges for future work in swabbing in small screens: (1) less space is available for users to stabilize the trajectory, and (2) visual guidance and feedback during swabbing could be hard to see due to the occlusion by the palm.

User performance models

Input accuracy in swabbing can be influenced by many factors. Two of such factors are the opening angle available for each target and the distance allowed for swabbing. Although a layout with a large opening angle is less error prone, only a few targets can be fit into such layout. This could increase the amount of pages required to select a desired target, hence slowing down the input. Longer swabbing distance allows the linear regression to stabilize, and allows users to use immediate visual feedback to correct the trajectory. However, both of these actions require longer input time per stroke. As a future work, modeling the relationship between opening angle, swabbing distance, error rate, and stroke time will allow interaction designers to find a sweet spot for future swabbing interaction designs.

2.7 Conclusion

In this chapter, we learned about how intention tremor causes both space and state inaccuracy on touch screens. We proposed a swabbing input technique that allows users to select targets more accurately. Thus, one technique to address the inaccuracy problem is to design an alternative interaction technique. In the next chapter, we will turn to an accuracy problem for able-boided users.

Chapter 3

Evaluating State-switching Techniques in Indirect Multi-touch Input

Large touch screens, whether placed horizontally or vertically, are unergonomic.

Large horizontal multi-touch displays causes users ergonomic problems. An observation by Morris et al. [2007] found that many users work with such displays in a standing position, or have to lean over to reach objects at the far side of the screen. Placing the touch screen vertically is more comfortable for reading without neck pain [Forlines et al., 2007, Schmidt et al., 2009]. However, lifting the arm continuously to interact with vertical displays leads to fatigue, as known as *the gorilla arm effect* [Boring et al., 2009, Hincapié-Ramos et al., 2014].

Muscle activations in large touch screen usage is higher than other form factors.

Bachynskyi et al. [2015] captured postures of users while performing the Fitts's law task on different setups of touch screens. The data is then used to calculate muscle usage from a biomechanical simulation. As shown in Figure 3.1, using the touch screen vertically yields highest muscle activation, especially from the shoulder. Although placing the screen horizontally allows the arms to rest on the screen, it still requires more muscle exertion than using a typical tilted laptop screens. The difference is pronounced in the lower back because none of the participants rest their backs against the chair.

Publications: The work in this chapter was done in collaboration with Simon Voelker. Part of this work was published as a full paper at CHI 2013 conference [Voelker et al., 2013]. The author of this dissertation contributed to the design of the experiments and data analysis.

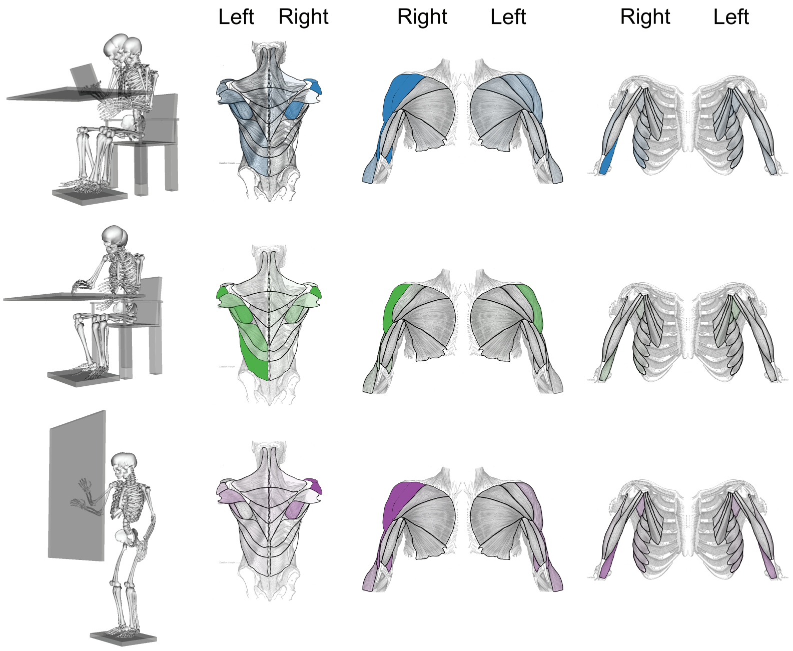


Figure 3.1: Muscle groups activated during horizontal and vertical display usage. (Adapted from [Bachynskyi et al., 2015] with permission.)

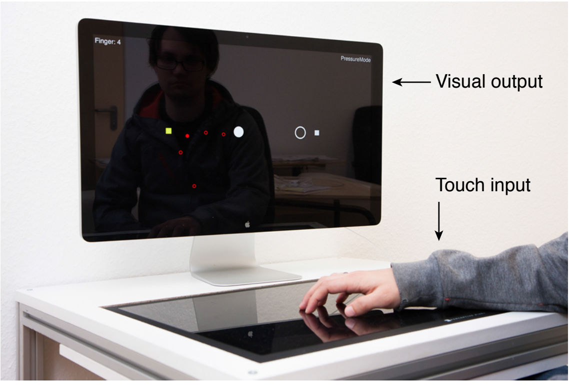


Figure 3.2: An indirect multi-touch system used in our experiments. (Modified image from [Voelker et al., 2013].)

Indirect multi-touch systems avoid fatigue problems and reduce occlusion.

To address these ergonomic issues, previous research proposes the use of both vertical and horizontal touch displays together [Morris et al., 2007, Weiss et al., 2010b, Schmidt et al., 2009]. Among these proposals are *indirect multi-touch systems* that only allow input on the horizontal screen and output on the vertical screen. Figure 3.2 shows an example of such system. This prevents both neck pain and the gorilla arm effect [Schmidt et al., 2009, Moscovich and Hughes, 2008]. Additionally, such indirect input systems also avoid space inaccuracy caused by finger occlusions, resulting in higher input precision [Forlines et al., 2007, Knoedel and Hachet, 2011].

However, indirect input requires cursor tracking.

Nevertheless, to allow users to be aware of the current position of their fingers without having to look down to the horizontal screen, a cursor tracking (state 1) is needed. In a system by Schmidt et al. [2009], cursors are tracked when the fingers hover near the horizontal surface. The system was compared with a direct multi-touch system in an aiming task. The results indicate that users perform slower in the indirect system. The authors surmised that the user experience degraded due to the fatigue of hovering fingers to track the cursors. Therefore, resting the hand on the screen should be used for cursor tracking (state 1). But how would the user activate the object below the cursor (switching to state 2)? In this chapter, we compare four techniques in terms of their state accuracy: how each technique enables users to switch between tracking the cursor and activating the target as intended on an indirect multi-touch system (Figure 3.3).

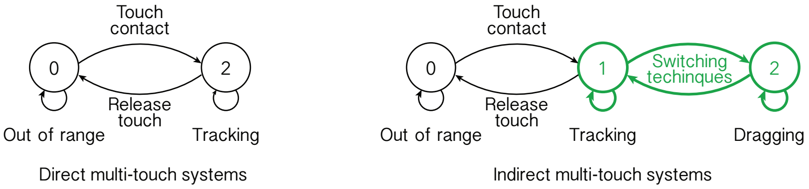


Figure 3.3: A state model for each touch in direct (left) and indirect (right) multi-touch systems. Highlighted are the state transitions that need to be investigated.

3.1 Design Considerations

1. Can be executed by each finger individually.

We belive that state-switching techniques for indirect multi-touch input should preserve the independence of individual fingers. Although 80% of the degrees of freedom come from the index finger and the thumb [Moscovich and Hughes, 2008], the remaining fingers can be used for gesture disambiguation [Kin et al., 2009].

2. Minimize state slips

State accuracy is essential for such techniques. In particular, it should be easy for the user to maintain the cursor in an intended state. In psychology, errors that occur in execution whilst a correct goal are called *slips* [Norman, 2002]. Slips occur in behaviors that depend on stored knowledge (e.g., motor programs) instead of feedback from the environment [Reason, 1990]. Therefore, we will call the errors in controlling touch states the *state slips*, which falls into two types:

Definition: *Slips in indirect multi-touch input*

SLIPS IN INDIRECT MULTI-TOUCH INPUT:
Slip-in: A slip-in occurs when the user unintentionally switches from state 1 to state 2 (e.g., from tracking the cursor to dragging an object)

Slip-out: A slip-out occurs when the user unintentionally switches from state 2 to state 1 (e.g., from tracking the cursor to dragging an object)

3. Ergonomic

Finally, state-switching techniques themselves should not be physically demanding. Otherwise, they would defeat the original purpose of the indirect multi-touch form factor.

3.2 Related Work

Our interaction designs were inspired by the literature from the single-touch input and direct touch input.

3.2.1 Single-touch Input

Pressure was used for state switching.

Several techniques were proposed for single-touch input. For indirect single-touch input, Buxton et al. [1985] used pressure to distinguish between state 1 and state 2. Touching on the screen activates state 1. State 2 is activated by increasing the pressure. In a follow-up work, Forlines et al. [2005] presented several applications for pressure-based state switching.

Lift-and-tap is a gesture for state switching.

MacKenzie and Oniszczak [1998] compared three switching techniques on trackpad: depressing a physical button, exerting the finger pressure (and receiving tactile feedback), and lift-and-tap. To click with the *lift-and-tap* technique, the user starts from (1) having the finger on the trackpad (tracking the cursor), (2) lifting the finger, (3) landing the finger on the same position within a short duration. (4) lifting the finger again. (Step 2–4 corresponds to a tap.) From their results, the most accurate and the slowest condition was the physical button. The pressure technique was the fastest but the most error prone. The lift-and-tap technique balanced the error rate and speed. Thus, the lift-and-tap technique was another candidate for our test.

Using take-off action for state switching prohibits dragging.

Potter et al. [1988] proposed the *take-off* technique. In their direct-touch system, touching the screen shows a cursor with a constant offset from the fingertip. To activate an object below the fingertip (switching from state 1 to state 2), the user had to lift the finger while the cursor is above the object. Olwal et al. [2008] introduced finger *rubbing* to zoom into a target and release the contact point to select the object below the finger, similar to take-off. Both take-off and rubbing prohibits the user from dwelling in state 2 (no dragging).

3.2.2 Multi-touch Input

Other techniques used multiple fingers or hands.

Olwal et al. [2008] also proposed to use tapping from the non-dominant hand to switch between states. Matejka et al. [2009] used multi-touch gestures for emulating mouse input. State switching is done by depressing multiple fingers (chording) on the screen. Nevertheless, both techniques reduced touch expressiveness by coupling multiple fingers into one input point.

3.3 Candidate State-Switching Techniques

We considered only the techniques that use properties of each finger individually.

One design consideration is that each finger can switch the input state independently from other fingers. Therefore, we screened the techniques from the literature for those that use only the input properties of each individual finger. Wang and Ren [2009] characterized four types of finger input properties on touch screens:

1. *Position properties:* coordinate (x, y)
2. *Motion properties:* velocity and acceleration
3. *Physical properties:* contact area (size, shape, and orientation) and touch pressure
4. *Event properties:* such as tap and flick

The position and motion properties are already occupied.

The position and motion properties are already used to communicate spatial information to the screen (i.e., "where it is being touched"). For communicating state information, we are left with the physical and event properties.

We used pressure for state switching.

Wang and Ren [2009] found that the orientation of contact area is hard to control for one finger without influencing others. Due to the softness of the skin, it is hard to control finger pressure separately from the size and shape of contact area [Pawluk and Howe, 1999]. In fact, the size and shape had been used to estimate pressure [Benko et al., 2006]. Thus, in the following, we use the term pressure to refer to these properties together.

We used events for state switching.

For the event property, flicking was discarded because it influences the position and motion of the contact point. Two events are widely used in touch input: the tap event (same as lift-and-tap above) and the hold event (dwelling on the same position longer than a threshold).

Four techniques were derived from these properties.

From the combination of the event and pressure properties, we derived four techniques from in literature: tap, hold, pressure hold, and pressure switch. Below, we describe each technique and its gesture recognizer. To determine the thresholds for the recognizers, we elicited behavioral data from five participants in an informal study. We used the same hardware as in our main study. (See Box 1 on page 51) Figure 3.4 presents a schematic of the gesture recognizer and the state model of each technique.

3.3.1 Tap Technique

Definition: *tap technique*

The *tap technique* is designed based on the Lift-and-tap technique by MacKenzie and Oniszczak [1998]. Here, the user lifts the finger off the screen and quickly lands it back. Lifting the finger may slightly change the contact point due to the softness of the fingertip and drifting. Each tap switches from state 1 to state 2, or vice versa. The user may leave state 2 directly to state 0 by keeping the finger lifted longer. To ensure space accuracy, we used the centroid of the last contact point before lifting for the tap event.

A radius threshold and two time thresholds were used in our tap recognizer.

To discriminate a tap from a finger repositioning (lifting and landing at another position), our tap recognizer registers a tap if the landed point is within a radius threshold r_{max}. This threshold needs to be

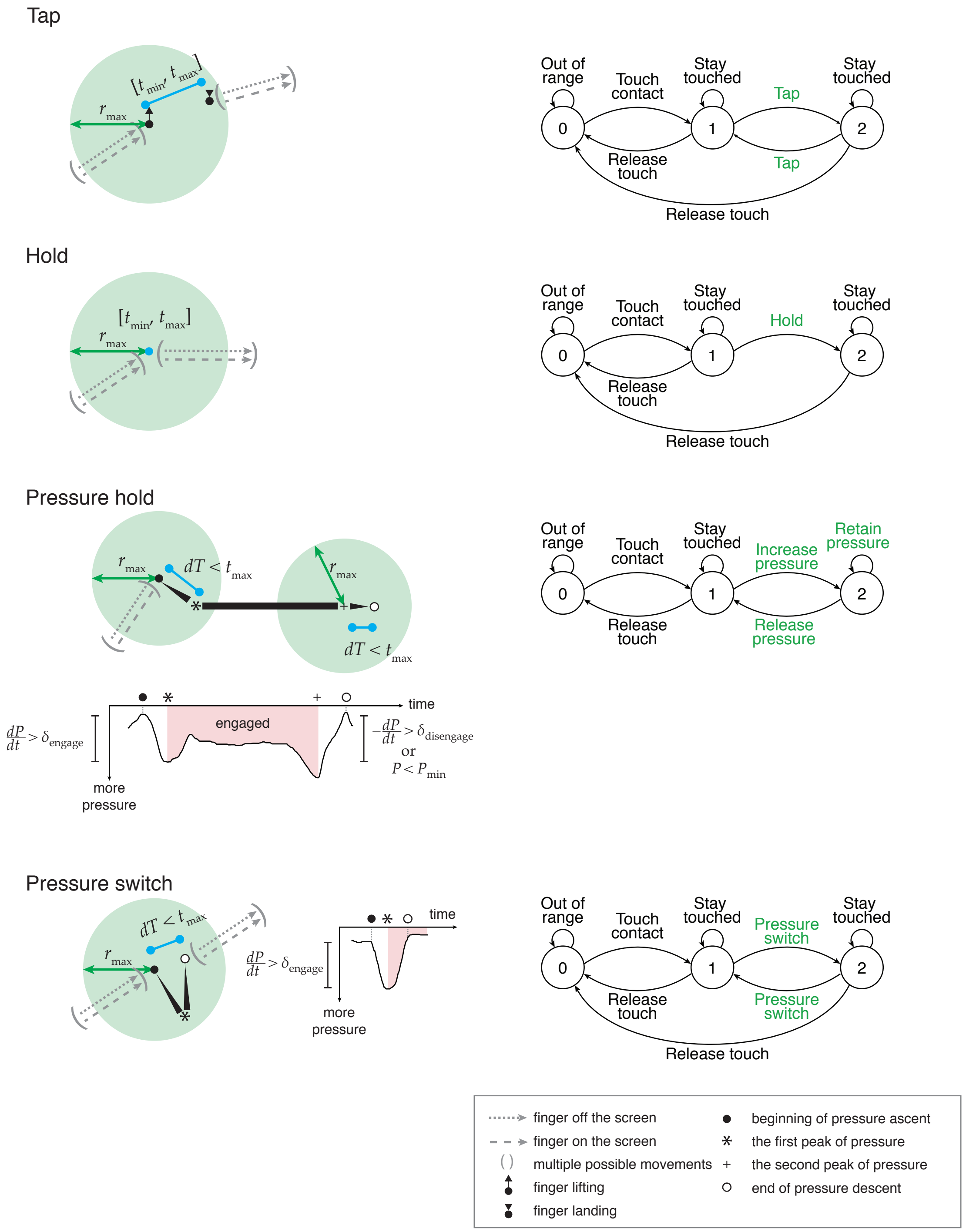


Figure 3.4: Interaction design schematic and state transitions of the four techniques. (Modified image from [Voelker et al., 2013].)

small enough to prevent confusion from a contact point of adjacent fingers. We used two duration thresholds: t_{min} and t_{max}. The lower bound t_{min} is needed in our system because sliding a finger quickly may cause a discontinuous touch signal. A system with higher frame rate may not require this threshold. The upper bound t_{max} allows us to classify a tap from an act of intentional lifting and landing touches.

Thresholds in the tap technique

To elicit the thresholds, we recorded users' taps on 12 positions, evenly spaced in a grid across the screen. After fine-tuning, we found that an r_{max} of 4.14 millimeters (75th percentile of the captured data) allows taps to be recognized without incorrect registration with the adjacent fingers. We used $t_{min} =$ 0.09 seconds and $t_{max} =$ 1.18 seconds (25th and 75th percentile of the captured data).

3.3.2 Hold Technique

Definition: *hold technique*

The *hold technique* is a common touch-screen gesture. It is used in Apple iPhone, for example, to move icons in the home screen. To switch from state 1 to state 2 with the hold technique, the user places the finger still on the screen exceeding a specific duration t_{max}. The cursor, then, stays in state 2 until the finger is lifted from the surface. The hold technique does not allow transitioning from state 2 back to state 1. We used the t_{max} of 0.5 seconds, based on the threshold in iOS 5 and Android 4.1. To determine how still the finger needs to be, we asked our users to place each finger still on the screen and capture the centroid of the contact point for 0.5 seconds. We used the radius threshold $r_{max} =$ 1.94 millimeters which is the 75th percentile of the captured data.

3.3.3 Pressure Hold Technique

Definition: *pressure hold technique*

The *pressure hold technique* is based on a technique by Buxton et al. [1985] and Miyaki and Rekimoto [2009]. Hard pressure is used to switch from state 1 to state 2. To dwell in state 2, the pressure must be maintained. Reducing the pressure switches back to state 1.

The rate of pressure change is used for gesture recognition.

Ohtsuki [1981] found that the ability of exerting pressure differs across fingers and changes when pressing multiple fingers simultaneously. Therefore, the absolute pressure should be avoided in recognizing pressure gestures. Thus, we used the rate of pressure change ($\frac{dP}{dT}$) in our recognition. Nevertheless, the user may intentionally release pressure slowly. To handle such situation, we used a minimum pressure threshold P_{min}.

We use touch ellipse to approximate touch pressure.

Like the majority of large capacitive touch screens, ours did not sense pressure directly. We needed to derive the pressure from other properties. The contact region of the finger is always near-elliptical [Cappelli et al., 2001], and the contact radius is proportionate with the pressure [Pawluk and Howe, 1999]. Therefore, we used the length of the semi-major axis of the touch ellipse, the ellipse approximation of touch contact. Henceforth, we will use the term pressure to refer to this approximation, and the pressure thresholds below refers to the length of the semi-major axis in millimeters.

Two issues from using touch ellipse: centroid drifts and ellipse changes during sliding with hard pressure.

Nevertheless, there are several challenges in using the touch ellipse to represent pressure. Firstly, unless the finger is perpendicular to the screen, the pressure usually is exerted from acute angles ($< 90°$). Because fingertips are soft, such exertion results in a small drift of the centroid of the touch ellipse. We use the threshold r_{max} to address this problem. Secondly, when the user slides a finger on the touch screen (e.g., while tracking the cursor or dragging an object), the shape of the touch ellipse also changes quickly according to the contact angle of the fingertip. Thus, we register a pressure hold only when the pressure change occurs within a short duration t_{max}.

Pressure hold gesture recognition

In summary, to switch from state 1 to state 2, the user increases the pressure quickly ($\frac{dP}{dT} > \delta_{engage}$ and $dT < t_{max}$), while the centroid of the contact point changes within the radius r_{max}. To switch from state 2 to state 1, the user either releases pressure quickly ($\frac{dP}{dT} > \delta_{disengage}$), or reduces pressure in less than an absolute threshold P_{min}.

Pressure thresholds were elicited from an imaginary dragging task.

To elicit these thresholds, we asked our participants to use the technique as if they were dragging an object from a starting point to a goal on the screen. Visual feedback was provided as normal touch screen dragging. The pressure level was not visualized. There were 12 start positions evenly distributed as a grid on the screen. The goal was randomized among these positions. We used the 75th percentiles of the elicited thresholds: t_{max} = 0.70 seconds, r_{max} = 7.07 millimeters, δ_{engage} = 1.30 millimeters/second, $\delta_{disengage}$ = 1.14 millimeters/second, and P_{min} = 0.55 millimeters.

3.3.4 Pressure Switch Technique

Definition: *pressure switch technique*

Instead of holding pressure, an impulse of pressure can be used for state switching as a *pressure switch*. In this technique the user toggles between state 1 and state 2 by exerting an impulse of pressure. Apple iPhone 6 used this technique to invoke quick actions menu on application icons.

Speed threshold for pressure switching

To recognize this gesture, we only need the rate of change $\frac{dP}{dT}$. We elicited this threshold by asking our users to perform the gesture on 12 positions, evenly spaced in a grid across the screen. We used the $\frac{dP}{dT}$ = 1.30 mm/s, which is the same for the pressure hold technique.

3.3.5 Potentials and Limitations

Table 3.1 summarizes the benefits and limitations of each technique.

Issue	Tap	Hold	Pressure hold	Pressure switch
Tactile awareness of input state		+	+	
Users' familiarity	+	+		
Flexion-extension coupling between fingers [Häger-Ross and Schieber, 2000]	-		-	-
Exertion and risk of repetitive strain injuries [Marras, 2012]			-	
Accidental activation while resting the finger		-		
Lack of state 2 to state 1 transition		-		

Table 3.1: Potentials and challenges of state switching methods

3.4 Experiments

We compared four state-switching techniques in three experiments.

We compared these techniques in three experiments: single-finger, two-fingers, and two-hands. These experiments aim to cover common use cases of multi-touch input. We hypothesized that the choice of technique affects state switching accuracy. Since the procedure of these experiments are similar, we describe the tasks and operational definition of accuracy measures for these experiments combined.

3.4.1 Tasks

Our hardware setup is described in Box 1. The task is displayed on the vertical screen, while the user manipulated the cursor by multi-touch input on the horizontal touch screen.

Box 1: Apparatus

As shown in Figure 3.2, we used two displays of the same size and output resolution. Both displays were 27″ (597 × 336 mm, 2560 × 1440 pixels) and were connected to a Mac Pro running the software for the experiments.

For input, touches are sensed by a capacitive touch screen from Perceptive Pixel (touch frame rate: 205 Hz). Nothing was displayed on this screen. The task was shown on a vertical Apple Cinema Display of the same screen size and resolution.

Participants controlled cursors on the vertical screen by touching the horizontal screen. Each contact point shows a circle cursor with a diameter of 7 mm (30 px). The cursors were outlined in the Tracking state and were filled in the Engaged state.

Having to manipulate multiple cursors can incour cognitive load. Therefore, we chose a straight forward mapping between input and output. We used 1:1 absolute mapping withtout any pointer acceleration.

Task: grab, drag, and drop objects without slipping.

To cover all state switching conditions, we created the following tasks based on the Drag and Drop task from Forlines et al. [2006]. The schematic of the three tasks are shown in Figure 3.5 (page 54).

Visual appearance of objects and targets

Before each task, the user crossed the cursors at the starting point (blank square), dragged the objects (filled circles) to the targets (blank circles), then moved the cursors to cross the finishing area (filled square). In experiment 3, the user had to drag two objects across the colored gates. Each gate can be opened by activating the buttons with the corresponding color. These buttons were placed on the side of the opposite hand.

No precise cursor positioning required.

Since we focus on state switching, we minimized the time for aiming at the drop target by the following: We used large objects and targets that can be easily grabbed without positioning the cursor precisely. We allowed a successful drop when the centroid of the object is within 1 cm from the target. We provided visual feedback by changing object and target colors when the cursor was in range for grabbing or dropping the object.

Multiple fingers are rested on the touch screens and multiple cursors were shown.

To mimic the ergonomic use of the setup we envisioned, the participants were asked to place as many fingers on the screen as they see comfortable, resulting in multiple cursors on the vertical screen. However, we instructed the participants to use only the cursors that are associated with the assigned fingers to interact with the objects, and we only used the data from these cursors in our analysis.

We asked for qualitative feedback after each technique.

After using each technique, we asked the participants to comment on speed, accuracy, and fatigue as well as to choose a preferred technique.

3.4.2 Dependent Variables

1. state slips

We count the number of slips. Users may *slip in* or *slip out* of the desired input state. For example, between the starting point and the object, the user may accidentally engage the cursor. In real use, this slip can degrade the user experience, e.g., by clicking other objects that the cursor passed over. We operationalized types of slips in Box 3. The definitions are slightly modified from [Voelker et al., 2013] for clarity. The locations of our tasks that each slip may occur are shown in Figure 3.5.

2. trial completion time

In experiment 1, we also measured trial completion time. Each trial comprised of the movement in all directions for each finger.

BOX 2: THE OPERATIONAL DEFINITION OF SLIPS

Experiment 1: Single Finger
Tracking slip-ins (TSI): The number of slip-ins that occur between the starting point and the object.

Dragging slip-outs (DSO): The number of slip-outs that occur while dragging the object towards the target. This slip-out causes the finger to drop the object.

Placement slip-ins (PSI): The number of slip-ins after the object is dropped onto the target. This slip-in re-grabs the object, and the participant needs to drop the object onto the target again.

Experiment 2: Two Fingers
Acquisition slip-ins in the second finger (ASI2): While the first finger is trying to acquire the object, ASI is the number of slip-ins in the second finger.

Acquisition slip-outs in the first finger (ASO1): While the second finger is trying to acquire the object, ASO is the number of slip-outs in the first finger.

Box 3: The Operational Definition of Slips (Continued)

Experiment 2: Two Fingers (Continued)
Dragging slip-outs in the first finger (DSO1) and in the second finger (DSO2): The number of slips-outs during dragging from each of the fingers.

Placement slip-outs in the second finger (PSO2): During the placement of the first object, PSO2 is the number of slip-outs of the second finger.

Placement slip-ins in first finger (PSI1): During the placement of the second object, PSI1 is the number of slip-ins of the first finger.

Experiment 3: Two Hands
Dominant hand slip-outs (SODH), and **Opposite hand slip-outs (SOOH):** The number slip-outs of the respective hand during bi-manual interaction.

3.4.3 Design of Experiments

We used a within-subjects design for our experiment. Table 3.2 summarizes the experimental design and demographics. We have eight participants in each experiment.

within-subjects design: 4 techniques

	Experiment 1: Single finger	Experiment 2: Two fingers	Experiment 3: Two hands
Age	24–34	24–38	24–30
Gender	All males	One female	Two females
Handedness	All right-handed	All right-handed	All right-handed
Movement directions	←, →, ↑, ↓	←, →	←, →
Fingers used	10 fingers	6 combinations of thumb, index, middle finger of each hand	Thumb and index finger
Per participant	160 trials 30 minutes	48 trials 35 minutes	8 trials 15 minutes

Table 3.2: Demographics information and experimental designs

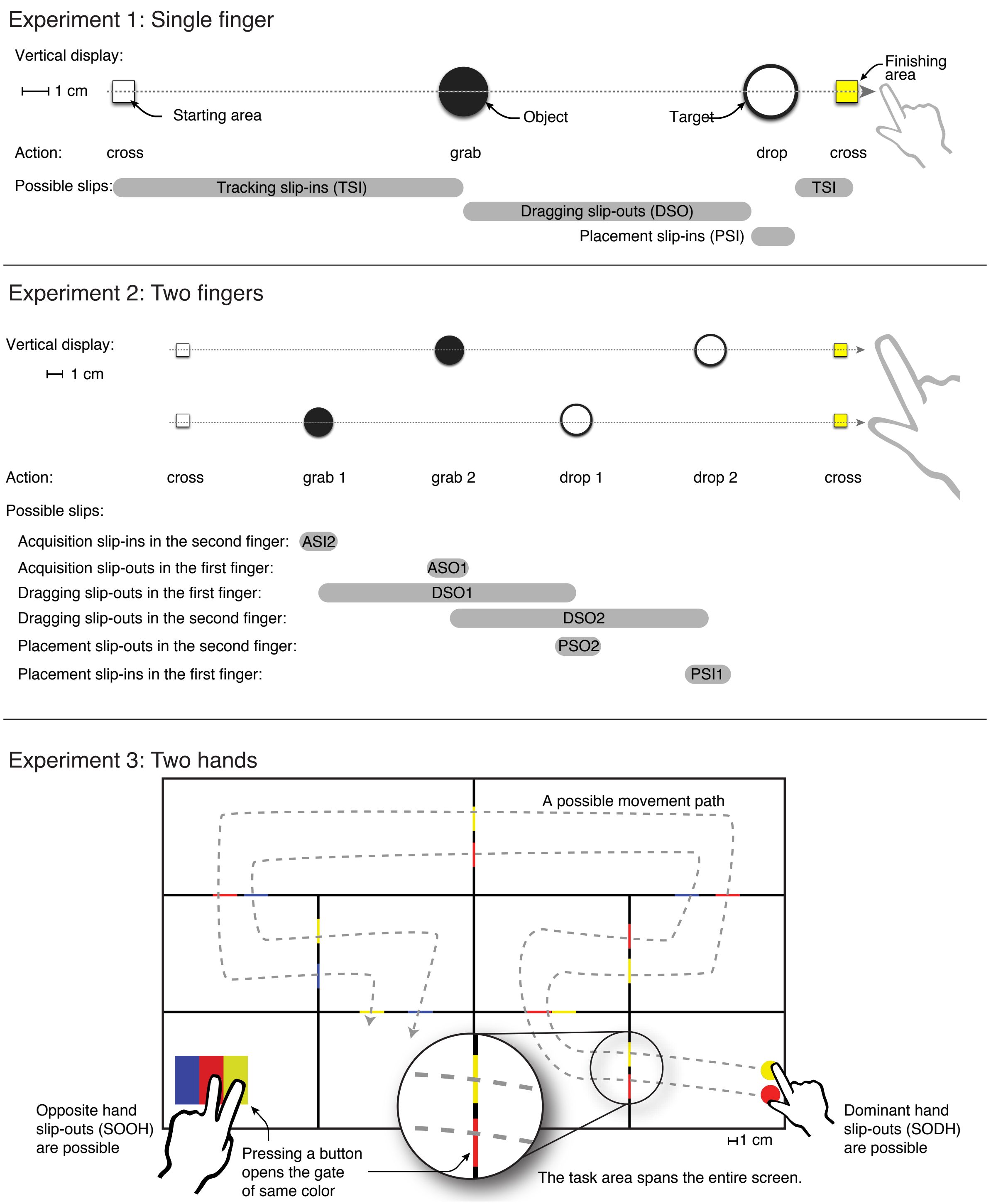

Experiment 1: Single finger
Vertical display:
1 cm
Starting area
Object
Target
Finishing area
Action: cross grab drop cross
Possible slips:
Tracking slip-ins (TSI)
TSI
Dragging slip-outs (DSO)
Placement slip-ins (PSI)
Experiment 2: Two fingers
Vertical display:
1 cm
Action: cross grab 1 grab 2 drop 1 drop 2 cross
Possible slips:
Acquisition slip-ins in the second finger: ASI2
Acquisition slip-outs in the first finger: ASO1
Dragging slip-outs in the first finger: DSO1
Dragging slip-outs in the second finger: DSO2
Placement slip-outs in the second finger: PSO2
Placement slip-ins in the first finger: PSI1
Experiment 3: Two hands
A possible movement path
Opposite hand slip-outs (SOOH) are possible
Pressing a button opens the gate of same color
1 cm
The task area spans the entire screen.
Dominant hand slip-outs (SODH) are possible

Figure 3.5: The task configuration for each experiment and associated state accuracy slips. (Modified image from [Voelker et al., 2013].)

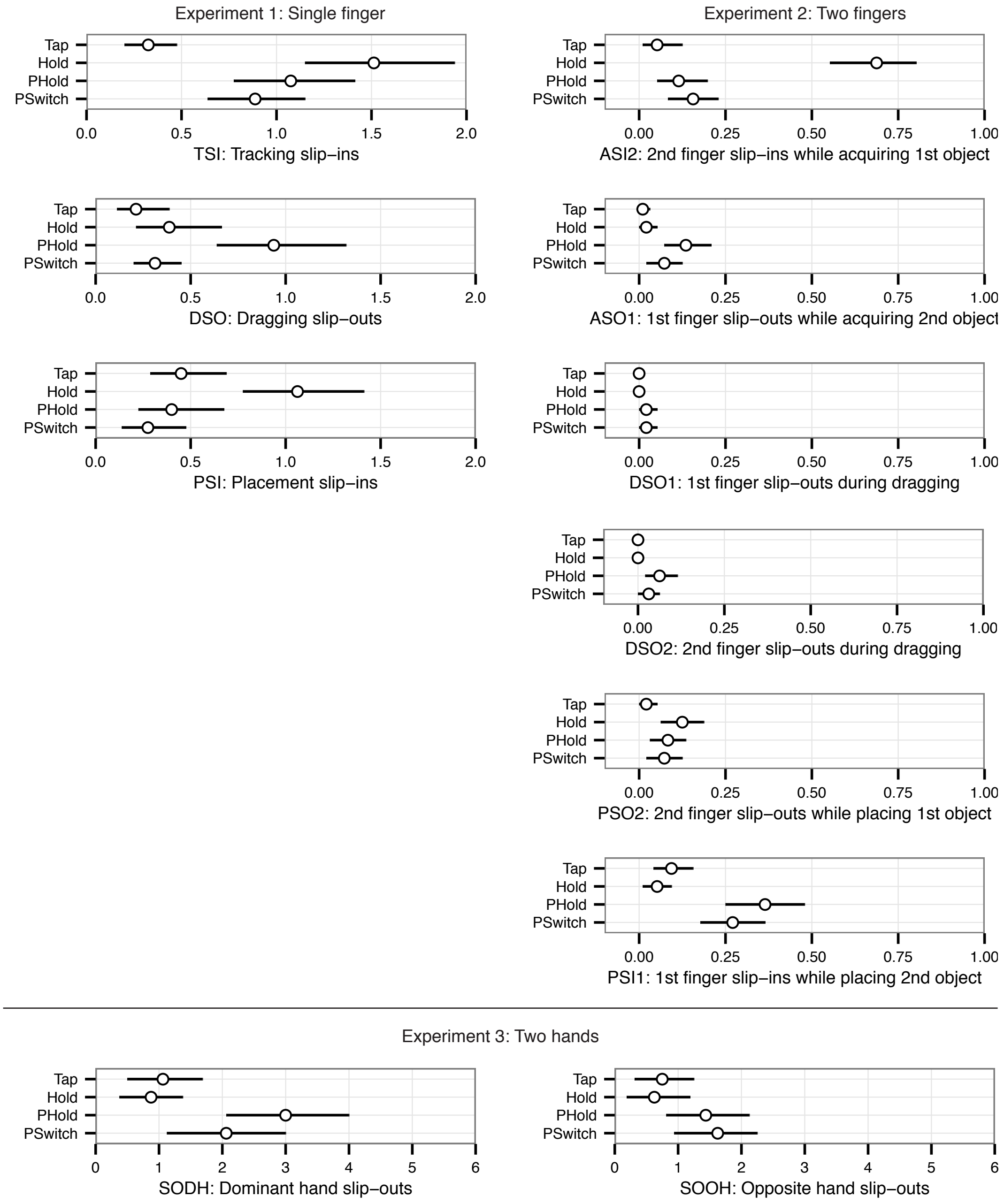


Figure 3.6: Slips per trial from the three experiments. (Mean and 95% CI without within-subjects adjustment) The length of each abscissa differs between experiments.

3.4.4 Data Analysis

Bootstrapping was used in analysis.

For each type of state slips, we averaged the number of occurrences per technique per user. To calculate means, we used ordinary non-parametric bootstrapping (10,000 replicates). CIs were calculated with the bias-corrected and accelerated method (BCa). Some confidence intervals are asymmetric, which reflects the distribution of the data. All error bars are 95% CIs.

Speed measurement were log-transformed.

For the speed measurement (experiment 1), we log-transformed the data prior of all calculation. The plots presented are anti-logged to the original scale. Thus, in the speed measurement, the means are geometric, and the differences between means are ratios.

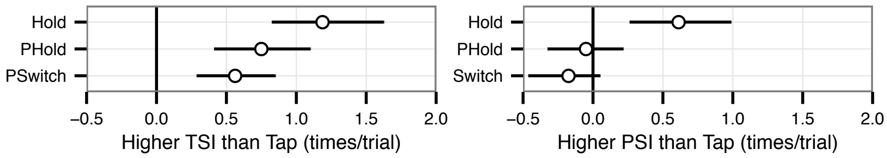
3.4.5 Results: Accuracy and Speed

Below, a point on the right means being worse than the tap technique.

The bootstrapped sample statistics of results is shown in Figure 3.6. We now take a closer look at interesting effects by plotting within-subjects mean differences in the figure below. We used the tap technique as a baseline: If a statistical estimate of a technique is on the right of the zero line, that technique yields more slips than the tap technique.

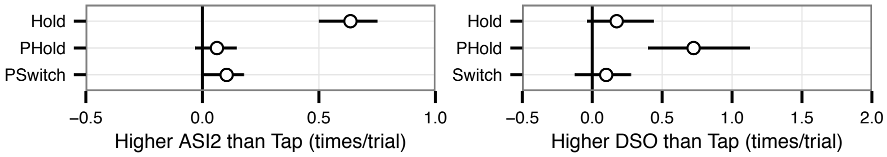
Tap: generally yielded least slips

As shown in the left chart below, the tracking slip-ins (TSI) estimate of all other techniques are right from zero. This is a strong evidence that the tap technique is more accurate than others for cursor tracking (dwelling in state 1). In general, the tap technique performed similar or better than other techniques across all measures (Figure 3.6).



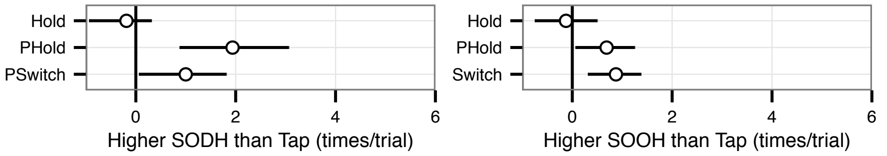
Hold: often yield slip-ins.

When compared with the tap technique, the hold technique is more likely to suffer from slip-ins. In the single-finger experiment, this is evident in both tracking (TSI) and object placement (PSI) as shown in the two charts above. In the two-finger experiment, when the first finger acquired the object, the second finger frequently slipped into state 2 (ASI2, in the left chart below).



However, the hold technique has low slip-outs, when the user needs to dwell long in state 2. It was comparable with the tap technique when dragging a single object (DSO, in the right chart above), dragging two objects, and holding a button in place (SODH and SOOH, in the charts below).

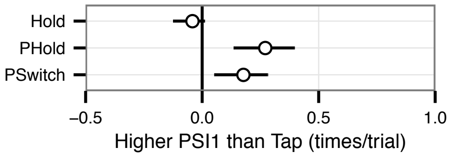
Hold: comparable to tap in preventing slip-outs.



Between the two pressure techniques, our users slipped-out more often when they used the pressure hold in the single-finger experiment (DSO). However, this effect seems to diminish when two fingers are dragging on the screen simultaneously (DSO1, DSO2, SOOH, and SODH). Using two fingers might allow users to maintain the contact area more stably. Both of the pressure techniques tend to be prone to slip-ins, when the second finger releases the object (PSI1, in the chart below). An explanation is that relieving the pressure from the second finger shifts the pressure to the first finger.

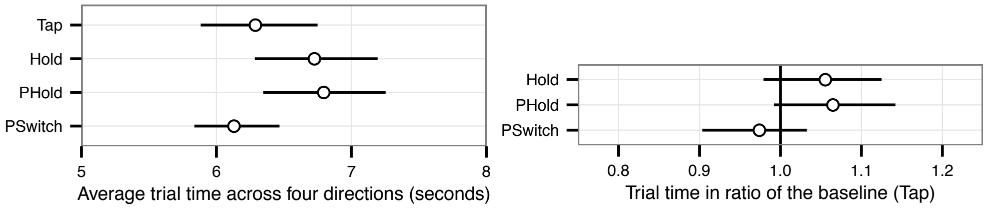
Pressure hold: high slip-outs when used with single finger.

Both pressure techniques often slipped in from shifting pressure between two fingers.



In terms of speed, the differences among the techniques are largely inconclusive as shown in the two charts below. The confidence interval of the differences crosses 1.0, indicating that the effect of technique on the speed is negligible.

All techniques took similar time to switch.



3.4.6 Results: Subjective Feedback

Participants were familiar with the tap technique.

The qualitative results agree with the results above. Most of the participants chose the tap technique as their preferred technique (Table 3.3). They mentioned the familiarity with the technique: *"I would use this if there were no instructions."* and *"It's the closest to the mouse."*

Participants mentioned slip-ins in the hold technique.

For the hold technique, users were bothered by the slip-ins: *"You get stuck somewhere all the time."* and *"You are always pressing something that you don't want."* However, our participants mentioned one benefit of hold in experiment 3: *"You are not losing the target on accident. You don't have to do anything with the right hand and tap the [buttons] with your left."*

Participants complained about exertion in the pressure techniques.

For the pressure hold technique, our participants complained about exertion needed: *"This cramps my hands up more than the others."* The pressure switch was reported to be *"hard to press just the [finger] you want"* in the two-finger experiment.

	Experiment 1: Single finger	Experiment 2: Two fingers	Experiment 3: Two hands
Tap	4	7	5
Hold	3		3
Pressure hold			
Pressure switch	1	1	

Table 3.3: The number of participants who chose each technique as their preferred technique.

3.5 Discussion

Participants were familiar with the tap technique

Table 3.4 summarizes our recommendation for choosing switching technique for indirect multi-touch systems. From the results, the tap technique outperformed or, at least, was comparable with other techniques. The coupling between fingers did not seem to degrade users' performance (low ASI2, ASO1, PSO2, and PSI1). We did not find the absence of tactile awareness influential in the tap technique, even when the locus of attention is away from the fingers (low SOOH).

The hold technique had low slip-outs (DSO, SODH, and SOOH) but high slip-ins (TSI, PSI, and ASI2). This indicates a potential for usage in specific tasks such as keep pressing modifier buttons that put the system into a quasi-mode (like in experiment 3).

The hold technique may be used for dwelling in state 2

Although less problematic when multiple fingers were used, the pressure hold technique should be avoided when using one finger due to fatigue and high slip-outs (DSO).

Avoid the pressure hold technique in single-finger usage.

The performance of the pressure switch technique was in the middle. However, it suffers when users need to reduce pressure in one finger but not the other (PSI1).

	Single finger	**Two fingers**	**Two hands**
Recommended	Tap	Tap	Tap
Suitable when needed to dwell long in state 2 (e.g., keeping a button pressed)	Hold		Hold
Neutral	PSwitch		PHold, PSwitch
Avoid when needed to dwell long in state 1 (e.g., cursor tracking)	Hold	PSwitch, PHold	
Always avoid	PHold	Hold	

Table 3.4: Our recommendation for state-switching techniques

3.6 Limitations and Future Work

The thresholds for the gesture recognizers are specific to our hardware setup. Since our hardware has relatively high input resolution and frame rate, we believe that these parameters are generalizable to other systems. Therefore, we mentioned these thresholds in both standard units (e.g., millimeters instead of pixels) and the percentile rank.

Future work: confirming the thresholds in other hardware setups

While these values may serve as a guideline for other hardware setups, their robustness has yet to be tested.

From a follow-up work: CD ratio is unlikely to influence speed.

To minimize cognitive load from manipulating multiple cursors, we used the 1:1 absolute mapping. However, using different control-display (CD) ratios and aspect ratios may benefit some use cases such as controlling cursors on wall-sized displays [Malik et al., 2005]. Linden [2013][1] compared speed in the steering task on four different CD ratios. The results suggest that the influence of CD ratio to task completion time is unlikely.

Future work: Comparison in the direct pressure sensing touch screens.

In our study, the pressure hold technique yielded unsatisfactory performance. There are three influential factors: exertion, pressure approximation method, and user familiarity. Our results indicates that the finger exertion is the major concern. Less exertion may be possible with a better pressure sensing method, such as one that is independent of contact size. Recently, several touch screen products start to directly register finger pressure for input[2]. This could lead to better user familiarity in the future.

3.7 Reflection

The tap technique differs from the lift-and-tap technique.

In the original paper [Voelker et al., 2013], we used the term *lift-and-tap* technique to refer to the tap technique. The two techniques are different. Assuming that a finger is *on* the screen in state 1, Buxton et al. [1985]'s lift-and-tap technique uses the sequence *on-off-on-off-on* to switch to state 2. After several informal tests, we frequently made mistakes in ourselves. Thus, our tap technique only uses the sequence *on-off-on*. This is a misnomer, which is corrected in this thesis.

Future work: A comparison with the original lift-and-tap

Nevertheless, our results suggested that our participants were familiar with the interaction design and performed well, compared to other candidate techniques. It is unclear whether the lift-and-tap technique would have been better than our technique. The additional lift (on-off) in the lift-and-tap technique makes the gesture more explicit, which could allow users to be even more aware of the state switching, hence lower the slip-ins. However, our tap technique already yielded low in slip-ins (TSI, PSI, ASI2, PSI1). This potential reduction in slip-ins may

[1] A thesis supervised by Simon Voelker, one of the co-author of the original paper.

[2] e.g., in Apple's iPhone 6S (`http://www.apple.com/iphone-6s`)

not matter. On the contrary, the mechanical coupling between fingers may be amplified, and could lead to more slips in the lift-and-tap technique.

Re-analysis supports the claims originally made in the paper.

In the original paper [Voelker et al., 2013], there are several results that we found statistically significant from the p-values. These results mostly concern the differences between the pressure switch and the pressure hold technique. In the re-analysis presented above, statistical estimations indicate that these differences are less probable than before. Nevertheless, none of the claims in the original paper depends on these weak evidences. The results of the re-analysis above still support the conclusions of the original paper.

3.8 Conclusion

In this chapter, we presented a space accuracy problem that is caused by the distance between touch input and visual display in the indirect multi-touch input. To address this problem, we added state 1 to allow cursor tracking. This effectively transformed the space accuracy problem to the state accuracy problem. We then addressed the state accuracy problem by determining suitable state-switching techniques by three user studies. In the next chapter, we look at another state accuracy technique in the midair near the surface of touch screens.

Chapter 4

Eliciting the Thickness of the Near-surface Input Layer

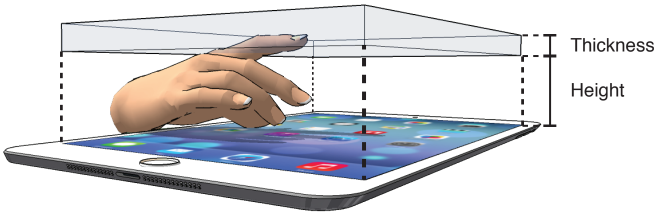


Figure 4.1: A schematic of a near-surface input layer

In addition to on-surface input, recent research proposes augmenting touch input by tracking the fingertip near-surface. This space in midair allows the user to quickly switch between on- and off-surface input. Usage scenarios of near-surface input include controlling cursor speed [Yu et al., 2011], reducing on-screen occlusion [Hilliges et al., 2009], revealing information overlay [Subramanian et al., 2006], and selecting a brush size in a painting application [Desmurget et al., 1997]. Marquardt et al. [2011] provided a survey of how the midair space above the surface is used for input. One of the methods is to divide the space into a stack of discrete layers.

Touch input are augmented by near-surface tracking.

Publications: The work in this chapter was done in collaboration with Kashyap Todi and Marty Pye. Part of this work was published as a full paper at CHI 2014 conference [Wacharamanotham et al., 2014]. The author of this dissertation contributed to the design of the experiments, implementation, and data analysis.

Definition: linebreakemphnear-surface input layer

To enter a near-surface mode, the finger needs to be far enough from the surface (to avoid generating touch signals) and still near enough to be within the sensor range. Previous work proposed dividing the near-surface interaction space to multiple layers [Subramanian et al., 2006, Spindler and Dachselt, 2009]. To have multiple layers, each layer needs to be limited to a certain thickness. In summary, the near-surface input space can be seen as a flat volume of a certain *thickness* placed at a certain height over the touch surface as shown in Figure 4.1. We call this volume a *near-surface input layer*.

The absence of tactile feedback may lead to state inaccuracy.

The thickness of the near-surface input layer has an influence on state accuracy. For example, if the layer is too thin, the absence of tactile feedback during midair hovering causes the finger to drift frequently over and below the layer. The possibility of state inaccuracy is shown in Figure 4.2. To minimize state inaccuracy in near-surface input, we conducted two studies to determine appropriate layer thickness and placement.

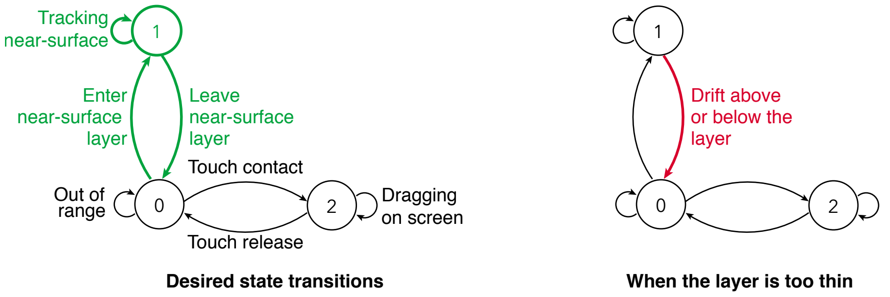


Figure 4.2: Left: A desired state model of near-surface input over a touch screen. Right: Input layers that are too thin cause the user to drift out of state 1.

4.1 Related Work

Although thinner layers are preferred, but they are more difficult to use.

In the same volume, the thinner each input layer is, the more layers can be stacked. Though more layers allow for a richer input channel, interaction with thin layers is more difficult because of physiological tremors, positional drifts in human hands [Brown et al., 2003], and the absence of tactile feedback.

Thickness 4 cm for tangible lens

Several works investigated the thickness of midair layers for different input devices. For example, Spindler et al. [2012] found that users can hold a tangible magic lens can be held still reliably within 1 cm

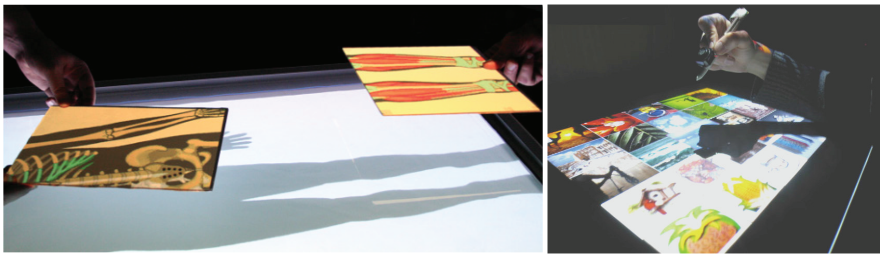


Figure 4.3: Other instruments used in the near-surface layer: tangible magic lens (left) and stylus (right). (Courtesy of [Spindler et al., 2012] and [Subramanian et al., 2006], respectively)

layer thickness (Figure 4.3 (left)). However, to allow users to move the lens horizontally, 4 cm thickness is required for a satisfiable user experience. In their study, visual feedback was provided only when the user left the layer.

Thickness 2–4 cm for stylus

Subramanian et al. [2006] found that 4 cm thickness is also appropriate for stylus input with users' arm resting on a tabletop touch screen (Figure 4.3 (right)). However, in follow-up experiments, Kattinakere et al. [2007] let their participants steer the stylus tip within constrained interactive volumes. The results indicated that the thickness can be reduced to 2 cm when the layer is 0.2 cm above the surface. In both studies, users received a continuous visual indicator of the height of the stylus tip with respect to the layer.

It is unclear whether 4 cm is a suitable thickness for bare fingers.

Will the 4 cm thickness be applicable to the near-surface input with finger? On the one hand, it is possible that all near-surface movements (whether with a fingertip, a stylus, or a tangible magic lens) are controlled by the same muscles in the upper and lower arm. Thus, the thickness should be the same. Chan et al. [2010] created a tilted intangible display system, in which the screen seems to appear in the air, allowing the user to hover, touch, and penetrate the screen. In their study, participants touched the object that appeared on the screen. They found that 80% of the touches were also around 4 cm range from the midair screen. However, in their study, the finger approached the screen from the right angle, and they did not ask the users to move their fingers along the screen plane.

Fingertip movements are more curved than stylus movements.

On the other hand, Desmurget et al. [1997] found that the movement trajectories of of the fingertip are longer and more curved than those of the stylus movements in midair. It is possible that the longer movement of the fingertip may necessitate a thicker layer. We investigate this conundrum in an experimental study below.

4.2 Experiment

Thickness studies:
A: 1, 2, 3, 4 cm
B: 4, 5, 6, 7 cm

To determine the appropriate thickness of near-surface layers, we conducted an experiment comparing the thickness of 1, 2, 3, and 4 cm. After the experiment, we found that the 4 cm thickness yielded the least number of errors and fastest completion time. Therefore, we replicated the experiment to compare the 4 cm layer with thicker layers: 5, 6 and 7 cm. Below, we describe both experiments together. Wherever a distinction is necessary, we will refer to the first experiment as study A and the replication as study B.

4.2.1 Apparatus

We used a marker-based motion tracking.

The hardware setup is shown in Figure 4.4. We tracked the 3D position of the fingertip, wrist, and elbow using eight Vicon Bonita cameras (sub-millimeter accuracy, effective frame rate 100 Hz). On each of the tracked body parts, we attached a lightweight patch (<8 g) with a unique constellation of three retroreflective markers. Visual output was shown on an Apple Cinema Display (49.5 cm × 30.5 cm; 1920 × 1200 pixels), placed approximately 50 cm away from the user.

4.2.2 Participants

8 participants for each experiment.

We recruited participants from our campus. In each experiment, we had eight participants (6 males and 2 females). The average age of the participants in study A was 24 years, in study B 26 years. All participants had a computer science background, normal or corrected sight, and no motor impairments.

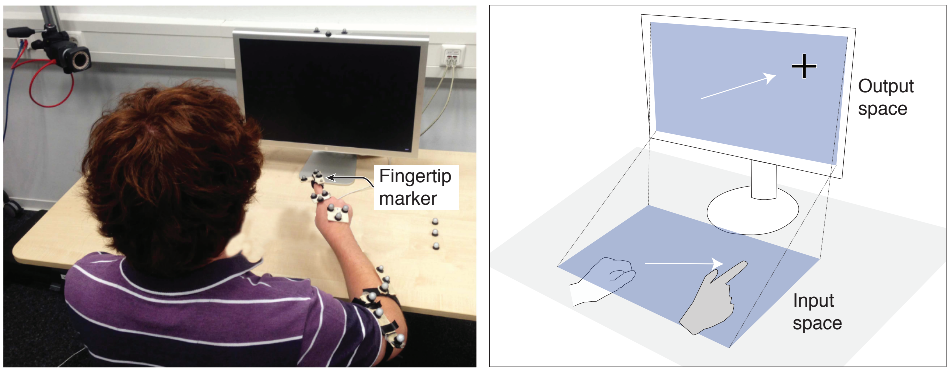


Figure 4.4: The hardware setup for our user studies (left), and the cursor mapping (right)[1].

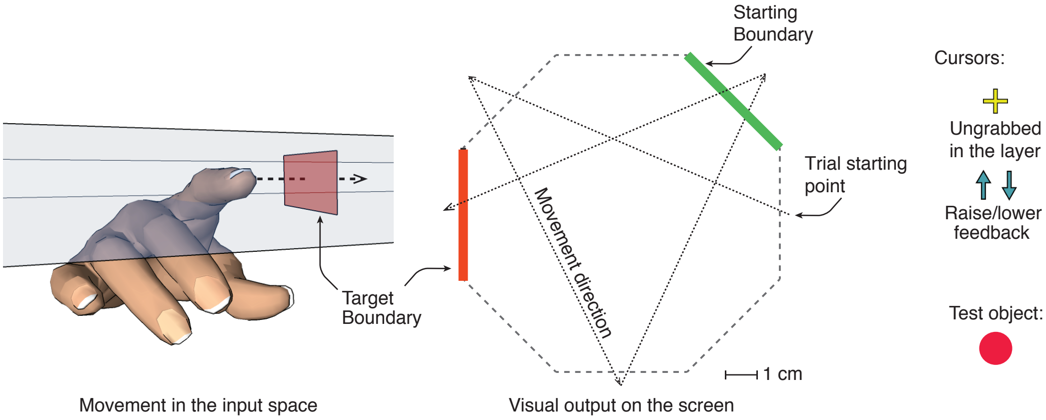


Figure 4.5: Task in the input space (left), on the screen (center) and cursors (right) [1]

4.2.3 Task

Task: crossing a goal in midair while maintaining the finger in the layer.

Our task is based on the 1D tunneling task by Kattinakere et al. [2007] (constraining only the height). On the screen, the targets were visualized in 2D as shown in Figure 4.5 (right)[1]. The participants were asked to move a test object (red circle) from the starting boundary (green bar) to a target boundary (red bar). To move the cursor, users had to dwell the index finger in the near-surface layer (Figure 4.5 left). The position of the cursor corresponds to the orthogonal projection of the fingertip onto the surface. Moving the cursor back into the test object automatically grabs the object. Leaving the near-surface layer drops the object, and the users have to move the cursor into the object to re-grab. We instructed the users to be as quickly and as accurately as possible.

[1] Images are modified from [Wacharamanotham et al., 2014].

4.2.4 Design of the Experiment

4 *Thicknesses* ×

In addition to the *Thickness* levels (study A: 1,2,3,4 cm; study B: 4, 5, 6, 7 cm), we manipulated two additional independent variables in study A: Firstly, we controlled whether the users were allowed to rest their palm on the surface or not (*SurfaceSupport*). Having support from the surface could increase finger stability because less muscle activation is needed to hold the hand. However, resting the palm on the surface may generate spurious touch input on the screen. We placed the layer at 0.25 cm above the surface for the support condition, and at 8 cm for the midair condition.

2 *SurfaceSupports* ×

2 *InputDistances* ×

Secondly, we controlled the *InputDistance*, which is the distance necessary to move the cursor from the starting boundary to the target boundary. This distance is in the input space and measured in 2D on the plane of the desk. We tested two distances: long (10 cm) and short (1 cm). The long movement condition represents a cursor tracking use case, while the short movement condition represents a near-surface gesture shortcut such as selecting an item on a marking menu [Bailly et al., 2011]. While completing the task in the short distance is possible by using the wrist movement only, the long distance needs movement in the lower arm. In both *InputDistance* conditions, the users could see the same visual output on the screen. For the short *InputDistance* (1 cm), we adjusted the control-display ratio to be 1:10 (moving the finger 1 cm will move the cursor 10 cm).

8 targets ×
4 repetitions
= 3072 trials per user
in study A

There were four repetitions. In each, participants crossed eight targets, placed around the center of the screen. For study B, we tested only the *Thicknesses* levels with long *InputDistance* and no *SurfaceSupports*.

4.2.5 Dependent Variables

Definition: *DriftCount*

To operationalize the fitness of each thickness level, we counted how many times the finger drifted below or over the midair layer (*DriftCount*). In order to aim for low *DriftCount*, users were able to trade speed for accuracy, by moving their fingers more carefully than normal.

Definition: *TotalTime*

Thus, we also measured *TotalTime*: the duration that the user manipulates the object from the starting boundary to the target boundary. Lower *TotalTime* indicates that the user moves within the layer with confidence.

4.2.6 Data Analysis

We removed data from one participant from study B due to extreme spurious movements (three times longer duration and finger movement distance than the rest).

Data aggregation: Sum(*DriftCount*), Mean(Log(*TotalTime*))

A suitable thickness should allow users to perform well in all directions. Therefore, we summarize data across all targets by summing *DriftCount*. For *TotalTime*, we log-transformed each measurement before averaging all across targets. (The results below were anti-logged.)

Bootstrap procedure for calculating means

To calculate means, we used ordinary non-parametric bootstrapping (10,000 replicates). CIs were calculated with the bias-corrected and accelerated method (BCa). Some confidence intervals are asymmetric, which reflects the distribution of the data. All error bars are 95% CIs.

Bootstrap procedure for calculating the differences between means

To calculate the differences between means, we used the 4 cm thickness as the baseline. (As mentioned above, the 4 cm thickness was recommended to be used with both stylus and tangible magic lens.) The same bootstrap process was used to sample within-subjects differences (grouped appropriately by *InputDistance* and *SurfaceSupport*).

4.2.7 Results

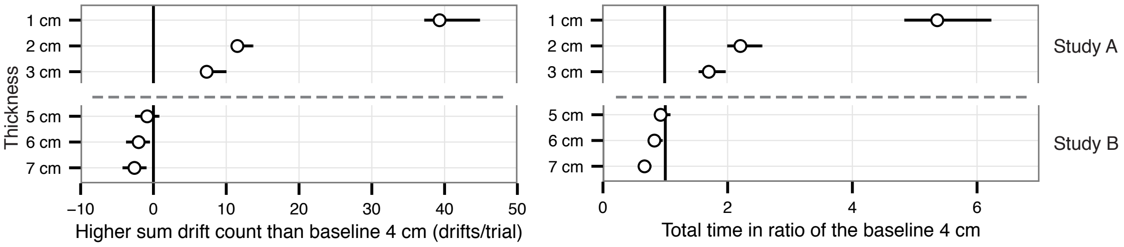
We first analyze *InputDistance* = large *SurfaceSupport* = no

We first describe the results that are common from both study A and study B (large (10 cm) *InputDistance* with no *SurfaceSupport*). Figure 4.6 (on page 71) shows descriptive statistics for all layer thicknesses. As expected, there is a tendency that both dependent variables were lower in thicker layers, but the benefit was diminishing.

DriftCount in the 1 cm thickness was outstandingly worse.

The within-subjects differences of *DriftCount* from the baseline (*Thickness* = 4 cm) are shown in the left charts below. In the 1 cm thickness, the confidence interval of *DriftCount* is farther right from zero than in other thickness levels. This is a strong evidence that the participants tended to make much more errors. In thicker layers, *DriftCount* is closer to the baseline. The 2 cm and 3 cm thicknesses still yielded higher *DriftCount*, but they were much lower compared to the 1 cm. Adding thickness beyond 4 cm seems to improve *DriftCount*. However, the differences seem to be smaller than between 3 cm and 4 cm.

Layers thicker than 4 cm benefit less.



TotalTime decreases for thicker layers. The reduction tapered off around the 4 cm thickness.

For *TotalTime*, the right charts above shows the within-subjects differences. This plot is in ratio with respect to the baseline (*Thickness* = 4 cm). For example, in the 1 cm layer, participants took around 3–4 times the duration they took in the 4 cm layer. Similar to the *DriftCount*, our participants were faster in the thicker layers. The differences among the thickness levels above 4 cm are much smaller than those below 4 cm.

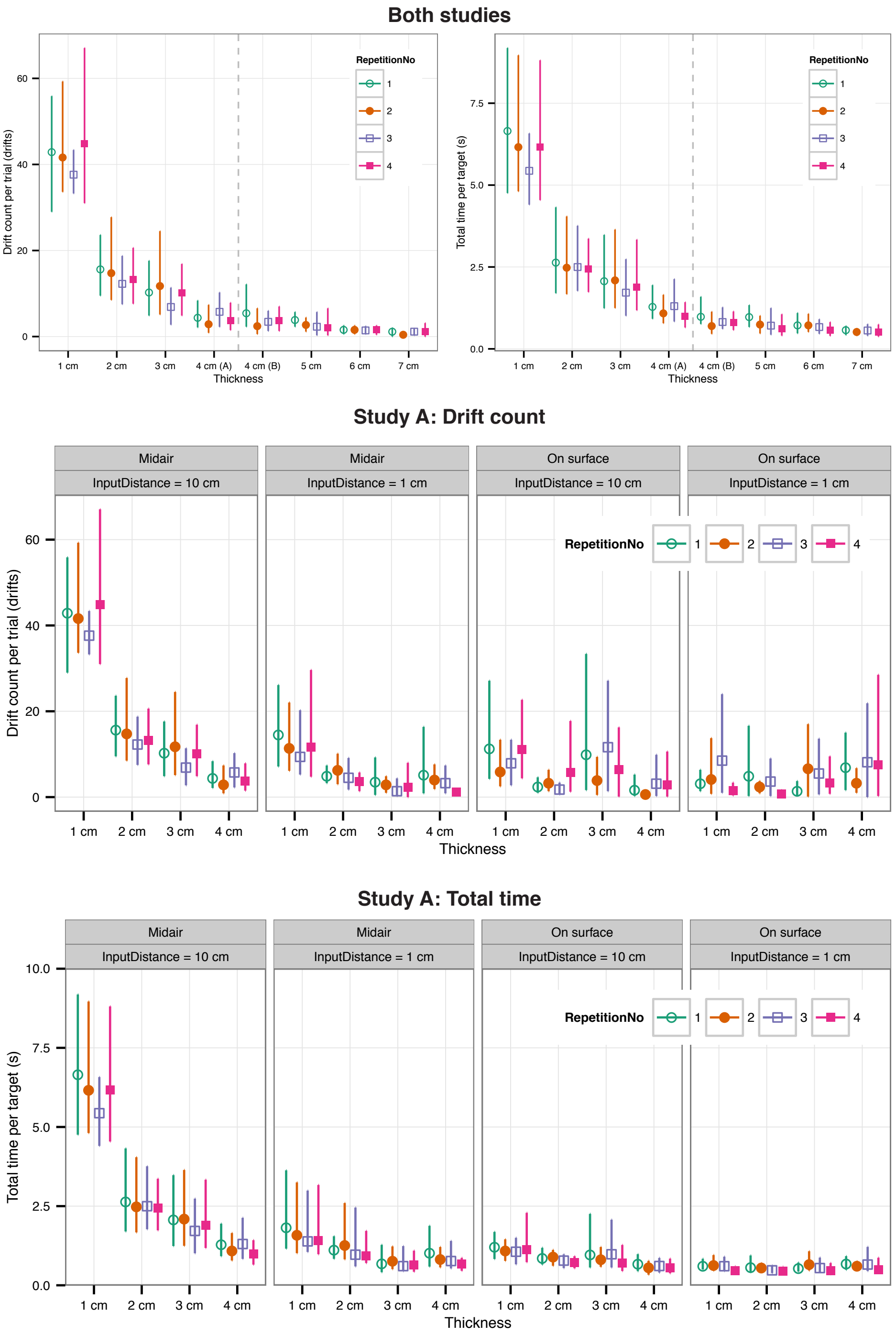


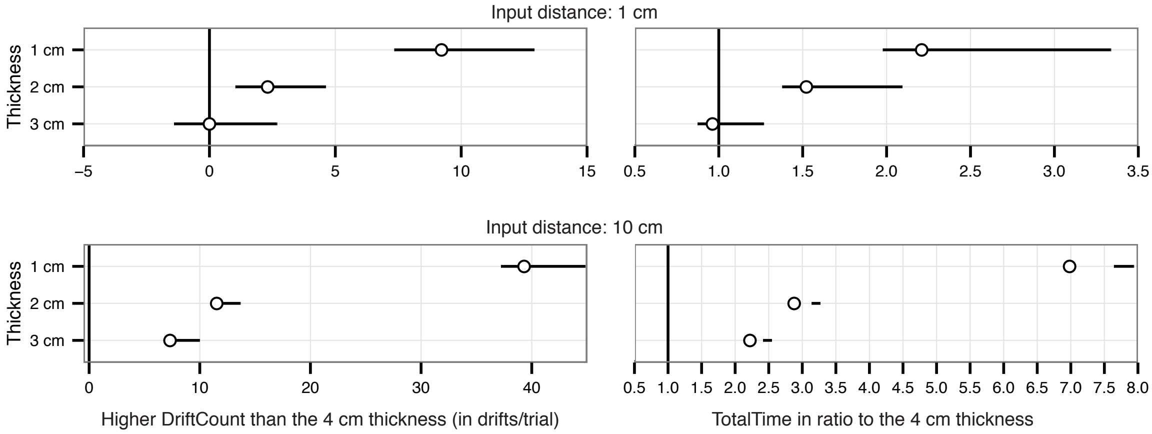
Figure 4.6: The descriptive statistics of *DriftCount* and *TotalTime* by *Thickness*. Top: collective results of study A and B (large *InputDistance* with no *SurfaceSupport*). In each plot, the results from study A is to the left of the dashed line, and from the study B to the right. Middle and bottom: The results from study B, broken down by *InputDistance* and *SurfaceSupport*. Each point indicates the mean across participants.

The effect of *InputDistance* × *SurfaceSupport*

We now focus on study A, which involves two *InputDistance* and two *SurfaceSupport*. The middle panels of Figure 4.6 show the descriptive statistics of the *DriftCount*. The charts below show the within-subjects differences for the midair condition. The baseline for each graph is the 4 cm thickness in the same *InputDistance* conditions. Some of the point estimates are outside the CI due to the bias correction in the BCa method. Interpretations should focus on the CIs.

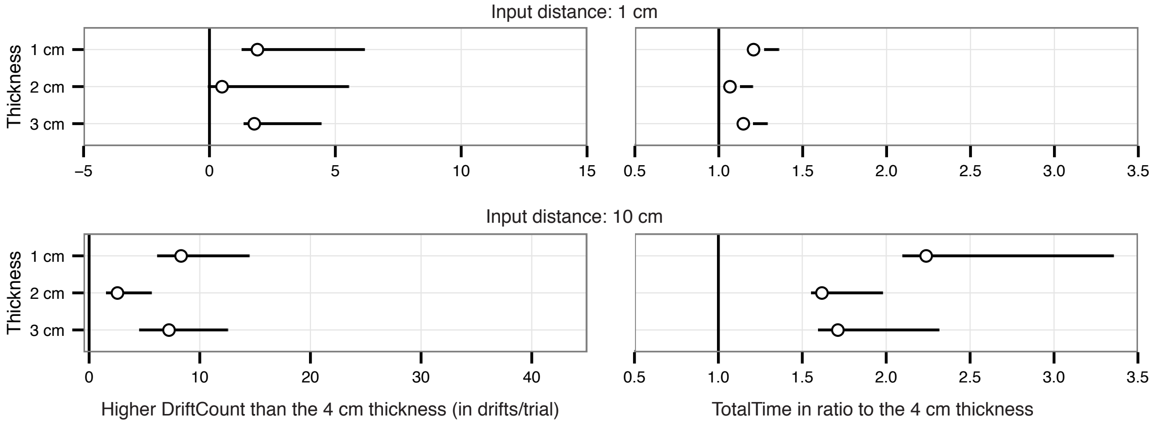
The 3 cm thickness is feasible for short movements in midair.

For the short movement distance (1 cm), the 3 cm thickness seems to yield similar *DriftCount* and *TotalTime*. However, this similarity diminished in the longer movement distance (10 cm).



The surface support reduces the differences across conditions.

The charts below show the differences for the condition with surface support. The effect of thicknesses are smaller than those in midair. However, the 4 cm thickness seems to be constantly superior than other thicknesses.



4.2.8 Discussion

Thicker layers have lower *DriftCount* and *TotalTime* than thinner layers. However, the reduction in both dependent variables is much smaller when the thickness is beyond 4 cm. This suggests that making layers thicker will not have much benefit. Therefore, we recommend 4 cm as a general thickness level, which is applicable for longer movement (10 cm) without resting the arms on the surface.

General recommendation for thickness: 4 cm

However, for the interaction in short input distances, the 2 cm thickness is likely to perform as well. For example, when the near-surface space is used for invoking a marking menu. Once the menu appears, the user just needs to move the finger in a short distance to select an item.

Special case: short distance movements may use 2 cm.

Our results agree with the thicknesses in the literature for other devices (stylus and tangible magic lens). In particular, our results agree with the findings from Spindler et al. [2012] and Subramanian et al. [2006]. However, Kattinakere et al. [2007] recommended a thinner layer (2 cm). This is likely due to the continuous visual indicator of the height from the surface. (Our studies and that of Spindler et al. [2012] provided visual feedback only when the user drifted outside the layer.) From these consistent findings, it is possible that the 4 cm thickness may be generalizable to other hand-held devices.

Our results agree with the findings in stylus and tangible magic lens.

4.3 Limitations and future work

We had a small number of participants with a bias towards male users. Since the ability to maintain finger position depends on muscle strength, it is possible that the gender of users influences their performance. Nevertheless, we do not have adequate evidence from our data to indicate such difference.

Future work: The influences of physical strength

Previous findings by Kattinakere et al. [2007] indicate that it is possible to use thinner layers when they are placed nearer (0.2 cm) to the surface. Although we tested two layer heights (0.25 and 8 cm), these heights were used to control whether the users could rest their hand on the surface or not. A lower height may potentially reduce muscle load, and allow users to stay stable in thinner layers. Nevertheless, placing the near-surface input layer too close to the surface may result

Future work: Placing layer nearer to the surface.

in the ambiguity between on-surface versus near-surface state. In particular, for capacitive touch screens, there is an electrical field outside of the screen for touch detection. A finger hovering very close may be registered as a touch. Further studies as well as sensor development are needed to investigate this balance.

4.4 Reflection

Statistical estimation improves awareness of data anomaly

The conclusion of the re-analysis agrees with the published paper, [Wacharamanotham et al., 2014]. During the preparation of the paper, we used null-hypothesis significance tests (NHSTs). In the re-analysis presented in this chapter, we used bootstrapped statistical estimation. This method forced me to be more aware of the effect size in each step of inference. This revealed data anomaly in study B, which was further diagnosed and removed from the analysis, as mentioned in section 4.2.6.

Enforced iterations on data visualization simplifies plots of the results.

To interpret statistical estimates and confidence intervals, many plots were created. Choosing the parameters of the plot (abscissa, ordinate, shapes, and facets) to communicate the results is more demanding than in the previous analysis. When NHSTs were used, we treated the plots as an auxiliary evidence: The main interpretations were based on statistical significance. In contrast, in the re-analysis, appropriate plot parameters are necessary to reach any interpretation as well. This enforcement results in graphics that are much simpler to understand than the original paper.

> *"Numerical quantities focus on expected values, graphical summaries on unexpected values."* —John Tukey (1915–2000)

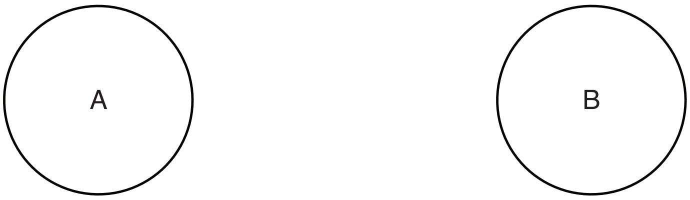
4.5 Conclusion

In this chapter, we took a look at the state inaccuracy in the near-surface input layers that are too thin. To address this problem, we determined the appropriate thickness that allows users' finger to stably dwell inside. In the next chapter, we discuss a use of the near-surface layer for haptic feedback.

Chapter 5

Reducing Drifts with Haptic Feedback

To start off this chapter, I would like to invite the reader for an experiment. First, place this page flat on a firm surface, e.g., your desk. Then, imagine that the two circles, A and B, below are two buttons on a touch screen. With the index finger of the your dominant hand, try pressing these two buttons alternatively (A, B, A, B, ...) for 20 times each with the eyes closed. On the last touch, keep your finger in contact with the page, and look where it touches.



Publications: The work in this chapter was done in collaboration with Malte Weiss and Simon Voelker. Part of this work was published as a short paper and a demo at UIST 2011 conference [Weiss et al., 2011]. The author of this dissertation contributed to the design of the experiments, data analysis, and implementation of the demo.

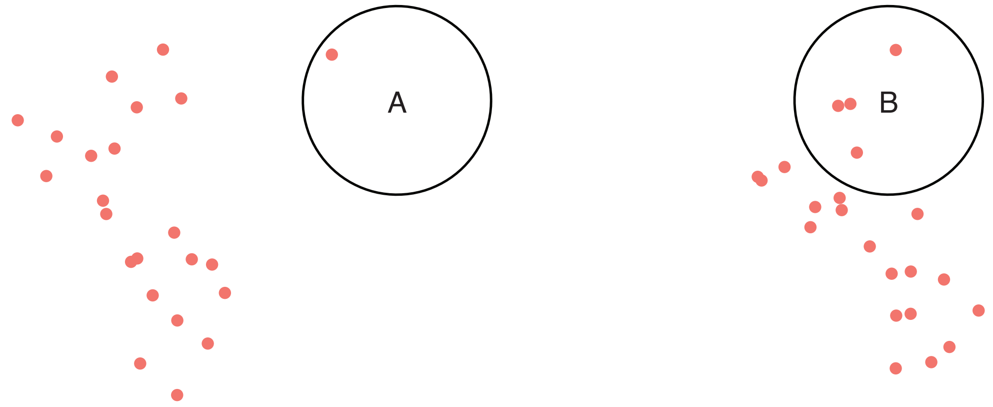


Figure 5.1: A possible trace of contact points. Without visual awareness of the targets and the finger position, drifts occur.

Without visual feedback, touches gradually drift away from the targets

Where was your last touch? It would be no big surprise if it drifted away from the buttons. A plot showing your contact points may be similar to what is shown in Figure 5.1. In fact, this phenomenon was studied by Brown et al. [2003]. They asked participants to alternate their touch between a starting point and a target on a flat surface for 75 repetitions. After the first 5 repetitions, they took away visual feedback of the finger and the target. On average, the participants' finger drifted 8 centimeters away from the initial position. They surmised that drifts accumulate because human positional control system depends on visual feedback of limb position.

Drifting occurs when using touch screens eyes-free.

The drifting touch problem happens on touch screens when the visual attention is demanded elsewhere. For example, when transcribing text using a touch screen keyboard. In a study conducted by [Findlater et al., 2011], expert typists were assigned to transcribe text on a touch screen with and without showing visual layout of the keyboard. The contact points in the invisible condition deviated further than when the keyboard was visible. In summary, using touch screen eyes-free reduces space accuracy.

The differences between this chapter and previous publications

In this chapter, we describe FingerFlux, a prototypical system that substitutes visual feedback with haptic feedback from electromagnetic actuation. Weiss [2012], a co-author of FingerFlux, already published an extended discussion on FingerFlux in his dissertation, which is summarized in the next section. The remainder of this chapter expands the data analysis concerning drift reductions.

5.1 FingerFlux Recap

To provide context for the rest of this chapter, this section summarizes works that add haptic feedback on touch screens. Then, we briefly summarize the FingerFlux prototype and two user studies.

5.1.1 Haptic Feedback On and Over Touch Screens

Approaches in providing haptic feedback:
1. on-screen

Haptic feedback on touch screens can be provided from three locations: from the screen itself upon contact, from a physical device worn by users, or from a projection of feedback directly onto users' hands. In the first category, a strong feedback from the screen can be provided by vibrating the entire screen [Fukumoto and Sugimura, 2001]. TeslaTouch provides a more subtle feedback by changing friction between users' skin and the screen with electrovibration [Bau et al., 2010]. It is also possible to change stiffness of the screen itself. Jansen et al. [2010] used a latex pouch as a screen. The pouch is filled with magenetorheologic fluid which can be manipulated by an electromagnetic field.

2. user instrumentation and hand-held devices

In the second category, haptic feedback may be generated by a physical device that is held or worn by users. In the Haptic Tabletop Puck, the feedback is provided through a tangible enclosing a rod that can mechanically change its height according to the content on the screen below the puck [Marquardt et al., 2009]. Madgets tabletop uses electromagnetic actuation to control tangible widgets on touch screens [Weiss et al., 2010a]. In Senseable Rays, Rekimoto [2009] used an actuator worn on fingernails to provide tactile feedback. Another approach is to radiate laser through a thin elastic tape worn on users' fingers to create tactile sensation Lee et al. [2015]. The Senseable Rays and the laser radiation can be appropriated to provide haptic feedback prior to touching the screen.

3. projection of feedback onto users' hands

In the third category, feedback is projected by, e.g., an array of air jets [Suzuki and Kobayashi, 2005] or ultrasound emitters [Hoshi et al., 2010, Monnai et al., 2014]. However, these technologies are yet to be mature enough to apply to touch screens. They require placing haptic projectors at the same side of the screen to project the haptic feedback to users' finger pads. This would sacrifice the quality of either the haptic or visual output of touch screens.

5.1.2 FingerFlux Operating Principles and the Prototype

FingerFlux uses electromagnets below the touch screen to actuate permanent magnets attached to fingertips.

FingerFlux consists of an array of electromagnets, situated below a touch screen (Figure 5.2), and small permanent magnets attached to users' fingers. To provide haptic feedback for a finger that is near the screen surface, FingerFlux changes the polarization of each electromagnet. This creates a magnetic force field that attracts, repels, or sends vibrations to the permanent magnets. As a result, FingerFlux can softly push or pull the fingertip in certain directions. It can also create a feeling of bumps or vibration when the fingertip is near the surface, e.g., about to press a dangerous button.

A haptic feedback scheme for two buttons.

Figure 5.3 shows an electromagnetic rendering scheme for two buttons. The permanent magnet at the fingertip has a negative pole (-) facing the surface. To attract the finger towards a button, the center of the button is rendered with the opposite polarity (+). Meanwhile, the area around the button is rendered with the same polarity (-), repulsing the fingertip away from hitting on the area outside the buttons. We used this rendering scheme to investigate how haptic feedback helps reducing drifts.

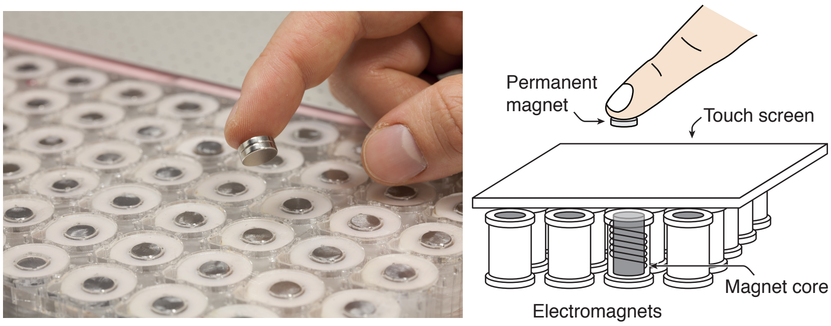


Figure 5.2: The FingerFlux prototype. An array of electromagnets, placed below a touch screen. Powering these electromagnets creates a magnetic force field that attracts or repels the permanent magnet at the fingertip. (Left image courtesy of [Weiss et al., 2011])

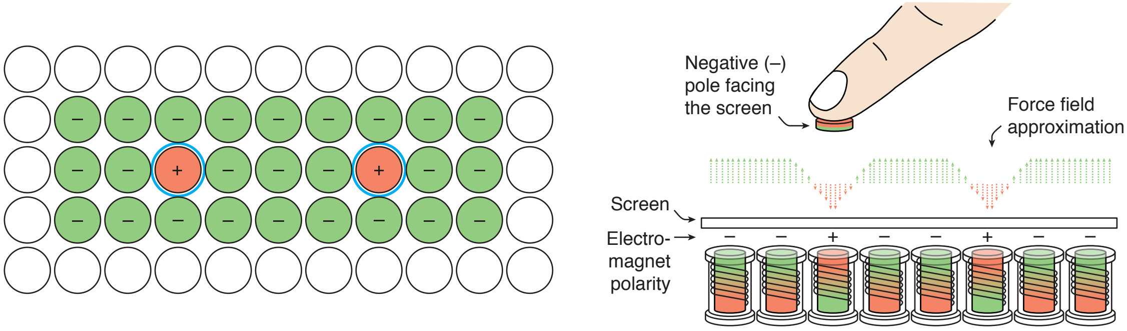


Figure 5.3: Left: A haptic feedback rendering scheme for two buttons (blue circles). Right: a cross-sectional view of the button row. The center of the buttons attracts the fingertip, while the area outside repulses. (Modified images from [Weiss et al., 2011].)

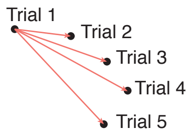
5.2 Experiment: Reducing Drifts with Haptic Feedback

To evaluate whether FingerFlux reduces drifts, we used a *reaching task* [Brown et al., 2003, Gordon et al., 1994]. The participants were asked to press two targets alternately on the screen. Each trial is a press on the left target. In the first seven trials, the participants were allowed to see the targets and were asked to memorize their movements. Afterwards, they closed their eyes and continued for 22 trials.

Task: tapping two targets alternately. Seven trials with vision and 22 trials without

During the eyes-free trials, the contact points on the left target drifted from the original position. Each time, the subsequent contact point drifted slightly from the previous one. These small drifts may accumulate over time, resulting in a large deviation from the first contact point. To quantify this deviation, Brown et al. [2003] defined *cumulative drift* as the Euclidean distance between the initial contact point and the current contact point. The cumulative drift is our main dependent variable.

Dependent variable: *cumulative drift*



In their study, Brown et al. [2003] found that cumulative drifts increase over trials when the vision is absent. We surmised that the presence of haptic feedback from FingerFlux would reduce the cumulative drifts. All participants tested both conditions, the order was counterbalanced. There were ten participants (age: 23–29; two females; all right-handed).

Independent variable with vs. without haptic feedback

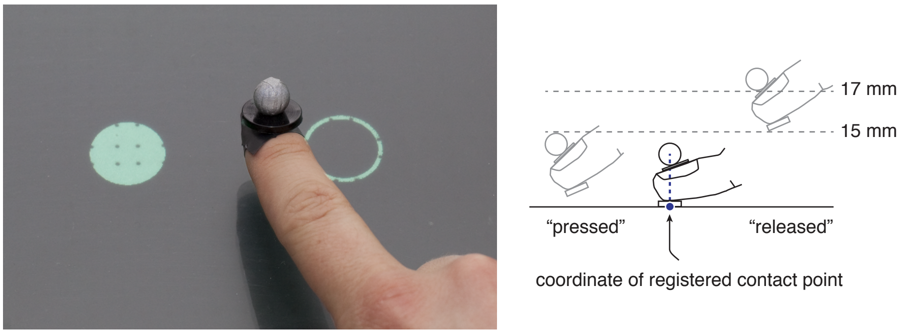


Figure 5.4: Hardware setup for the user study. Left: the marker setup. Right: The touch recognition thresholding.
(Modified images from [Weiss et al., 2011].)

5.2.1 Apparatus

A marker and two permanent magnets were attached to the fingertip

All participants used the index finger of the dominant hand. They wore a retro-reflective marker on their fingernail and two cylindrical neodymium magnets (each has 10 mm diameter, 2 mm height) on the finger pad.

Motion tracking system with hysteresis thresholding.

For the precision in position tracking, we used eight Vicon Bonita cameras to track the marker on the users' fingers. As shown in Figure 5.4, a "press" event is fired when the marker is closer than 15 mm from the surface, and a "release" event is fired when the marker is lifted higher than 17 mm from the surface. The contact point is the orthogonal projection of the marker on to the surface when the marker is at its lowest position. The press event of the first trial is used to represent the center of the button according to the mental model of each user.

Madgets table was used.

We used Madgets Table from [Weiss et al., 2010a]. The magnetic actuation scheme is shown in Figure 5.3. Each magnet was driven with full power (40 V): A force of 0.8 N is needed to lift a neodymium permanent magnet (10 mm diameter, 2 mm height) from the surface [Weiss, 2012].

The appearance of the buttons

Two circular buttons with a diameter of 25 mm were displayed on the surface. The center of each button was at the center of the electromagnets. The gap between the two buttons (measured from their circumferences) is 40 mm (two electromagnets). Both of buttons were drawn in outline, and filled when they were pressed.

5.2.2 Data Analysis

Due to errors in the motion tracking system, we removed 13 trials from our analysis. In these trials, the contact points were reported more than 10 cm away from the test area.

Bootstrapping was used in analysis.

To summarize the data of in each trial, we calculated mean and 95% CI of the data from all participants. We used ordinary non-parametric bootstrapping (10,000 replicates) to calculate the means, and we used the bias-corrected and accelerated method (BCa) to calculate the CIs. All error bars are 95% CIs.

Trial groups: *vision*, *early* no-vision, and *late* no-vision

In the analysis below, we also grouped trials in ranges and compared cumulative drifts among them. We excluded the first and the last trial to prevent the effect of onset and ending. For the same reason, we excluded trial 8, which is the first trial when the vision was absent. From the rest of the trials, we drew three ranges: *vision* (trial 2–7), *early* no-vision (trial 9–14), *late* no-vision (trial 23–28). When we compared these ranges, we first calculated the difference between matched pairs of trials (e.g., trial 2 vs. 9, 3 vs. 10, etc.) within each participant. The differences were then averaged for each participant. The final mean and CIs are calculated by bootstrapping from these averages.

Additional analyses: drift accumulation rate and drift direction

In addition to the cumulative drifts which were reported in [Weiss et al., 2011, Weiss, 2012], we performed an exploratory data analysis on the *drift accumulation rate* (first derivative of cumulative drift with respect to trial number) and the *drift direction*. For the direction, we used circular statistics to calculate means and SDs. Hence, 359° and 0° were treated as closer to each other than 359° and 350°.

5.2.3 Result

The haptic feedback reduces average distances.

Figure 5.5 shows the contact points on the left target from each participant. The average direction and distance of each participant is shown in the inset of Figure 5.5. For all participants, haptic feedback reduced the average distance of contact points from the original position.

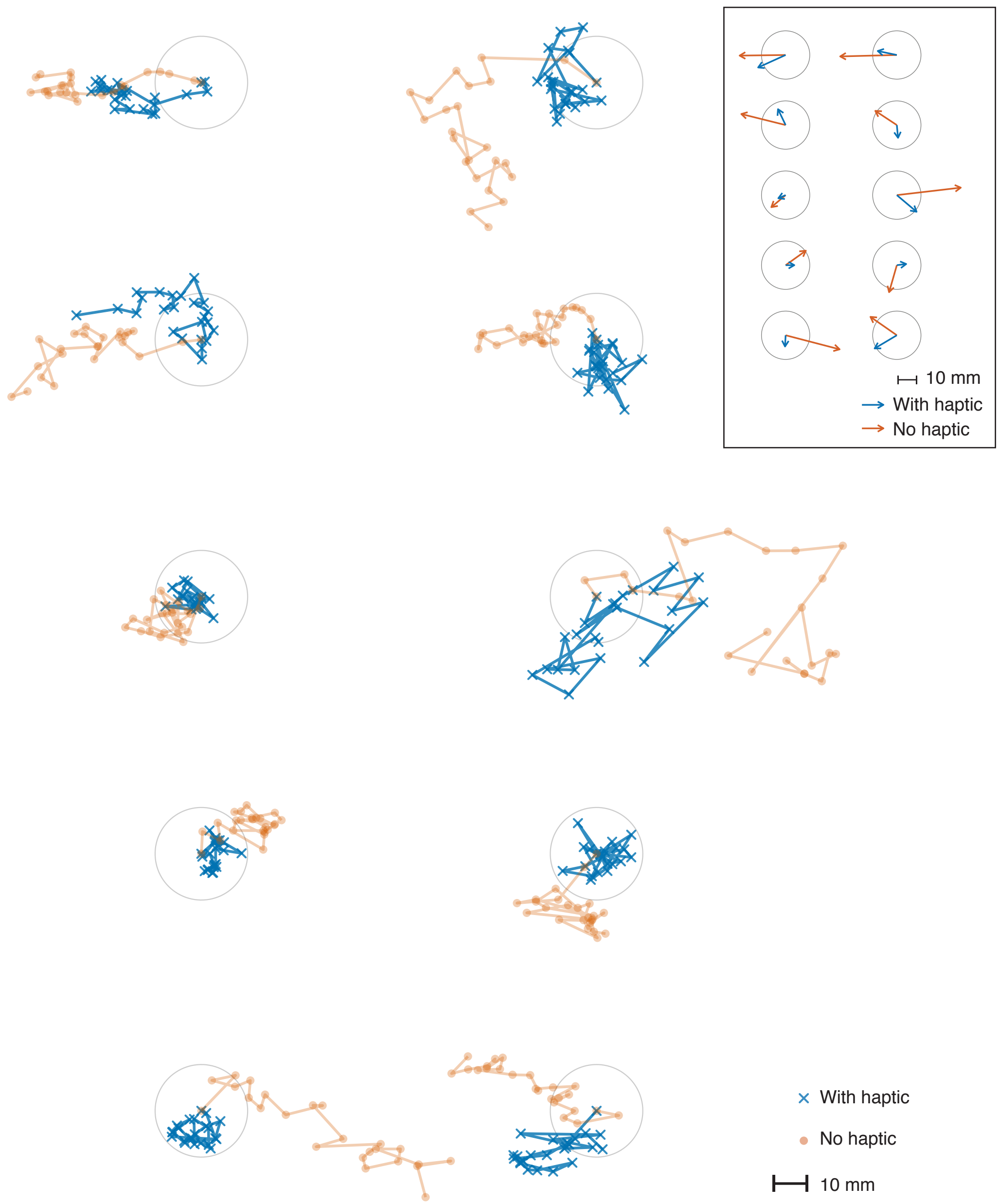


Figure 5.5: Contact points from each of the ten participants. The haptic feedback apparently reduces drifts. Only left targets were shown. Subsequent trials are connected with a line. The circle represents the button that is visible to the participants. The first trial always started at the center of the button. The inset shows the average direction and distance of these contact points.

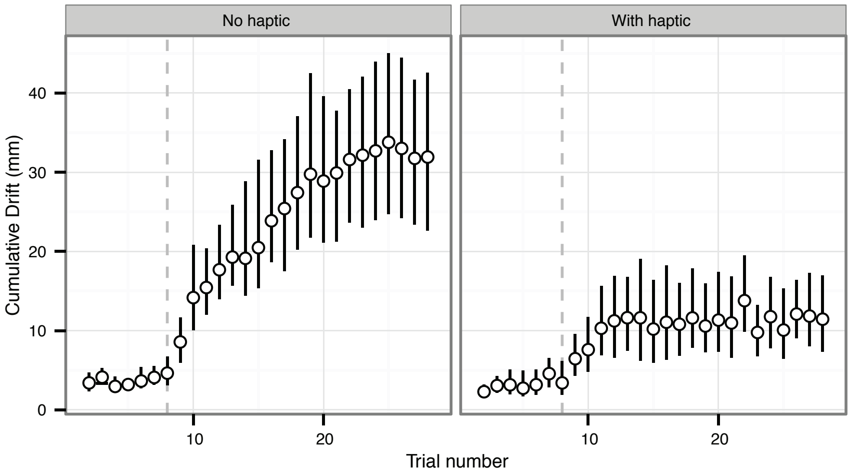


Figure 5.6: Cumulative drift across trials in each condition. The dashed line indicates the first trial without vision. Each point is the mean across users. Each error bar is the 95% CI of mean.

The haptic feedback reduces cumulative drifts.

As shown in Figure 5.6, the cumulative drifts (the distance from the first contact point) increased over trials for both conditions. There seems to be a sharp increase immediately after the vision was absent. The drifts stabilize afterwards. However, when haptic feedback was provided, the cumulative drift appears to be shorter and accumulated slower. In Figure 5.7, we compared the *vision* range (trial 2–7) with the *late* no-vision range (trial 23–28). We used the vision range of the non-haptic condition as the baseline (zero). In the late range of the non-haptic condition, the value is on the right of zero, indicating that this condition yielded larger drifts than the baseline. Between the two late ranges, the large difference between the haptic and non-haptic conditions is a strong evidence that haptic feedback reduces cumulative drifts when operating eyes-free. However, the late haptic range still has higher drift than the vision ranges. This indicates that our haptic feedback is still not a full substitute for vision.

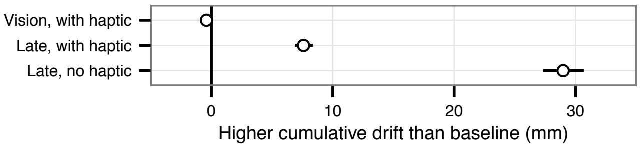


Figure 5.7: The difference in cumulative drifts among trial ranges. The haptic feedback reduces the drift in the late trial range. The baseline is the vision range without haptic feedback.

Haptic feedback slows down drift accumulation.

To analyze how fast drifts accumulated, we compared the slope of the cumulative drift. Higher slopes indicate that drifts accumulate quickly. Figure 5.8 shows the within-subjects differences of the slope in the trial ranges. *Early* after the vision was taken out (trial 9–14), the slopes were higher in both conditions. However, between the two haptic conditions, the slope was lower when haptic feedback was provided (1.04 mm/trial 95% CI [0.32, 1.79]). This suggests that haptic feedback causes the drift to accumulate slower. In the late period, drifts accumulation rate stabilized in both conditions. This suggests that the participants adapted to the influence of haptic feedback and produced stable drift distances.

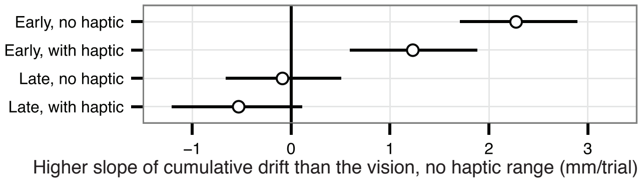


Figure 5.8: The differences in the rate of change in cumulative drift. Drifts accumulate slower in the haptic condition.

Haptic feedback consistently changes the drift direction to compensate the natural cumulative drifts.

In terms of direction that the contact points drift, Figure 5.9 shows the standard deviation (circular SD) of cumulative drift directions (calculated from a sliding window of five trials). The SD is high when the direction changes frequently. Otherwise, it is low. Without haptic feedback, the SDs are consistently low from trial 15 onward, indicating that the users drifted in a consistent direction after the vision was removed. In the same trial numbers, the haptic condition yielded relatively higher SDs. This suggested that haptic feedback helped the participants to consistently change the direction of drift, hence compensating the natural cumulative drifts.

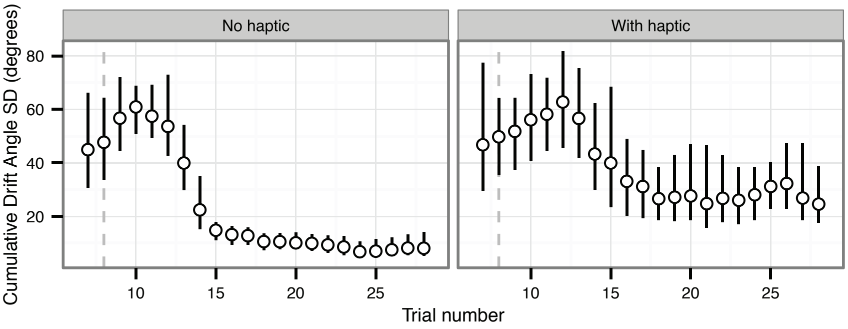


Figure 5.9: The SD of cumulative drift direction across trials. The low SD in the non-haptic condition indicates a systematic drift direction.

5.2.4 Discussion

Our results support the existence of drifts in eyes-free input.

Our results in the non-haptic condition confirm those reported in [Brown et al., 2003]. Taking out vision causes the contact points to drift, and these drifts accumulate over time (Figure 5.6) in a consistent direction (Figure 5.9). Nevertheless, this is unsurprising because the length of the gap (40 mm in ours; 150 mm in [Brown et al., 2003]) influences the cumulative drift distance [Gordon et al., 1994]. Our participants accumulate drifts at a similar speed as in [Brown et al., 2003]. The speed was close to 3 mm/trial in the early trials and close to 0 mm/trial in the late trials.

FingerFlux reduces drifts in eyes-free touches.

By introducing the haptic feedback with electromagnetic force field, the cumulative drifts distance was reduced (Figure 5.7) and the rate of accumulation was slowed down (Figure 5.8). Supposed that a system similar to FingerFlux will be used in a touch screen. Upon detecting that the user is not looking at the screen, such system can activate haptic feedback to reduce drifts.

The haptic feedback alone cannot replace vision.

Nevertheless, there are still higher drifts on a touch screen with haptic feedback than when used with vision. Further UI adaptation may be necessary. For example, enlarging the button sizes during eyes-free usage. According to Figure 5.7, without FingerFlux, these buttons should be enlarged from the current diameter 25 mm to 55 mm. In our configuration, this enlargement would already make two buttons overlap. With FingerFlux, however, the button can be only around 10 mm wider, leaving the two buttons separated.

Other factors: proprioception, ballistic movement force

According to Figure 5.6 drifts seems to be quickly stabilized (around trial 25 without haptic and trial 12 with haptic). The reason for this stabilization could be the proprioception, which determines the upper bound of the drift distance. In our experiment the buttons were 40 mm apart. On the one hand, drift reduction may be accomplished by placing the buttons closer to each other. For example, when the distance requires moving only the metacarpalphalangeal joint. In such situation, the benefit of the haptic feedback may be less pronounced. on the other hand, placing the buttons further away may risk reducing the benefit from the haptic feedback. For example, when the force exerted by the arm during the ballistic movement is far greater than the electromagnetic force.

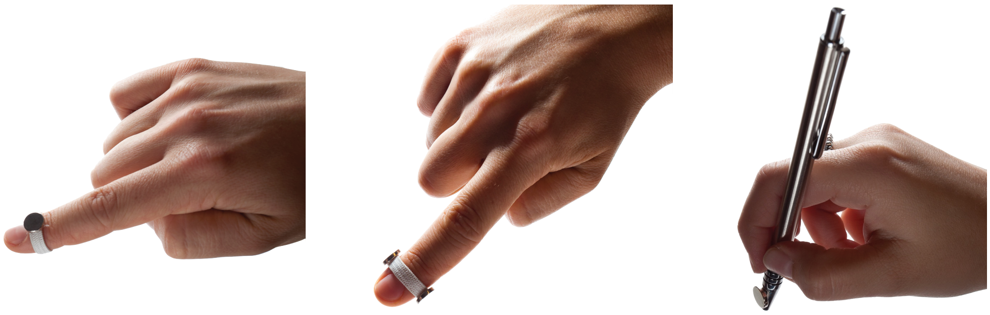


Figure 5.10: Alternative placements of the permanent magnets. (Courtesy of [Weiss, 2012].)

5.3 Limitations and Future Work

Explore different strengths of the force field.

The attraction and repulsion forces in FingerFlux depend on both the electromagnetic array below the screen and the permanent magnet attached to the fingertip. We have explored only a single parameter set, which is specific to our hardware setup. Further studies could investigate how increasing the strength of the magnetic force field (e.g., by increasing the voltage) would influence the quality of haptic feedback.

Explore different placements and associated actuation scheme.

Having a permanent magnet attached to the fingertip can be cumbersome. Different placements of the magnet (e.g., in Figure 5.10) are possible to leave the finger pad exposed. Different electromagnetic actuation schemes may be required for these alternative placements, especially when the face of the permanent magnet is not parallel to the touch screen.

5.4 Reflection

Clearer quantification of drifts.

In the original paper [Weiss et al., 2011], visual evidences similar to Figure 5.5 and 5.6 were used to substantiate the drift reduction. In this chapter, we compared the differences by focusing on specific ranges of trials. As a result, the differences are interpretable in terms of button sizes (Figure 5.7), enabling us to relate the discussion to design implication (e.g., button size). Additional findings from exploratory analysis (the rate of drift accumulation and the changes in drift directions) further support the results we previously reported.

5.5 Conclusion

In this chapter, we described drifts that occur when using touch screen without vision. This is a space accuracy problem. To address this problem, we created a system that guides users' fingertips with a magnetic force field. In the next chapter, we will tie all findings from the four use cases that we described together.

Chapter 6

Conclusion

Although direct input with fingertips on touch screens is beneficial for user interactions, the directness of input and the bareness of fingertips pose a challenge to the input accuracy. This thesis investigates issues in two types of fundamental touch input accuracy: space accuracy ("where it is being touched") and state accuracy ("whether it is being touched"). Four usage scenarios concerning the input accuracies were considered: the touch-based selection technique for users with hand tremors, the state-switching technique for indirect multi-touch systems, the thickness of the near-surface input layer, and the haptic guidance with magnetic force. In this chapter, we first recapitulate our contribution in the individual areas. Following this, we discuss how these investigations suggest possible design strategies for both types of accuracy. We then describe possible research directions and revisit the *raison d'être* of the dichotomy between space and state accuracy.

There are two types of touch input accuracy: space and state accuracy

We investigated the two types of accuracy in four usage scenarios.

6.1 Contributions

Our first contribution benefits users with hand tremors. Involuntary movements in hand tremors impede touch screen input. Tremor oscillations parallel to the screen surface cause users to miss the intended targets, degrading space accuracy. The oscillation in the direction orthogonal to the screen surface causes spurious contacts and lifts, reducing state accuracy. To address these problems, we investigated how touch screen input techniques (in particular, tapping and swip-

Swabbing addresses state and space inaccuracy caused by hand tremors.

ing) influence the tremor oscillation. Based on this knowledge, we designed *swabbing*, an input technique for selecting targets on touch screens. Our user studies show that users can select targets more accurately with swabbing.

We tested four state-switching techniques for indirect multi-touch systems.

Our second contribution targets at indirect multi-touch systems. Such systems combine a horizontal touch screen with a vertical display, allowing users to use expressive multi-touch input in an ergonomic sitting posture. However, the increased distance between input and output space makes it difficult for users to accurately activate the desired position upon touch contacts, which is a problem in space accuracy. To improve space accuracy, touch contacts were repurposed to track cursors. This necessitates a switching technique to engage the cursors (which corresponds to depressing a mouse button). In other words, changing the space accuracy problem to a state accuracy problem. We empirically compared four state-switching techniques and found that the *tap* technique yielded the best state accuracy.

The tap technique yielded the best state accuracy.

We found the thickness of 4 cm to be a sweet spot for the near-surface input layer.

Our third contribution informs the design of input techniques near the surface of touch screens. Such techniques augment touch screens with near-surface sensing that can locate the position of fingertips in a thin layer above the screen surface. On the one hand, input layers that are too thin are difficult to maintain the finger inside. The fingers are likely to drift below and above the layer, resulting in state inaccuracy. On the other hand, input layers that are too thick limit the possible number of layers that can be stacked on each other. Through experimental studies, we found that the thickness of 4 cm allows the fingers to stay accurately inside the layer without an excess toll on manipulation speed. We also found that increasing the thickness more than 4 cm is not especially beneficial.

We used magnetic force to guide users' fingertips, reducing drifts.

Our fourth contribution is a prototypical system to guide users' fingertips. When touch screens are used without vision, touches tend to drift away from the intended targets, reducing space accuracy. To improve accuracy, we presented *FingerFlux*, a system that provides haptic feedback prior to touch contact. The system consists of an array of electromagnets and a permanent magnet that users wear on their fingertips. By changing the polarity of electromagnets, the permanent magnet can be vibrated, pulled toward, or pushed away from the screen. Our study showed that the haptic feedback reduces drifts when the user taps two targets alternatively without vision.

These contributions can be summarized in the Table 6.1.

Usage scenario	Problems in		Contribution
	Space accuracy	State accuracy	
Users with hand tremor	Missing the intended position while tapping	Spurious contacts and lifts cause unintended input	Swabbing improves input accuracy by using swiping trajectories for selecting targets
Indirect multi-touch systems	Direct activation upon touch contact misses the intended position		The tap technique allows cursor tracking and has best state accuracy
Near-surface input layer		Fingers drift above and below the input layer	The thickness of 4 cm is a sweet spot
Eyes-free touch input	Touches drift away from the intended positions		FingerFlux reduces drifts by provides haptic feedback near screen surfaces

Table 6.1: Contributions summary

6.2 Design Strategies for Space Accuracy and State Accuracy

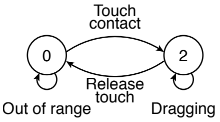
From the works described above, we draw two design strategies for improving space and state accuracy for touch input.

6.2.1 Strategy 1: Add State 1 to Improve Space Accuracy; Be Careful Not to Reduce State Accuracy.

The importance of feedback is well known.

Providing feedback for users' actions is one of the golden rules in user interface design [Shneiderman et al., 2009]. Rapid and continuous feedback was also identified as an essential element in direct manipulation user interfaces by Hutchins et al. [1985] because it "removes the perception of the computer as intermediary" and allows users to "watch the action take place, monitoring [the state of the system] much like [...] monitor[ing] interactions with the physical world".

Touch screens lack immediate feedback during aiming (no state 1).



Problems in space accuracy occur in touch screen input because of the absence of continuous feedback for aiming. Making a touch contact on the screen changes from input state 0 to input state 2. Typical touch systems provide feedback after touch contacts; however, no feedback is provided during aiming due to the absence of input state 1. Although users may perceive the position of their fingertip relative to the screen, the size of the fingertip as well as the parallax effect between the top of the finger and the finger pad [Holz and Baudisch, 2011] impede accurate touch contacts. Augmenting state 1 (aiming) to provide immediate feedback can improve space accuracy.

Works in the literature add immediate feedback in state 1.

Immediate feedback for aiming has been used to improve space accuracy in the literature. For example, to address the fat finger problem (page 2), touch contacts are repurposed to show cursors on the screen (state 1), and additional gestures are used to activate the target (state 2). Examples of such gestures are lifting the finger [Potter et al., 1988] or rocking the finger on the screen [Benko et al., 2006].

Our works also add immediate feedback in state 1.

Similarly, our work addressed the space accuracy problem by adding state 1. In indirect multi-touch systems, touching a horizontal touch screen shows cursors on a vertical screen (Chapter 3). For users with hand tremors, immediate feedback was provided while performing the swabbing gesture (Chapter 2). In FingerFlux, although we did not add any input states, hovering the finger near the screen surface allows users to perceive haptic feedback from the magnetic force, reducing drifts in eyes-free tapping (Chapter 5).

The added states should not degrade state accuracy.

However, additional input states may create state inaccuracy. Each of the additional states requires at least two switching techniques to transition in and out of the state. Designers should ensure that these new transitions are unambiguous with the existing transition and can be controlled reliably by users. Our investigation in state-switching techniques for indirect multi-touch systems (Chapter 3) is an apt example. There, we investigated the effect of state-switching techniques from simple to complex tasks (single-touch, two-touch, and two-hand). Although several techniques are viable in easy tasks, only the tap technique performed well in complex tasks.

6.2.2 Strategy 2: Improve State Accuracy by Accommodating Users' Abilities

State inaccuracy, whether inherent in the interaction technique or incurred upon applying strategy 1, can be addressed either by adding feedback on the states or by designing state transitions that accommodating users' abilities.

Adding visual feedback can improve state accuracy.

Works in the literature added visual feedback to indicate input state. For example, Wigdor et al. [2009] visualized the state of touch contacts by rendering a ripple animation centered on the contact points, analogous to when touching the surface of a still waterbed. In this work, different animations were used to indicate whether a touch captures a virtual object (animation: contracting circle) or not (expanding circle). In the near-surface input, Kattinakere et al. [2007] continuously show the height of the tip of a graphics stylus from the surface with a vertical progress bar that follows the cursor.

These include situations in which visual feedback is not applicable.

However, visual feedback may be undesirable in some situations. Frequent animations from the visual feedback may clutter the screen, adding cognitive load to users. For example, in an application for creating graphical animation, having animated visual feedback for every touch may steal visual attention, resulting in frustrated users. Moreover, users cannot benefit from visual feedback when they are out of their locus of attention—e.g., during eyes-free use.

Consider leveraging users' motor control and proprioception to minimize visual feedback.

Therefore, to ensure state accuracy, techniques for state transitions should be designed according to the users′ abilities, which are the stability of users′ motor control and users′ proprioception of hands and fingers. This may allow reducing visual feedback to minimal or eliminating it altogether. Our study on the thickness of the near-surface input layer exemplifies this strategy (Chapter 4). Even without continuous feedback on the fingertip height (we only indicate when the finger is out of the layer), users can still maintain their fingers reliably within the same layer thickness as in when continuous feedback is provided (e.g., [Kattinakere et al., 2007]). By focusing on user′s abilities (instead of their disabilities), alternative input techniques can be designed to improve both types of accuracy—e.g., in Swabbing (Chapter 2).

6.3 Future Work

On top of the future work mentioned at the end of Chapters 2–5, below are two possible meta-level research directions:

A pattern language for space and state accuracy can be created from the two strategies above.

As discussed above, it is possible to improve space accuracy at the cost of state accuracy. Beyond the literature and our work, this pattern may be repeated in other touch input techniques or in other input devices. Improving space accuracy by adding input states may incur problems in state accuracy, and these problems may be solved by similar solutions. An organized collection of lessons learned from these trade-offs in a pattern language format could aid future interaction designers by providing caveats and suggesting possible solutions. The two strategies above can be considered as a starting point.

It is important to derive a common set of accuracy measures for state accuracy.

Input accuracy is an abstract concept that can be quantified only with a concrete definition, called *accuracy measures*. For space accuracy, many measures are well established, such as MacKenzie et al. [2001]'s seven measure for pointing devices. These accuracy measures capture different facets of space accuracy, which can be used to indicate specific strengths and weaknesses between input techniques. For state accuracy, however, the common measurement is the error rate. The definition of error rate often varies by the task used in each experiment, making it hard to compare the results across experiments. A possible future work is to establish common state accuracy measures with a set of reference values from standard input devices. Chapters 3 and 4 provide several accuracy measures and reference values as a basis for establishing the common set of state accuracy measures.

Chapters 3 and 4 provide some data points to start with.

6.4 Closing Remarks: A False Dichotomy?

Alternative view: state accuracy is the third dimension of space accuracy.

Space accuracy and state accuracy were derived from the definition of touch screens by Buxton et al. [1985]. However, it is possible to contest this dichotomy by reducing both types into space accuracy with a third dimension indicating the states. This is a valid perspective, especially when the input state is determined by continuous movements. For example, in the near-surface input, the input state is determined by the height of the finger. The height is controlled by moving the finger orthogonal to the screen (along the Z-axis), which is similar to the movement that controls the position on the screen (the X- and Y-axis). Therefore, a finger drifting above or below the layer can be viewed as a mistake in space accuracy.

State accuracy concerns with discrete levels and therefore is easier to tackle.

Nevertheless, we argue that considering state accuracy separately from space accuracy makes the problems in input accuracy easier to tackle. Firstly, it is easier to tackle state inaccuracy individually because it is concerned with discrete levels. Distinguishing input errors is clear-cut ("Is the current input state correct or not?"). In contrast, space accuracy concerns a continuous space on the screen. The definition of error in space accuracy (i.e., "How much deviation is considered inaccurate?") can be judgmental, and appropriate error thresholds could vary across tasks.

The decomposition into state accuracy allows more detailed investigation.

Secondly, space and state accuracy may be addressed by different solutions. Each of the solutions for state inaccuracy can be tested independently at the different difficulty levels without influences of space accuracy. For example, in Chapter 3, in which we compared state-switching techniques, although the tap technique is generally well performed, the hold technique can also be used when dwelling long in state 2 is needed; however, it is not to be used by two fingers of the same hand. These details may not be apparent when considering the problem in terms of 3D space accuracy.

State accuracy can be traded in for space accuracy.

Thirdly, this separation allows trading one type of accuracy for another, as described in strategy 1 above. With these merits, we believe that the dichotomy between space and state accuracy serves well as one thinking tool in the mental toolbox of each interaction designer.

> *"Essentially, all models are wrong, but some are useful."*
> —Box and Draper [1986]

Bibliography

L. Anthony, Y. Kim, and L. Findlater. Analyzing user-generated youtube videos to understand touchscreen use by people with motor impairments. In CHI '13: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1223–1232, New York, NY, USA, 2013. ACM.

M. Bachynskyi, G. Palmas, A. Oulasvirta, J. Steimle, and T. Weinkauf. Performance and ergonomics of touch surfaces: A comparative study using biomechanical simulation. In CHI '15: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1817–1826, New York, NY, USA, 2015. ACM.

G. Bailly, R. Walter, J. Müller, T. Ning, and E. Lecolinet. Comparing free hand menu techniques for distant displays using linear, marking and finger-count menus. In P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, and M. Winckler, editors, *Human-Computer Interaction – INTERACT 2011*, volume 6947 of *Lecture Notes in Computer Science*, pages 248–262. Springer Berlin Heidelberg, 2011.

O. Bau, I. Poupyrev, A. Israr, and C. Harrison. Teslatouch: Electrovibration for touch surfaces. In UIST '10: *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, pages 283–292, New York, NY, USA, 2010. ACM.

H. Benko, A. D. Wilson, and P. Baudisch. Precise selection techniques for multi-touch screens. In CHI '06: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1263–1272, New York, NY, USA, 2006. ACM.

A. Beric, P. J. Kelly, A. Rezai, D. Sterio, A. Mogilner, M. Zonenshayn, and B. Kopell. Complications of Deep Brain Stimulation Surgery. *Stereotactic and Functional Neurosurgery*, 77(1-4):73–78, 2001.

S. Boring, M. Jurmu, and A. Butz. Scroll, tilt or move it: Using mobile phones to continuously control pointers on large public displays. In OZCHI '09: *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7*, pages 161–168, New York, NY, USA, 2009. ACM.

G. E. P. Box and N. R. Draper. *Empirical Model-building and Response Surface*. John Wiley & Sons, Inc., New York, NY, USA, 1986.

L. E. Brown, D. A. Rosenbaum, and R. L. Sainburg. Limb position drift: Implications for control of posture and movement. *Journal of Neurophysiology*, 90(5):3105–3118, 2003.

W. Buxton. A three-state model of graphical input. In *Proceedings of the IFIP TC13 Third Interational Conference on Human-Computer Interaction*, pages 449–456, Amsterdam, The Netherlands, The Netherlands, August 1990.

W. Buxton, R. Hill, and P. Rowley. Issues and techniques in touch-sensitive tablet input. In SIGGRAPH '85: *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, pages 215–224, New York, NY, USA, 1985. ACM.

R. Cappelli, D. Maio, and D. Maltoni. Modelling plastic distortion in fingerprint images. In S. Singh, N. Murshed, and W. Kropatsch, editors, *Advances in Pattern Recognition — ICAPR 2001*, volume 2013 of *Lecture Notes in Computer Science*, pages 371–378. Springer Berlin Heidelberg, 2001.

L.-W. Chan, H.-S. Kao, M. Y. Chen, M.-S. Lee, J. Hsu, and Y.-P. Hung. Touching the void: Direct-touch interaction for intangible displays. In CHI '10: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2625–2634, New York, NY, USA, 2010. ACM.

K. B. Chen, A. B. Savage, A. O. Chourasia, D. A. Wiegmann, and M. E. Sesto. Touch Screen Performance by Individuals With and Without Motor Control Disabilities. *Applied ergonomics*, 44(2): 297–302, March 2013.

M. Desmurget, M. Jordan, C. Prablanc, and M. Jeannerod. Constrained and unconstrained movements involve different control strategies. *Journal of Neurophysiology*, 77(3):1644–1650, 1997.

G. Deuschl, P. Bain, and M. Brin. Consensus statement of the movement disorder society on tremor. *Movement Disorders*, 13(S3):2–23, 1998.

A. Dix, J. E. Finlay, G. D. Abowd, and R. Beale. *Human-Computer Interaction (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003.

O. Dogu, S. Sevim, H. Camdeviren, T. Sasmaz, R. Bugdayci, M. Aral, H. Kaleagasi, S. Un, and E. D. Louis. Prevalence of essential tremor: door-to-door neurologic exams in Mersin Province, Turkey. *Neurology*, 61(12):1804–1806, December 2003.

P. Dragicevic. HCI Statistics without p-values. Research Report RR-8738, Inria, June 2015.

W. J. Elias, D. Huss, T. Voss, J. Loomba, M. Khaled, E. Zadicario, R. C. Frysinger, S. A. Sperling, S. Wylie, S. J. Monteith, J. Druzgal, B. B. Shah, M. Harrison, and M. Wintermark. A Pilot Study of Focused Ultrasound Thalamotomy for Essential Tremor. *New England Journal of Medicine*, 369(7): 640–648, August 2013.

L. Findlater, J. O. Wobbrock, and D. Wigdor. Typing on flat glass: Examining ten-finger expert typing patterns on touch surfaces. In CHI '11: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2453–2462, New York, NY, USA, 2011. ACM.

C. Forlines, C. Shen, and B. Buxton. Glimpse: A novel input model for multi-level devices. In CHI EA '05: *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, pages 1375–1378, New York, NY, USA, 2005. ACM.

C. Forlines, D. Vogel, and R. Balakrishnan. Hybridpointing: Fluid switching between absolute and relative pointing with a direct input device. In UIST '06: *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, pages 211–220, New York, NY, USA, 2006. ACM.

C. Forlines, D. Wigdor, C. Shen, and R. Balakrishnan. Direct-touch vs. mouse input for tabletop displays. In CHI '07: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 647–656, New York, NY, USA, 2007. ACM.

J. Froehlich, J. O. Wobbrock, and S. K. Kane. Barrier pointing: Using physical edges to assist target acquisition on mobile device touch screens. In Assets '07: *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 19–26, New York, NY, USA, 2007. ACM.

M. Fukumoto and T. Sugimura. Active click: Tactile feedback for touch panels. In CHI EA '01: *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, pages 121–122, New York, NY, USA, 2001. ACM.

J. P. Giuffrida, D. E. Riley, B. N. Maddux, and D. A. Heldman. Clinically deployable kinesia^{TM} technology for automated tremor assessment. *Movement Disorders*, 24(5):723–730, 2009.

J. Gordon, M. Ghilardi, S. Cooper, and C. Ghez. Accuracy of planar reaching movements. *Experimental Brain Research*, 99(1):112–130, 1994.

M. Gresty and D. Buckwell. Spectral analysis of tremor: understanding the results. *Journal of Neurology, Neurosurgery & Psychiatry*, 53(11):976–981, 1990.

C. Häger-Ross and M. H. Schieber. Quantifying the Independence of Human Finger Movements: Comparisons of Digits, Hands, and Movement Frequencies. *The Journal of Neuroscience*, 20(22): 8542–8550, November 2000.

N. Hawthorne and W. Crane. A Wonder Book for Girls and Boys... With 60 Designs by Walter Crane, 1892.

O. Hilliges, S. Izadi, A. D. Wilson, S. Hodges, A. Garcia-Mendoza, and A. Butz. Interactions in the air: Adding further depth to interactive tabletops. In UIST '09: *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*, pages 139–148, New York, NY, USA, 2009. ACM.

J. D. Hincapié-Ramos, X. Guo, P. Moghadasian, and P. Irani. Consumed endurance: A metric to quantify arm fatigue of mid-air interactions. In CHI '14: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1063–1072, New York, NY, USA, 2014. ACM.

K. Hinckley. The human-computer interaction handbook. chapter Input Technologies and Techniques, pages 95–132. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 2012.

K. Hinckley, K. Yatani, M. Pahud, N. Coddington, J. Rodenhouse, A. Wilson, H. Benko, and B. Buxton. Pen + touch = new tools. In UIST '10: *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, pages 27–36, New York, NY, USA, 2010. ACM.

C. Holz and P. Baudisch. Understanding touch. In CHI '11: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2501–2510, New York, NY, USA, 2011. ACM.

T. Hoshi, M. Takahashi, T. Iwamoto, and H. Shinoda. Noncontact tactile display based on radiation pressure of airborne ultrasound. *Haptics, IEEE Transactions on*, 3(3):155–165, July 2010.

C. Hsieh. Touch Panel Market Tracker. Technical report, September 2015.

N. Huck. Touchscreen test input for users with hand tremor. Diploma thesis, RWTH Aachen University, Aachen, January 2012.

J. Hurtmanns. Swabbing: Touchscreen input for users with hand tremor. Bachelor's thesis, Aachen, March 2011.

E. L. Hutchins, J. D. Hollan, and D. A. Norman. Direct manipulation interfaces. *Human–Computer Interaction*, 1(4):311–338, 1985.

C. B. Irwin and M. E. Sesto. Performance and touch characteristics of disabled and non-disabled participants during a reciprocal tapping task using touch screen technology. *Applied Ergonomics*, 43 (6):1038 – 1043, 2012.

R. J. K. Jacob. The use of eye movements in human-computer interaction techniques: What you look at is what you get. *ACM Transactions on Information Systems*, 9(2):152–169, April 1991.

Y. Jansen, T. Karrer, and J. Borchers. Mudpad: Tactile feedback and haptic texture overlay for touch surfaces. In ITS '10: *ACM International Conference on Interactive Tabletops and Surfaces*, pages 11–14, New York, NY, USA, 2010. ACM.

E. T. Jaynes. *The Physical Basis of Music*. Washington University, 1996.

Z. Jin, T. Plocher, and L. Kiff. Touch screen user interfaces for older adults: Button size and spacing. In C. Stephanidis, editor, *Universal Acess in Human Computer Interaction. Coping with Diversity*, volume 4554 of *Lecture Notes in Computer Science*, pages 933–941. Springer Berlin Heidelberg, 2007.

R. S. Kattinakere, T. Grossman, and S. Subramanian. Modeling steering within above-the-surface interaction layers. In CHI '07: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 317–326, New York, NY, USA, 2007. ACM.

D. Kehrig. Swabbing in the wild - a longitudinal study of touchscreen input methods for users with hand tremor. Diploma thesis, RWTH Aachen University, Aachen, June 2013.

K. Kin, M. Agrawala, and T. DeRose. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In GI '09: *Proceedings of Graphics Interface 2009*, pages 119–124, Toronto, Ont., Canada, Canada, 2009. Canadian Information Processing Society.

S. Knoedel and M. Hachet. Multi-touch rst in 2d and 3d spaces: Studying the impact of directness on user performance. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on*, pages 75–78, March 2011.

T. S. Kuhn. *The structure of scientific revolutions*. University of Chicago press, 2012.

C. E. Lang and M. H. Schieber. Human finger independence: Limitations due to passive mechanical coupling versus active neuromuscular control. *Journal of Neurophysiology*, 92(5):2802–2810, 2004.

J. Lazar, J. H. Feng, and H. Hochheiser. *Research Methods in Human-Computer Interaction*. Wiley Publishing, 2010.

H. Lee, J.-S. Kim, S. Choi, J.-H. Jun, J.-R. Park, A.-H. Kim, H.-B. Oh, H.-S. Kim, and S.-C. Chung. Mid-air tactile stimulation using laser-induced thermoelastic effects: The first study for indirect radiation. In *World Haptics Conference (WHC), 2015 IEEE*, pages 374–380, June 2015.

R. Linden. User performance effects of input/output ratio on indirect touch systems. Master's thesis, RWTH Aachen University, Aachen, September 2013.

E. D. Louis, S. J. Frucht, and E. Rios. Intention Tremor in Essential Tremor: Prevalence and Association with Disease Duration. *Movement disorders : official journal of the Movement Disorder Society*, 24(4): 626–627, March 2009.

I. S. MacKenzie and A. Oniszczak. A comparison of three selection techniques for touchpads. In CHI '98: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 336–343, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

I. S. MacKenzie, T. Kauppinen, and M. Silfverberg. Accuracy measures for evaluating computer pointing devices. In CHI '01: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 9–16, New York, NY, USA, 2001. ACM.

S. Malik, A. Ranjan, and R. Balakrishnan. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In UIST '05: *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, pages 43–52, New York, NY, USA, 2005. ACM.

N. Marquardt, M. A. Nacenta, J. E. Young, S. Carpendale, S. Greenberg, and E. Sharlin. The haptic tabletop puck: Tactile feedback for interactive tabletops. In ITS '09: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 85–92, New York, NY, USA, 2009. ACM.

N. Marquardt, R. Jota, S. Greenberg, and J. Jorge. The continuous interaction space: Interaction techniques unifying touch and gesture on and above a digital surface. In P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, and M. Winckler, editors, *Human-Computer Interaction – INTERACT 2011*, volume 6948 of *Lecture Notes in Computer Science*, pages 461–476. Springer Berlin Heidelberg, 2011.

W. S. Marras. *Basic Biomechanics and Workstation Design*, pages 347–381. John Wiley & Sons, Inc., 2012.

J. Matejka, T. Grossman, J. Lo, and G. Fitzmaurice. The design and evaluation of multi-finger mouse emulation techniques. In CHI '09: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1073–1082, New York, NY, USA, 2009. ACM.

A. Mertens, N. Jochems, C. M. Schlick, D. Dünnebacke, and J. H. Dornberg. Design pattern trabing: Touchscreen-based input technique for people affected by intention tremor. In EICS '10: *Proceedings of the 2Nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, pages 267–272, New York, NY, USA, 2010. ACM.

A. Mertens, C. Wacharamanotham, J. Hurtmanns, N. Huck, and J. S. and. Swabbing: Touchscreen-basierte mensch-rechner-interaktion für personen mit tremorerkrankungen des hand-arm-apparats. *das wissenschaftliche Programm des Kongresses der Deutschen Gesellschaft für Neurologie (DGN)*, September 2011. (accepted for poster presentation).

A. Mertens, C. Wacharamanotham, J. Hurtmanns, M. Kronenbuerger, P. H. Kraus, A. Hoffmann, C. Schlick, and J. Borchers. *Model Based Processing of Swabbing Movements on Touch Screens to Improve Accuracy and Efficacy for Information Input of Individuals Suffering from Kinetic Tremor*, volume 98 of *Advances in Intelligent and Soft Computing*, pages 503–522. Springer Berlin Heidelberg, August 2012.

T. Miyaki and J. Rekimoto. Graspzoom: Zooming and scrolling control model for single-handed mobile interaction. In MobileHCI '09: *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 11:1–11:4, New York, NY, USA, 2009. ACM.

Y. Monnai, K. Hasegawa, M. Fujiwara, K. Yoshino, S. Inoue, and H. Shinoda. Haptomime: Mid-air haptic interaction with a floating virtual screen. In UIST '14: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, pages 663–667, New York, NY, USA, 2014. ACM.

K. Montague, H. Nicolau, and V. L. Hanson. Motor-impaired touchscreen interactions in the wild. In ASSETS '14: *Proceedings of the 16th International ACM SIGACCESS Conference on Computers & Accessibility*, pages 123–130, New York, NY, USA, 2014. ACM.

M. Morris, A. Brush, and B. Meyers. Reading revisited: Evaluating the usability of digital display surfaces for active reading tasks. In *Horizontal Interactive Human-Computer Systems, 2007. TABLETOP '07. Second Annual IEEE International Workshop on*, pages 79–86, Oct 2007.

T. Moscovich and J. F. Hughes. Indirect mappings of multi-touch input using one and two hands. In CHI '08: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1275–1284, New York, NY, USA, 2008. ACM.

H. Nicolau and J. Jorge. Elderly text-entry performance on touchscreens. In ASSETS '12: *Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 127–134, New York, NY, USA, 2012. ACM.

H. Nicolau, T. Guerreiro, J. Jorge, and D. Gonçalves. Mobile touchscreen user interfaces: bridging the gap between motor-impaired and able-bodied users. *Universal Access in the Information Society*, 13 (3):303–313, 2014.

D. A. Norman. *The Design of Everyday Things*. Basic Books, reprint edition, September 2002.

T. Ohtsuki. Inhibition of individual fingers during grip strength exertion. *Ergonomics*, 24(1):21–36, 1981.

A. Olwal, S. Feiner, and S. Heyman. Rubbing and tapping for precise and rapid selection on touchscreen displays. In CHI '08: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 295–304, New York, NY, USA, 2008. ACM.

D. T. V. Pawluk and R. D. Howe. Dynamic contact of the human fingerpad against a flat surface. *Journal of Biomechanical Engineering*, 121(6):605–611, 12 1999.

R. L. Potter, L. J. Weldon, and B. Shneiderman. Improving the accuracy of touch screens: An experimental evaluation of three strategies. In CHI '88: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 27–32, New York, NY, USA, 1988. ACM.

J. Reason. *Human error*. Cambridge university press, 1990.

J. Rekimoto. Senseablerays: Opto-haptic substitution for touch-enhanced interactive spaces. In CHI EA '09: *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, pages 2519–2528, New York, NY, USA, 2009. ACM.

H. Roels, J. Malchaire, J.-P. V. Wambeke, J.-P. Buchet, and R. Lauwerys. Development of a Quantitative Test for Hand Tremor Measurement Study of Factors Influencing the Response in a Control Population. *Journal of Occupational and Environmental Medicine*, 25(6):481, June 1983.

D. Schmidt, F. Block, and H. Gellersen. A comparison of direct and indirect multi-touch input for large surfaces. In T. Gross, J. Gulliksen, P. Kotzé, L. Oestreicher, P. Palanque, R. Prates, and M. Winckler, editors, *Human-Computer Interaction – INTERACT 2009*, volume 5726 of *Lecture Notes in Computer Science*, pages 582–594. Springer Berlin Heidelberg, 2009.

S. Sharifi, A. J. Nederveen, J. Booij, and A.-F. van Rootselaar. Neuroimaging essentials in essential tremor: A systematic review. *NeuroImage : Clinical*, 5:217–231, 2014.

B. Shneiderman, C. Plaisant, M. Cohen, and S. Jacobs. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley Publishing Company, USA, 5th edition, 2009.

M. Spindler and R. Dachselt. Paperlens: Advanced magic lens interaction above the tabletop. In ITS '09: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 7:1–7:1, New York, NY, USA, 2009. ACM.

M. Spindler, M. Martsch, and R. Dachselt. Going beyond the surface: Studying multi-layer interaction above the tabletop. In CHI '12: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1277–1286, New York, NY, USA, 2012. ACM.

S. Subramanian, D. Aliakseyeu, and A. Lucero. Multi-layer interaction for digital tables. In UIST '06: *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, pages 269–272, New York, NY, USA, 2006. ACM.

Y. Suzuki and M. Kobayashi. Air jet driven force feedback in virtual reality. *Computer Graphics and Applications, IEEE*, 25(1):44–47, Jan 2005.

K. C. Veluvolu and W. T. Ang. Estimation of physiological tremor from accelerometers for real-time applications. *Sensors*, 11(3):3020, 2011.

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

D. Vogel and P. Baudisch. Shift: A technique for operating pen-based interfaces using touch. In CHI '07: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 657–666, New York, NY, USA, 2007. ACM.

C. Wacharamanotham, J. Hurtmanns, A. Mertens, M. Kronenbuerger, C. Schlick, and J. Borchers. Evaluating swabbing: A touchscreen input method for elderly users with tremor. In CHI '11: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 623–626, New York, NY, USA, 2011. ACM.

C. Wacharamanotham, D. Kehrig, A. Mertens, C. Schlick, and J. Borchers. Designing a touchscreen web browser for people with tremors. In *CHI 2013 Workshop on Mobile Accessibility*, Paris, France, April 2013.

C. Wacharamanotham, K. Todi, M. Pye, and J. Borchers. Understanding finger input above desktop devices. In CHI '14: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1083–1092, New York, NY, USA, 2014. ACM.

F. Wang and X. Ren. Empirical evaluation for finger input properties in multi-touch interaction. In CHI '09: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1063–1072, New York, NY, USA, 2009. ACM.

M. Weiss. *Bringing Haptic General-Purpose Controls to Interactive Tabletops*. PhD thesis, Aachen, September 2012.

M. Weiss, F. Schwarz, S. Jakubowski, and J. Borchers. Madgets: Actuating widgets on interactive tabletops. In UIST '10: *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, pages 293–302, New York, NY, USA, 2010a. ACM.

M. Weiss, S. Voelker, C. Sutter, and J. Borchers. Benddesk: Dragging across the curve. In ITS '10: *ACM International Conference on Interactive Tabletops and Surfaces*, pages 1–10, New York, NY, USA, 2010b. ACM.

M. Weiss, C. Wacharamanotham, S. Voelker, and J. Borchers. Fingerflux: Near-surface haptic feedback on tabletops. In UIST '11: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, pages 615–620, New York, NY, USA, 2011. ACM.

D. Wigdor and D. Wixon. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2011.

D. Wigdor, S. Williams, M. Cronin, R. Levy, K. White, M. Mazeev, and H. Benko. Ripples: Utilizing per-contact visualizations to improve user interaction with touch displays. In UIST '09: *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*, pages 3–12, New York, NY, USA, 2009. ACM.

J. O. Wobbrock and K. Z. Gajos. Goal crossing with mice and trackballs for people with motor impairments: Performance, submovements, and design directions. *ACM Trans. Access. Comput.*, 1(1): 4:1–4:37, May 2008.

J. O. Wobbrock, S. K. Kane, K. Z. Gajos, S. Harada, and J. Froehlich. Ability-based design: Concept, principles and examples. *ACM Trans. Access. Comput.*, 3(3):9:1–9:27, April 2011.

C. Yu, X. Tan, Y. Shi, and Y. Shi. Air finger: Enabling multi-scale navigation by finger height above the surface. In UbiComp '11: *Proceedings of the 13th International Conference on Ubiquitous Computing*, pages 495–496, New York, NY, USA, 2011. ACM.

S. Zhai, C. Morimoto, and S. Ihde. Manual and gaze input cascaded (magic) pointing. In CHI '99: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 246–253, New York, NY, USA, 1999. ACM.

Index

Other publications by Chat Wacharamanotham (as of 08.02.16)

Peer-reviewed archival conference papers in HCI

Simon Voelker, Kjell Ivar Øvergård, Chat Wacharamanotham, and Jan Borchers. 2015. **Knobology Revisited: A Comparison of User Performance between Tangible and Virtual Rotary Knobs.** In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (ITS '15). ACM, New York, NY, USA. (to appear)

Chat Wacharamanotham, Krishna Subramanian, Sarah Theres Völkel, and Jan Borchers. 2015. **Statsplorer: Guiding Novices in Statistical Analysis.** In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15). ACM, New York, NY, USA, 2693-2702.

Florian Heller, Stefan Ivanov, Chat Wacharamanotham, and Jan Borchers. 2014. **FabriTouch: Exploring Flexible Touch Input on Textiles.** In *Proceedings of the 2014 ACM International Symposium on Wearable Computers* (ISWC '14). ACM, New York, NY, USA, 59-62.

Chat Wacharamanotham, Kashyap Todi, Marty Pye, and Jan Borchers. 2014. **Understanding finger input above desktop devices.** In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems* (CHI '14). ACM, New York, NY, USA, 1083-1092.

Simon Voelker, Chat Wacharamanotham, and Jan Borchers. 2013. **An Evaluation of State switching Methods for Indirect Touch Systems.** In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13). ACM, New York, NY, USA, 745-754.

Simon Voelker, Malte Weiss, Chat Wacharamanotham, and Jan Borchers. 2011. **Dynamic portals: a lightweight metaphor for fast object transfer on interactive surfaces.** In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (ITS '11). ACM, New York, NY, USA, 158-161. ***(Best note award)***

Malte Weiss, Chat Wacharamanotham, Simon Voelker, and Jan Borchers. 2011. **FingerFlux: near-surface haptic feedback on tabletops.** In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (UIST '11). ACM, New York, NY, USA, 615-620. ***(Best demo award 3rd place)***

Chat Wacharamanotham, Jan Hurtmanns, Alexander Mertens, Martin Kronenbuerger, Christopher Schlick, and Jan Borchers. 2011. **Evaluating Swabbing: A Touchscreen Input Method for Elderly Users with Tremor.** In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11). ACM, New York, NY, USA, 623-626.

Book chapter

Alexander Mertens, Chat Wacharamanotham, Jan Hurtmanns, Martin Kronenbuerger, Peter H. Kraus, Arndt Hoffmann, Christopher Schlick, and Jan Borchers. **Model Based Processing of Swabbing Movements on Touch Screens to Improve Accuracy and Efficacy for Information Input of Individuals Suffering from Kinetic Tremor**. *Human – Computer Systems Interaction: Backgrounds and Applications 2: Advances in Intelligent and Soft Computing,* 98:503–522, August 2012.

Work-in-progress articles and posters (peer-reviewed, non-archival)

Florian Heller, Simon Voelker, Chat Wacharamanotham, and Jan Borchers. 2015. **Transporters: Vision & Touch Transitive Widgets for Capacitive Screens.** In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (CHI EA '15). ACM, New York, NY, USA, 1603-1608.

Sukeshini A. Grandhi, Chat Wacharamanotham, Gina Joue, Jan O. Borchers, and Irene Mittelberg. 2013. **How we gesture towards machines: an exploratory study of user perceptions of gestural interaction.** In *CHI '13 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '13). ACM, New York, NY, USA, 1209-1214.

Alexander Mertens, Chatchavan Wacharamanotham, Jan Hurtmanns, Niels Huck, Jörg Schulz, Jan Borchers, Christopher Schlick, and Martin Kronenbuerger. **SWABBING: Touchscreen-basierte Mensch-Rechner-Interaktion für Personen mit Tremorerkrankungen des Hand-Arm-Apparats** (P438), In: *Abstract-CD des 84. Kongresses der Deutschen Gesellschaft für Neurologie 2011*, 2, Wiesbaden, pharma service, Hannover 2011, 1-2.

alt.chi (peer-reviewed, non-archival)

Christian Corsten, Chat Wacharamanotham, and Jan Borchers. 2013. **Fillables: everyday vessels as tangible controllers with adjustable haptics.** In *CHI '13 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '13). ACM, New York, NY, USA, 2129-2138.

Workshop papers (peer-reviewed, non-archival)

Chat Wacharamanotham, Dennis Kehrig, Alexander Mertens, Christopher Schlick, and Jan Borchers. **Designing a Touchscreen Web Browser for People with Tremors.** In *CHI 2013 Workshop on Mobile Accessibility*, Paris, France, April 2013.

Tongyan Ning, Jörg Müller, Robert Walter, Gilles Bailly, Chat Wacharamanotham, Florian Alt, and Jan Borchers. **No Need to Stop: Menu Techniques for Passing by Public Displays.** In *CHI 2011 Workshop on Large Displays in Urban Life: from Exhibition Halls to Media Facades*, Vancouver, Canada, May 2011.

Chatchavan Wacharamanotham, Jonathan Meyer, Jonathan Diehl, and Jan Borchers. **The Interactive Bracelet: An input device for bimanual interaction.** In *Mobile HCI 2010 Workshop on Ensembles of On-Body Devices*, Lisbon, Portugal, September 2010.

Doctoral consortium (juried, non-archival)

Chat Wacharamanotham. 2014. **Making bare hand input more accurate.** In *CHI '14 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '14). ACM, New York, NY, USA, 307-310.

Theses

Chatchavan Wacharamanotham. **Using semantic web technologies for data update**, Master's Thesis, School of Informatics, University of Edinburgh, 2009.

Chatchavan Wacharamanotham. **A study of substring search in massive databases**, Bachelor's Thesis, Faculty of Science, Chulalongkorn University, 2005.

Version: 1.0 Typeset Tuesday 9th February, 2016

For updates, see http://chat.info/driftsslipsmisses

http://chat.info/driftsslipsmisses