ABSTRACT

Alternatives using the Leap Motion to extend Mid-air Word-gesture Keyboards

Garrett Benoit, M.S.

Faculty Adviser: G. Michael Poor, Ph.D.

Lately, the application of touch-less, mid-air, gesture-based interactions has risen in popularity for a variety of fields (e.g., augmented reality, virtual reality, gaming, etc.). With this new-found, wide-spread application comes the need for efficient, mid-air, gesture-based text-entry. The solution: apply word-gesture keyboards to mid-air text-entry. Markussen et al. (2014) applied the first word-gesture keyboard for use in mid-air text-entry with the inception of Vulture, achieving a mid-air text-entry rate of 28.1 Words Per Minute (WPM) (59% of direct touch input) with repeated sessions and training, the fastest mid-air text-entry rate yet. This thesis builds on the findings of Markussen et al. (2014) and presents alternatives to mid-air word separation (the technique used to delimit between different words) to further benefit mid-air, word-gesture keyboards. This thesis also explores problems associated with these alternatives and presents possible solutions. Results have revealed that a bimodal approach to mid-air word separation holds great promise, reaching a mean text-entry rate, with no training, of 15.8 WPM (81% of direct touch input) as compared to Vulture ($M = 11.8$ WPM) for a single session.

Alternatives using the Leap Motion to extend Mid-air Word-gesture Keyboards

by

Garrett Benoit, B.S.

A Thesis

Approved by the Department of Computer Science

Gregory D. Speegle, Ph.D., Chairperson

Submitted to the Graduate Faculty of
Baylor University in Partial Fulfillment of the
Requirements for the Degree of

Master of Science

Approved by the Thesis Committee

G. Michael Poor, Ph.D., Chairperson

Bill Poucher, Ph.D.

Brian Garner, Ph.D.

Accepted by the Graduate School
December 2015

J. Larry Lyon, Ph.D., Dean

Page bearing signatures is kept on file in the Graduate School.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

To Dr. G. Michael Poor for allowing me to explore this idea in depth, guiding me to a path of success, and for putting up with me the whole time. Alvin Jude Hari Haran and Darren Guinness for their invaluable input that helped shaped the direction of this study. To all of those in the HCI Lab who helped with my pilot studies! To all the volunteers who made this research possible by generously donating their time. Finally, to my sister, Heather Benoit, and my parents who inspired me to go to Baylor and seek a greater future.

To those who pioneer the future.

CHAPTER ONE

Introduction

With the increase in gesture-controlled interfaces for touch screen and other modern devices, gesture-controls have started to see a transition for use in mid-air (i.e., screen-less interaction). Mid-air, gesture-controlled content has seen its emergence in large displays (Nancel, Wagner, Pietriga, Chapuis, and Mackay 2011, Xie, Xu, Guo, Zhu, and Wu 2013), smart phones (Jones, Sodhi, Forsyth, Bailey, and Maciocci 2012), augmented reality (Piumsomboon, Clark, Billinghurst, and Cockburn 2013), and desktop computers (Sutton 2013, Guinness 2015, Hari Haran 2014, Bachmann, Weichert, and Rinkenauer 2015). Mid-air pointing has been a common approach to many of these gesture-controlled interactions and is used to select and manipulate on-screen objects (Banerjee, Burstyn, Girouard, and Vertegaal 2012, Cockburn, Quinn, Gutwin, Ramos, and Looser 2011, Jota, Nacenta, Jorge, Carpendale, and Greenberg 2010, Shoemaker, Tang, and Booth 2007, Vogel and Balakrishnan 2005, Xie, Xu, Guo, Zhu, and Wu 2013). However, techniques for reasonable mid-air text-entry rates are fairly new. Past technologies allowed for mid-air text-entry (e.g., Touchpad, Airstroke), but those approaches had fallen short of any meaningful text-entry rates (Malik and Laszlo 2004, Ni, Bowman, and North 2011). More modern approaches of mid-air text-entry have seen improved results but still low, around 13.2 (Markussen, Jakobsen, and Hornbæk 2013) to 18.9 (Shoemaker, Findlater, Dawson, and Booth 2009) words per minute. These approaches were limited to selection of individual characters and, at best, similar to the multi-tap features of touch-based entry (Markussen, Jakobsen, and Hornbæk 2013, Ni, Bowman, and North 2011, Shoemaker, Findlater, Dawson, and Booth 2009). Last year, the largest improvement was seen in mid-air text-entry when Markussen et al. (2014) transitioned word-gesture keyboards to mid-air with the development of Vulture, reaching a text-entry rate of

20.6 words per minute for their initial study. They achieved an even higher text-entry rate of 28.1 Words Per Minute in a second study with training of single, short phrases, indicating that learning the new techniques will help bring mid-air text-entry closer to touch-based text-entry. Vulture reached 59% of the text-entry rate of touch-based inputs which is the fastest mid-air text-entry rate seen yet (Markussen, Jakobsen, and Hornbæk 2014).

The purpose of this thesis was to use the Leap Motion controller (Motion 2012), a new and emerging technology (Sutton 2013, Weichert, Bachmann, Rudak, and Fisseler 2013, Guna, Jakus, Pogačnik, Tomažič, and Sodnik 2014), that interprets mid-air gestural inputs, for text entry (Toimil, Shin, and Eisner 2013). The only previous attempt for text entry with the Leap Motion controller was in mid-air handwriting (Vikram, Li, and Russell 2013). However, even analog handwriting is slow and confined to around 15 words per minute (Devoe 1967). Instead, this study aims to follow the path of Markussen et al. (Markussen, Jakobsen, and Hornbæk 2014) and use the Leap Motion controller to extend word-gesture keyboards to mid-air text-entry. Word-gesture keyboards have garnered popularity with the advent of smart phones and tablets and have been proven to perform well on touch screens (Kristensson 2007, Zhai and Kristensson 2012, Zhai, Kristensson, Gong, Greiner, Peng, Liu, and Dunnigan 2009). This study intended to use the Leap Motion controller to find alternatives to word separation for text-entry mid-air, word-gesture keyboards and find a more natural approach to wearing a glove or detecting pinching (Markussen, Jakobsen, and Hornbæk 2014, Ni, Bowman, and North 2011), for the mid-air equivalent of tapping and releasing for delimiters of words. This study explored using the extra degrees of freedom available in mid-air (e.g., depth) to examine the problems and solutions to these techniques. This study also used other techniques of simulating touch (e.g., gesture prediction, bimodal input) for the mid-air equivalent of tapping and releasing for delimiters of words. The idea was to achieve something that was as simple as using touch-based devices while still allowing for a multi-functional workspace.

Touchless gesture-controllers for mid-air text-entry will benefit augmented reality (e.g., Google Glass, Microsoft HoloLens) and virtual reality (e.g., Oculus Rift). Furthermore, they will greatly benefit the medical industry (e.g., operating rooms, those with disabilities, pathogen research), providing sanitary, sterile, and efficient means of text-entry in fragile environments.

1.1 Motivation

This research was inspired by the older generation, especially veterans who are amputees (Yu, Kato, Fabio, Yokoi, and Kakazu 2003), and those who have trouble typing efficiently with standard computer keyboards (Finn 1997, Taveira and Choi 2009). Typing long emails or messages when using one hand, or an improvised device for double amputees, and having to type each character individually is an arduous and painstaking task. This led to seeking a better method for text-entry, focused on usability for those who are generally limited to single-input text-entry when using a standard keyboard; a method to make typing and other interactions with computers simpler and more efficient, not as tedious as they are for users restricted to single-input text-entry. Observing the work of Alvin Jude Hari Haran and Darren Guiness on accessibility with using mid-air gestures to control the mouse using the Leap Motion controller (Motion 2012, Hari Haran 2014, Guinness 2015) influenced this research to put the same technology to use with mid-air keyboards. With all of the phones and tablets in the market today shipping standard with word-gesture keyboards and efficient, single-handed text-entry, it was immediately evident that it would be beneficial to apply word-gesture keyboard techniques to mid-air to replace conventional keyboard use.

At the beginning of this research, there were no other studies on mid-air, word-gesture keyboards. This thesis sought to try many different techniques, focused mostly on word-gesture keyboards, to explore their usage in mid-air. Shortly after research began, Vulture was released (Markussen, Jakobsen, and Hornbæk 2014), an exciting study, and the first to apply word-gesture keyboards to mid-air text-entry. As a result, the focus of this research changed from seeing *if* word-gesture keyboards could work for mid-air text-entry to exploring other alternatives for *how* they can work.

1.2 Thesis Overview

This thesis was completed in compliance with the requirements for the Master of Science in Computer Science at Baylor University. To begin, background research was conducted on mid-air pointing, mid-air text-entry, word-gesture keyboards, and text-entry evaluation, which are presented in Chapter 2. With this understanding, Chapter 3 discusses the design and construction of this study's different word-gesture keyboards using a finger or stylus as input. The word-gesture keyboard implementations and keyboard layouts were heavily influenced by the initial pilot studies presented in Chapter 4. Chapter 5 details the task design, procedure, and dependent measures focused on in the full study while also highlighting a few challenges and limitations and their possible solutions. Results of the the various word-gesture keyboards are presented and discussed in depth in Chapter 6. Discussion of the findings, potential future work, and concluding remarks are given in Chapter 7.

CHAPTER TWO

Literature Review

The literature review covers several important topics relevant to design and construct a mid-air, word-gesture keyboard. Background information of mid-air pointing is discussed in Section 2.1 to understand methods for interacting with display areas from afar. The importance of barehanded interaction for gestural input is then examined in Section 2.2. Section 2.3 presents some of the most relevant past works on mid-air text-entry and how they have evolved over time. The influence of word-gesture keyboards for single-hand text-entry is analyzed in Section 2.4. Finally in Section 2.5, various techniques for text-entry evaluation are reviewed.

2.1 Mid-air Pointing

Finding alternatives to word separation (i.e., the technique used to delimit between different words) for mid-air text-entry heavily involves utilizing mid-air pointing to give character-level precision and determine the gesture-shapes being drawn. In the past, much of the research on mid-air pointing was done through casting rays onto an assortment of distant display screens (Banerjee, Burstyn, Girouard, and Vertegaal 2012, Cockburn, Quinn, Gutwin, Ramos, and Looser 2011, Jota, Nacenta, Jorge, Carpendale, and Greenberg 2010, Shoemaker, Tang, and Booth 2007, Vogel and Balakrishnan 2005, Xie, Xu, Guo, Zhu, and Wu 2013). However, these methods vary vastly in their efficiency, precision, accuracy, and fatigue. Even something as simple as hand jitter was a limiting factor to these sub-par detection techniques for mid-air pointing. These issues have recently begun to be overcome thanks to products like the Leap Motion controller, which can deliver tracking precision on the sub-millimeter level and implements techniques to detect and stabilize for hand jitter (Weichert, Bachmann, Rudak, and Fisseler 2013, Guna, Jakus, Pogačnik, Tomažič, and Sodnik 2014) Another major factor that had heavily affected these ray-casting techniques was their dependence on the distance from the display screen (Shoemaker, Findlater, Dawson, and Booth 2009). Markussen et al. (2013) chose to minimize the distance dependence by projecting movement orthogonally onto the display to maintain a constant control-display ratio at the cost of the user's reach. A second option, the option used in this thesis, was to instead project an interaction plane in front of the user within their reach that was independent from the display space (Hari Haran 2014, Guinness 2015, Bachmann, Weichert, and Rinkenauer 2015). As stated by the principle of stimulus-response compatibility (Proctor and Vu 2006), the major issue in decoupling (or separating) the motor space and display space was that participants had to recouple (or combine) these two spaces mentally (Markussen, Jakobsen, and Hornbæk 2014). Using different techniques to calibrate the motor space has shown to partially aid in re-coupling these spaces and provide for efficient pointing throughput (Hari Haran 2014, Guinness 2015).

2.2 Barehanded Interaction

Generally to track gestures, users are required to wear gloves or use reflective markers (Markussen, Jakobsen, and Hornbæk 2014, Ni, Bowman, and North 2011). However, users should not be encumbered by wearing wires or external devices for interactions (Wachs, Kölsch, Stern, and Edan 2011). This was known as the design principle: "Come As You Are" (Triesch and Von Der Malsburg 1998). Adhering to the "Come As You Are" design principle leads to the barehanded interaction method, allowing the participant to be free of attached wires or devices. The barehanded approach for gestural interactions has been shown to be better for gestures than traditional input devices (Von Hardenberg and Bérard 2001). The Leap Motion controller was used to track these gestures because it has been shown to be an acceptable tracking device for barehanded gestural interactions (Hari Haran 2014, Guinness 2015, Weichert, Bachmann, Rudak, and Fisseler 2013, Guna, Jakus, Pogačnik, Tomažič, and Sodnik 2014, Bachmann, Weichert, and Rinkenauer 2015) and supports the "Come As You Are" design approach.

2.3 Mid-air Text-entry

Since its inception, many techniques have been used for mid-air text-entry that all vary in error rates and text-entry rates. Mid-air text-entry has evolved along-side both the techniques for normal text-entry and the hardware available for tracking tools and hands.

2.3.1 Selection-based

Some of the earliest work on mid-air text entry focused on selection-based techniques (Shoemaker, Findlater, Dawson, and Booth 2009, Markussen, Jakobsen, and Hornbæk 2013). Such efforts were centered around using various layouts (Shoemaker, Findlater, Dawson, and Booth 2009, Markussen, Jakobsen, and Hornbæk 2013, Hincapié-Ramos, Guo, and Irani 2014, Jones, Alexander, Andreou, Irani, and Subramanian 2010), hovering (Malik and Laszlo 2004), multi-tap (Markussen, Jakobsen, and Hornbæk 2013), accelerometers (Jones, Alexander, Andreou, Irani, and Subramanian 2010, Shoemaker, Findlater, Dawson, and Booth 2009), bimodal entry (Shoemaker, Findlater, Dawson, and Booth 2009), or even selection-wheels (Shoemaker, Findlater, Dawson, and Booth 2009). The major limitation to these techniques was that all of them used single-input text-entry, which is inherently slow. One of the earliest implementations, Visual Touchpad, used web-cams to track hand motion (Malik and Laszlo 2004). However, it was limited by the hardware and processing of the time, requiring a 3-second period of hovering to select an individual key. More recent efforts have performed better for selection-based techniques reaching rates as high as 18.9 WPM (Shoemaker, Findlater, Dawson, and Booth 2009), the highest mid-air text-entry rate ever achieved before Vulture (Markussen, Jakobsen, and Hornbæk 2014). These studies were limited, some providing no character production on erroneous input (Shoemaker, Findlater, Dawson, and Booth 2009) and others requiring participants to learn a whole new keyboard layout (Shoemaker, Findlater, Dawson, and Booth 2009, Markussen, Jakobsen, and Hornbæk 2013, Hincapié-Ramos, Guo, and Irani 2014, Hincapié-Ramos, Guo, Moghadasian, and Irani 2014, Jones, Alexander, Andreou, Irani, and Subramanian 2010).

2.3.2 Gesture-based

More recent efforts, as technology improved and gestures became more easily detected, have focused on looking at gesture-based techniques for mid-air text-entry. One prominent example was AirStroke (Ni, Bowman, and North 2011), a Graffiti-based alphabet for text-entry, which achieved text-entry rates as high as 11.0 WPM. Implemented by using an Xbox Kinect, it also showed that gestures could be recognized continuously on an interaction plane that was projected in front of the user (Kristensson, Nicholson, and Quigley 2012) and could be used for recognizing gestures for text-entry (Castellucci and MacKenzie 2008). The idea of utilizing an interactive gesture-space in front of the user has been further supported by recent research using the Leap Motion controller (Hari Haran 2014, Guinness 2015). Other alternative gesture-based techniques have attempted to use handwriting in mid-air (Vikram, Li, and Russell 2013, Amma, Georgi, and Schultz 2012, Schick, Morlock, Amma, Schultz, and Stiefelhagen 2012, Shengli, Zhuxin, Li, and Chung 2008). However, even analog handwriting is known to be slow, limited to around 15 WPM (Devoe 1967).

2.3.3 Word-gesture Keyboards

Markussen et al. (2014) developed the first and only mid-air word-gesture keyboard, Vulture. The Vulture keyboard was able to achieve the highest text-entry rates seen for mid-air in two studies, reaching 20.6 WPM with repeated sessions and 28.1 WPM with training on single, short phrases (Markussen, Jakobsen, and

Hornbæk 2014). Vulture used one of the few publicly available and better-known implementations of word-gesture keyboards that was developed for phones, the SHARK[2] (Kristensson 2007, Zhai and Kristensson 2004), proving that word-gesture keyboards, implemented with traditional methods, work well for mid-air text-entry. However, Vulture did have some limitations; participants were required to use a glove with reflective markers for tracking pinching-based gestures over an orthogonally projected display (Markussen, Jakobsen, and Hornbæk 2013). Requiring the user to wear a glove with reflective markers violates the "Come As You Are" design principle, explained in Section 2.2. Additionally, an orthogonally projected display requires the user to be forced to be directly in front of the display (Markussen, Jakobsen, and Hornbæk 2013), whereas users should have the freedom to calibrate and move their interaction space anywhere (Wachs, Kölsch, Stern, and Edan 2011).

2.4 Word-gesture Keyboards

Word-gesture keyboards have seen their emergence along-side the growth of touch-based, hand-held devices, now being shipped as the standard keyboard on many smart phone devices. Originally, word-gesture keyboards were known as shape-writing keyboards because of the process of tracing a word's gesture-shape as opposed to conventional single-input or multi-tap text-entry (Kristensson 2007, Zhai and Kristensson 2012, Zhai, Kristensson, Gong, Greiner, Peng, Liu, and Dunnigan 2009, Zhai and Kristensson 2004, Zhai and Kristensson 2003). Traditional word-gesture keyboard implementations typically see text-entry rates as high as 25 WPM after 35 minutes of practice for beginners and up to around 46 WPM for well-practiced phrases or expert-level experience (Kristensson 2007, Zhai, Sue, and Accot 2002). These text-entry rates are the current standard for traditional, touch-based, word-gesture keyboards. However, it should be noted that these rates can be further increased, as shown by KeyScretch (Fuccella, De Rosa, and Costagliola 2014). KeyScretch is a gesture-based text-entry method for touch screens based on a menu-augmented, word-gesture keyboard and was able to achieve text-entry rates reaching 44-50 WPM, dependent on the language.

A traditional part of implementing word-gesture keyboards is to include shape-recognition algorithms in order to best determine what words the user was trying to create (Kristensson 2007, Zhai and Kristensson 2012, Zhai, Kristensson, Gong, Greiner, Peng, Liu, and Dunnigan 2009, Zhai and Kristensson 2004, Zhai and Kristensson 2003). A word-recognition based approach has been proven to work for word-gesture keyboards (e.g., SHARK^2, Vulture) (Kristensson 2007, Zhai and Kristensson 2004, Markussen, Jakobsen, and Hornbæk 2014). Therefore, a traditional word-gesture implementation was outside the scope and goals of this thesis. Because it was known in advance what word-gestures a user was going to attempt to produce, it was believed a word-gesture keyboard could be pseudo-implemented without word-recognition. This omission is detailed in Section 3.1.1 To ensure that the results were accurate and to analyze the effectiveness of a pseudo-implementation of a word-gesture keyboard, both a pinching-based and touch-based implementations were also examined.

2.5 Text-entry Evaluation

2.5.1 Text-entry Rates

The most important measure for testing the various methods of word separation was by monitoring the text-entry rate. Achieving text-entry rates on par with or greater than those produced in Vulture will shed light on improved techniques for mid-air, text-entry interactions. Text-entry rate was measured in Words Per Minute (Bi, Chelba, Ouyang, Partridge, and Zhai 2012). However, the time to produce the first character must be included when timing word gestures (MacKenzie and Tanaka-Ishii 2007).

2.5.2 Error Rates

The Vulture keyboard uses Minimum Word Distance to primarily detect error rates in the phrases that were transcribed (Markussen, Jakobsen, and Hornbæk 2014). However, due to the pseudo-implementation of the word-gesture keyboard and the fact that characters and errors could be produced mid-gesture, some character-level error rates were also observed. These error rates were determined using the Keystrokes Per Character, Minimum String Distance, and Total Error Rate (Soukoreff and MacKenzie 2003) methods. The Keystrokes Per Character method was not without its flaws, as noted by Soukoreff and MacKenzie (2003). However, because characters were produced during the gesturing process, this might give some insight on the production rate of erroneous characters when using the pseudo word-gesture keyboard implementation.

2.5.3 Correctness

Word-gesture keyboards heavily rely on shape-recognition algorithms in order to determine which words are recognized and produced. However, many studies on mid-air keyboards analyze error rates of the final transcribed text rather than evaluate the correctness between the user-generated gesture and the shape of the intended-word. The Fréchet Distance gives us the ability to do just that. The Fréchet Distance between two curves P and Q, or in this case gesture-shapes, was best defined as the minimum length leash needed to walk a dog when the person walks along P and the dog walks along Q (Har-Peled and Raichel 2014). The lower the calculated distance was between the two paths, the closer the two paths produced were to each other, indicating the correctness of the generated paths.

2.6 Summary

The literature review has covered a variety of important past works that have influenced the direction of this thesis to apply different methods of word separation to mid-air, word-gesture keyboards. Mid-air pointing has played an important role in mid-air selection and manipulation of on-screen objects by casting rays onto a distant display screen. Barehanded interaction is the ideal interaction method to produce natural gestures and follows the "Come As You Are" design principle. Mid-air text-entry and it's evolution over time have shown how techniques from touch-based text-entry have been applied in mid-air. Table 2.1 shows how rates have changed for the various text-entry techniques. The innovative boom of word-gesture keyboards on touch-based devices has changed the way that users are able to enter-text with a single hand. Finally, various text-entry evaluation techniques were identified to help analyze the usefulness of the different word separation techniques presented in this thesis.

Table 2.1. Summary of text-entry rates and methods examined in the literature review.

Type	Method	Text-entry Rate (WPM)	Reference
Touch	Handwriting	≈ 15.0	(Devoe 1967)
	Word-gesture Keyboard	≈ 46.0	(Zhai, Sue, and Accot 2002)
	KeyScretch	≈ 50.0	(Fuccella, De Rosa, and Costagliola 2014)
Mid-air	AirStroke	11.0	(Ni, Bowman, and North 2011)
	Projected QWERTY	13.2	(Markussen, Jakobsen, and Hornbæk 2013)
	Distant QWERTY	18.9	(Shoemaker, Findlater, Dawson, and Booth 2009)
	Vulture	28.1	(Markussen, Jakobsen, and Hornbæk 2014)

CHAPTER THREE

Keyboard Design

3.1 Word-gesture Keyboard Implementation

3.1.1 Lacking Word-recognition

Recreating a word-gesture keyboard from scratch with the limited, non-commercial information available was considered to be outside of the scope of this thesis. Word-recognition has already been proven to work and benefits word-gesture keyboards, including mid-air, by providing high recognition rates of word-gesture shapes (e.g., low error rate and high text-entry rates) (Kristensson 2007, Zhai and Kristensson 2012, Zhai, Kristensson, Gong, Greiner, Peng, Liu, and Dunnigan 2009, Zhai and Kristensson 2004, Zhai and Kristensson 2003, Markussen, Jakobsen, and Hornbæk 2014). Word-recognition is a requirement for traditional word-gesture keyboards because the intended word-gestures that are being generated are *unknown*. However, the word-gestures that were generated in this thesis were *known* in advance. Because the software has preexisting knowledge of the intended word-gesture shapes, it was concluded that word-recognition was not absolutely necessary to build a fully featured word-gesture keyboard. Therefore, a pseudo word-gesture keyboard implementation based on the presented gesture-shapes was used.

Using a pseudo word-gesture keyboard implementation presented its own challenges and limitations. One possible limitation was that character production, including detected errors, occurred mid-gesture. This meant that participants were more likely to stop mid-gesture and interrupt the gesturing-process to correct errors rather than following through with the full word-gesture shape as in traditional word-gesture keyboards. This was expected to slow text-entry rates. Another possible issue was that since the pseudo word-gesture keyboard did not implement shape-recognition, the gesture-path had to be analyzed as it was being drawn. Therefore, key "presses" had to be interpreted through path changes or by the user passing through the expected key. Section 3.1.2 explains how this works in greater detail. Lacking shape-recognition could lead to higher error rates. Additionally, because gesture-shapes were not recognized against a compendium of common words, these limitations allowed user-generation of non-words (e.g., words not in any dictionary). Although these limitations existed, the results of this thesis can be extrapolated to traditional word-gesture keyboards. This claim is justified by the Leap Pinch-air Keyboard (M = 11.3 WPM) performing consistently with the pinching-method from Vulture (M = 11.8 WPM) for text-entry rates in a single session with no training (Markussen, Jakobsen, and Hornbæk 2014). Additionally, the Leap Pinch-air Keyboard reached 58% of the text-entry rate of direct touch input, which was proportional to Vulture at 59% of the text-entry rate of direct touch input (Markussen, Jakobsen, and Hornbæk 2014). Many of these changes and limitations occurred on the back-end (software) while still presenting a similar experience for the user to traditional word-gesture keyboards.

3.1.2 Design

The pseudo word-gesture implementation was created by analyzing the user's generated gesture compared to the expected gesture as it was being drawn to determine which keys were being pressed. The assumptions for detecting a key press were based off the known word being gestured and noticeable deviations made in the gesture's direction. To reduce the chance of erroneous keys being pressed, the sizes of the characters expected to be pressed were exaggerated and the deviation threshold in gestures were lowered between key paths.

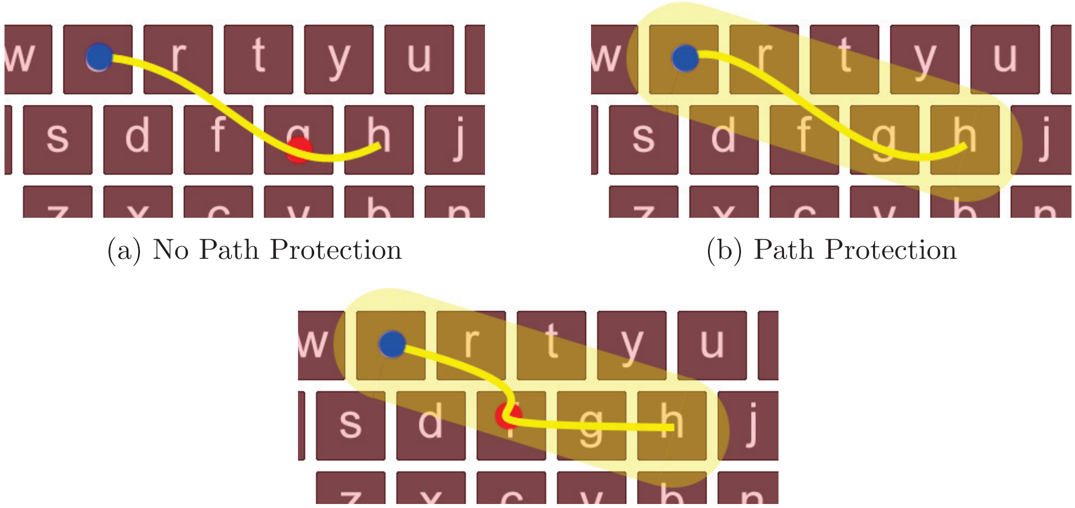
The displayed keys were 64x64 pixels in size with a gap of 10 pixels between each key. The actual size of the keys was dependent on the display device being used. The next expected letter to be pressed in a word was changed into a circular key with

Figure 3.1. The next expected key to be pressed was increased in size to make pressing it easier. The above visual shows how it was interpreted by the software, but there was no visual feedback presented to users.

a radius of 76.8 pixels, 20% larger than the key widths, in order to make hitting keys even easier. Figure 3.1 shows how keys were changed behind the scenes. However, the software presented no visual feedback to the participant of the increased key sizes.

An interpolated trail with points at a minimum of 16 pixels apart was used in determining deviations in word-gesturing. The angle of detecting a deviation was 165 degrees for all areas that were not on the expected path to the next key and was 90 degrees while on the expected path. Deviations in gesture path had to be at least 48 pixels away from each other to be counted as a "press". The expected path between two keys comprised an area from the previous expected key to the next expected key with a width 62.5% larger than key size, or 104 pixels. Figure 3.2 shows how the expected path protects against natural deviations when moving from one key to the next.

The specific values for detecting key presses were found using trial and error. These were used to create an experience as close to a traditional word-gesture keyboard as possible. Whereas the word-recognition implementation showed the transcribed word after the completed gesture, the pseudo-implementation showed participants real-time updates of detected character presses along the keyboard path. This was determined to be an acceptable limitation as explained in Section 3.1.1.



(a) No Path Protection

(b) Path Protection

(c) Error with Path Protection

Figure 3.2. A path between the currently pressed letter and the next letter significantly reduces the chance of detecting erroneous input. (a) shows an error detected with an angle less than 165 degrees; (b) shows how as long as the user stayed on the path, errors were significantly reduced; and (c) shows that errors could still be detected on a protected path if a deviation with an angle less than 90 degrees was detected.

3.1.3 Display

3.1.3.1 Keyboard layout. The keyboard layout, seen in Figure 3.3, was a typical QWERTY keyboard with key sizes of 64x64 pixels and gaps of 10 pixels. All special keys and number keys were removed to simplify the keyboard and a backspace key added to the keyboard's right side to allow for erroneous character deletion.

3.1.3.2 Text area. Figure 3.4 shows how two text areas were used to display text to participants. The top text-area displayed a presented word, shown in Figure 3.4a, and the bottom text-area displayed the presented word's transcription, shown in Figure 3.4b. Both the presented word and transcription's characters were colored green when correctly matched. If errors were made during transcription, only characters in the transcribed text would display in red. The participant could then

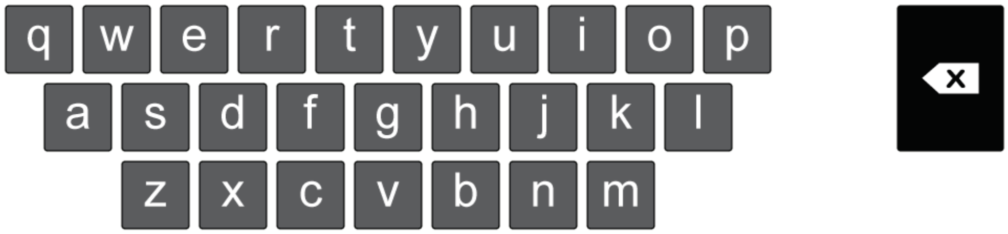
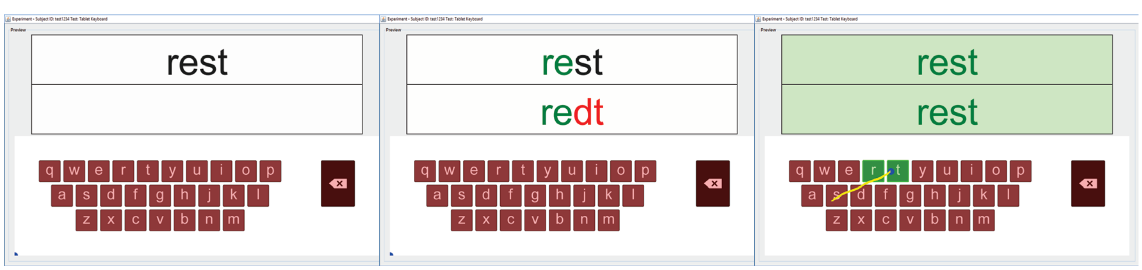


Figure 3.3. The keyboard layout used during the full study.

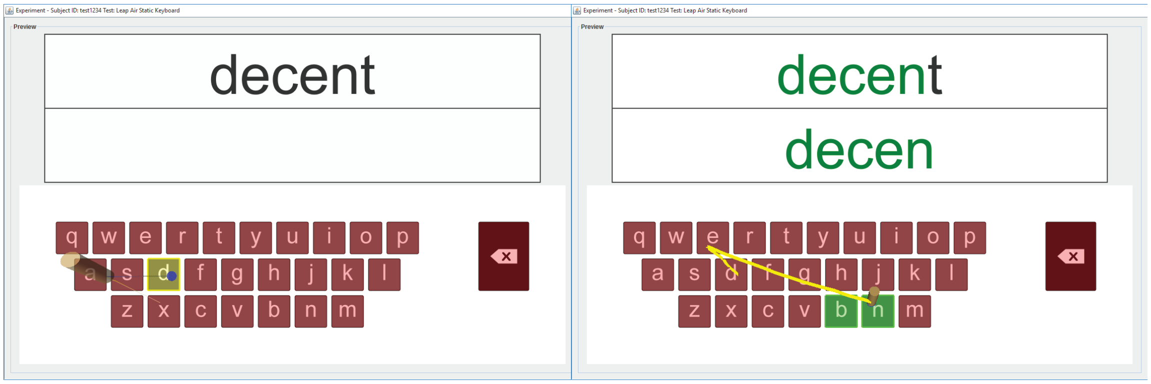
use the backspace to correct errors. When the transcription matched the presented word, both text-areas were highlighted in green, as seen in Figure 3.4c.

3.1.3.3 Real-time updates. As a participant was drawing the gesture-shape of a word, their progress was tracked in real-time, as shown in Figure 3.5. For the keyboards that tracked a finger or stylus, the software displayed a cylinder to indicate the input's position and direction in 3-dimensional space relative to the virtual keyboard as shown in Figure 3.5a. In addition, it displayed which letter the user's input hovered over by projecting a blue dot to the corresponding virtual



(a) Displayed Word (b) Transcribed Error (c) Completed Word

Figure 3.4. Examples of how the text areas change when showing transcribed text. (a) shows the word to be transcribed as it first appears; (b) shows how correct letters were colored green in both text areas and transcription errors were colored red; and (c) shows the user-generated text and presented word highlighted to indicate correct transcription.



(a) Prior to pressing ‘d’ (b) During the word-gesture

Figure 3.5. Examples of the real-time display for word-gesturing. (a) shows the user just about to press the first character; and (b) shows the word gesturing process for the word “decent.”

keyboard location. As seen in Figure 3.5b, the software showed the participant the gesture-path that they were traveling in addition to the letters that had been pressed. The gesture-trail decayed over time in order to not clutter the display.

3.1.4 Calibration

Each mid-air keyboard, and the Leap Surface Keyboard, could be calibrated in a manner similar to Personal Space (Hari Haran 2014), seen in Figure 3.6. Many of the calibration spaces, however, required direct interaction instead of projecting the inputs onto the interaction plane. Default calibrations were adequate for most participants, but recalibration was optional. However, calibration had less of a lasting effect because many participants repositioned the Leap Motion controller itself. Participants were encouraged to adjust the controller’s position to promote usability. This was sometimes a greater factor than motor space calibration for translating precise movement to the virtual keyboard. Because this thesis was not an accessibility study, participants were allowed to calibrate the keyboard with their arms rested or raised. Therefore, this thesis did not address the “Gorilla Arm Syndrome” mentioned in Section 5.8.3.

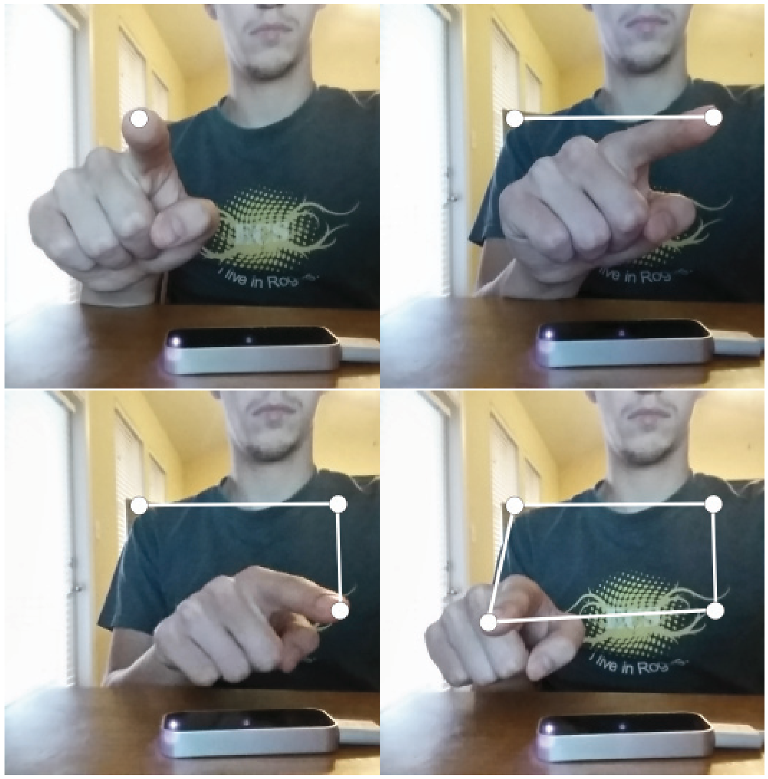
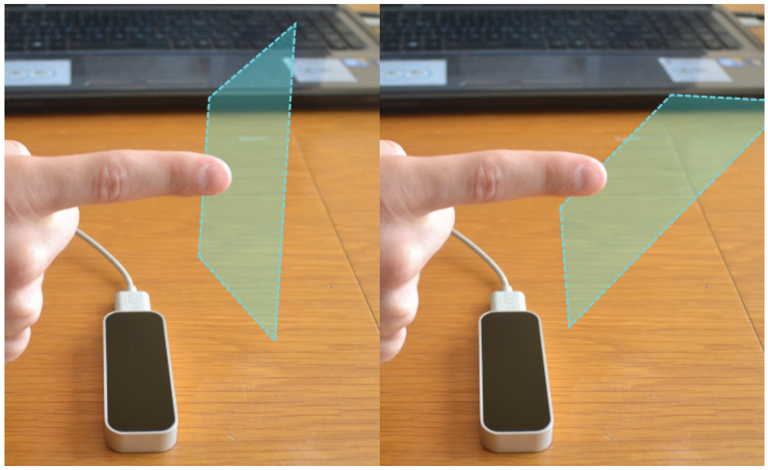


Figure 3.6. Participants were able to follow on-screen instructions to calibrate the interaction space using their finger.

3.1.4.1 Motor space and display space. The researcher attempted to place the Leap Motion controller to adjust the calibrated interaction plane, or the motor space, to be as parallel to the screen, or display space, as possible. However, participants were allowed to adjust the Leap Motion controller to a position that felt most comfortable to them. Moving the controller typically resulted in the motor space being oriented perpendicular to a participant's arm rather than parallel to the display space. Also to note, when working with keyboards that fully utilized the 3rd-dimension, an interaction plane angled away from a participant was sometimes more effective than a straight plane perpendicular to the floor. Figure 3.7 shows the difference between a straight plane and angled plane.

The size of the motor space was dependent on either the device the keyboard was presented on or the calibration of the keyboard's interaction plane. For all of the



(a) Straight Plane (b) Angled Plane

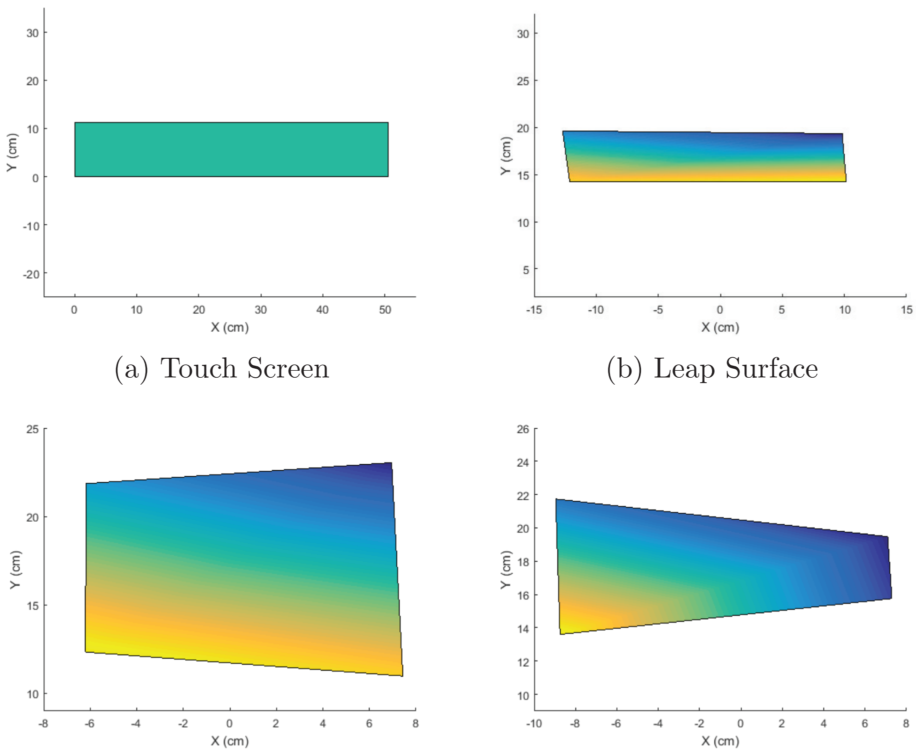
Figure 3.7. Examples of a straight plane versus an angled plane.

keyboards, the motor space was mapped to a display space of 952x212 pixels and keys that were 64x64 pixels with gaps of 10 pixels. The real-world size of the keyboard display was dependent on the screen being used. Figure 3.8 shows the average sizes of the motor spaces.

3.1.5 Dictionary Creation

For the purposes of this thesis, the term "dictionary" denotes the pool of words presented to the participant. To make each keyboard experience as similar as possible, a custom dictionary was created for each keyboard interaction style. While different words were used for each dictionary, the words were selected by using a custom gesture-shape dissimilarity algorithm. This algorithm minimized dissimilarity between word gesture-shapes as shown in Appendix A. The algorithm's results were further reduced to common words between 3 and 6 characters in length. These unique word sets became each keyboard's dictionary.

3.1.5.1 Deviating from the standard phrases. To evaluate text-entry, typically predefined phrases were generated and used (MacKenzie and Soukoreff 2003).



(a) Touch Screen

(b) Leap Surface

(c) Static/Predictive/Bimodal

(d) Leap Pinch-air

Figure 3.8. The average size of the motor spaces for each keyboard with a gradient color scale showing the plane orientation. The closest areas are represented by *yellow* and the farthest by *blue*. (a) shows the standard Touch Sceen motor space; (b) shows the average calibration for the Leap Surface motor space; (c) shows the average calibration for the Leap Static-air, Predictive-air, and Bimodal-air motor spaces (a single calibration was typically used for all three); and (d) shows the average calibration for the Leap Pinch-air motor space.

However, due to the limited number of trials and the abundance of different conditions, single words were chosen as opposed to randomly selecting from a compendium of predefined phrase sets. The goal was to create new word dictionaries, avoiding whole phrases to prevent confusing language, using a custom algorithm to minimize the dissimilarity of different gesture-shapes. No previous research existed on words with similar gesture-shapes, or their benefit, but this thesis's purposeful deviation created similar keyboard experiences to standardize results across many conditions and few trials.

3.1.5.2 Gesture-shape dissimilarity. Originally, this thesis considered the Fréchet Distance to find similar word gesture-shapes using sets of words with minimal distance between each letter within a gesture-shape. While Fréchet Distance gave acceptable results, there were noticeable differences in *some* of the gesture-shape sets. Figure 3.9 demonstrates these differences, which appears to show more than one primary gesture-shape returning for the set.

In order to achieve gesture-shapes that were more similar than shapes found by the Fréchet Distance, the custom dissimilarity algorithm was created. The words were pulled from the Oxford English Dictionary. The dissimilarity between two words' gesture-shapes was defined by the formula

$$dissimilarity(P,\ Q) = \frac{\sum_{i=2}^{N} \frac{1}{2}\left(\left(\frac{|dist(P_i,\ P_{i-1}) - dist(Q_i,\ Q_{i-1})|}{max\ distance}\right) + \left(\frac{angle(P_i - P_{i-1},\ Q_i - Q_{i-1})}{\pi}\right)\right)}{N-1} \tag{3.1}$$

where P and Q were two words of N characters in length, i was a particular character of P or Q, P_i and Q_i were the vector locations on the virtual keyboard, *max distance* was the maximum distance between any two letters on the virtual keyboard, $dist(...)$

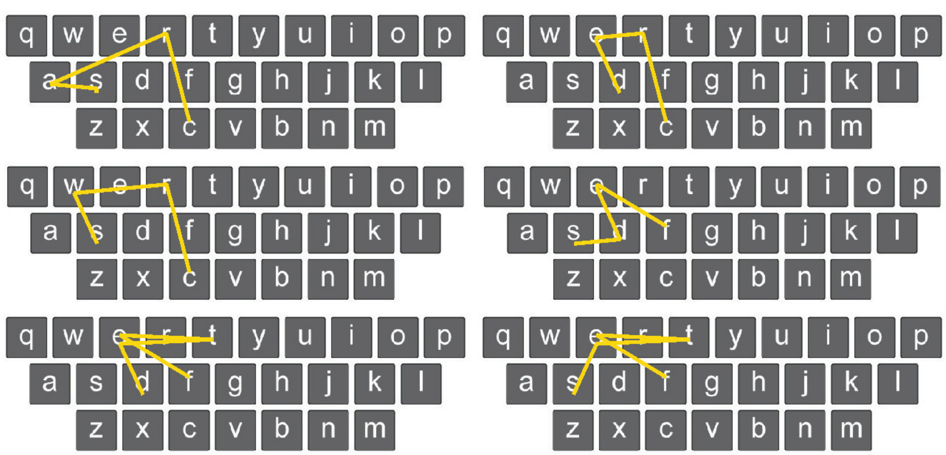


Figure 3.9. An example of a gesture-shape set generated by the Fréchet Distance algorithm: 'crass', 'creed', 'crews', 'feeds', 'feted', 'fetes'. The Fréchet Distance, at times, generated more than one gesture-shape pattern per word set.

was the distance between two vector locations, and $angle(...)$ was the angle between two vectors. The dissimilarity formula generated values between the range [0, 1] and treated every pair of paths between two letters of two words with equal weight. The objective was then to find the sets of words with the lowest dissimilarity.

3.2 Word-gesture Keyboards

All of the word-gesture keyboards created used the same word-gesturing implementation but differentiated by their interaction method and how touch was handled as a delimiter between words.

3.2.1 Touch Screen Keyboard

3.2.1.1 Interaction method. The Touch Screen Keyboard was implemented to mirror the de facto method for word-gesture keyboards, which are generally touch-based. The user interacts directly with a touch screen surface.

3.2.1.2 Word separation. Figure 3.10 shows that word separation for the Touch Screen Keyboard worked in the same way as typical word-gesture keyboards for phones and tablets. Touch was simulated simply by pressing a finger against the surface, drawing the word-gesture, and then removing the finger from the surface.

Figure 3.10. A touch was simulated when the tabletop screen was touched with the user's pointer finger.

3.2.1.3 Size of the motor space. The motor space for the Touch Screen Keyboard was larger than the other keyboard motor spaces because the display device was intrinsically larger and the Touch Screen's motor space and display space are coupled together. The display device used was a C4667PW boasting a 46″ display space and a maximum resolution of 1920x1080 pixels. Figure 3.11 shows the C4667PW, a 3M™ Multi-touch Display. When scaled for the maximum resolution, the Touch Screen display space and motor space were both 50.49x11.24 *cm*, with keys that were 3.39x3.39 *cm* and gaps between keys of 0.53 *cm*. If higher resolutions were available, a higher resolution would have been chosen to decrease the overall size of the display space and motor space to match the other keyboards more closely. Though larger than desired, the 3M™ Multi-touch Display was still preferred over using very small touch-based, word-gesture keyboards such as those on phones or tablets. A similar sized motor space helped standardize results between touch and mid-air.

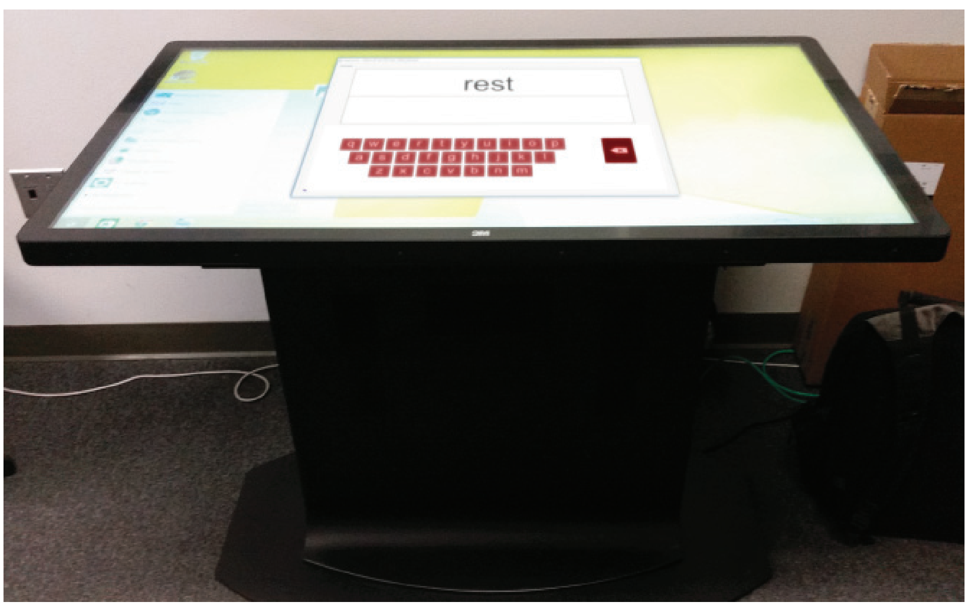


Figure 3.11. The 46″ C4667PW, a 3M™ multi-touch tabletop display.

3.2.2 *Leap Surface Keyboard*

3.2.2.1 Interaction method. The Leap Surface Keyboard used the Leap Motion controller to track a wooden stylus for interaction. It was designed so that it would simulate a touch screen using a mid-air plane projected onto a surface. This was done by inserting the Leap Motion controller into a custom holder, shown in Figure 3.12, and projecting the mid-air keyboard over a keyboard printed on paper. As an added note, the Leap Surface Keyboard works in the exact same way as the Static-air Keyboard in Section 3.2.3 by being calibrated to a surface rather than mid-air. A stylus was chosen to be used as an interaction tool to allow for accurate surface emulation because the Leap Motion controller was in a position that made it difficult to successfully track a participants hand or finger. Unfortunately, the Leap Controller hardware, at the time of this thesis, was only designed to recognize hands from one direction, necessitating the controller be positioned at the bottom of the holder rather than the top.

Figure 3.12. The custom built holder, projecting an interaction plane onto a printed keyboard surface.

Figure 3.13. A touch was simulated when the stylus hits the paper surface.

3.2.2.2 Word separation. Word separation for the Leap Surface Keyboard worked in a similar way to how touch was simulated using a stylus for a phone or tablet. Figure 3.13 shows how touch was simulated by pressing the tip of the stylus against the surface of the printed keyboard. The word-gestures were drawn and then the stylus removed from the surface to complete the action.

3.2.2.3 Size of the motor space. Figure 3.8b shows the average calibrated motor space for the Leap Surface Keyboard. The average keyboard was 22.28x5.41 *cm*, with keys that were 1.50x1.50 *cm* and gaps between keys of 0.23 *cm*.

3.2.3 Leap Static-air Keyboard

3.2.3.1 Interaction method. The Leap Static-air Keyboard used the Leap Motion controller to track the pointer finger of either hand for interaction. It was designed to simulate a virtual touch screen in mid-air by projecting a quadrilateral plane directly above the interactive surface. The pointer finger would then be used to penetrate the plane to simulate touch.

3.2.3.2 Word separation. Word separation for the Leap Static-air Keyboard worked in a similar way as any ordinary touch-based word-gesture keyboard, However, the simulated touch plane was in mid-air. Touch was simulated by using either pointer

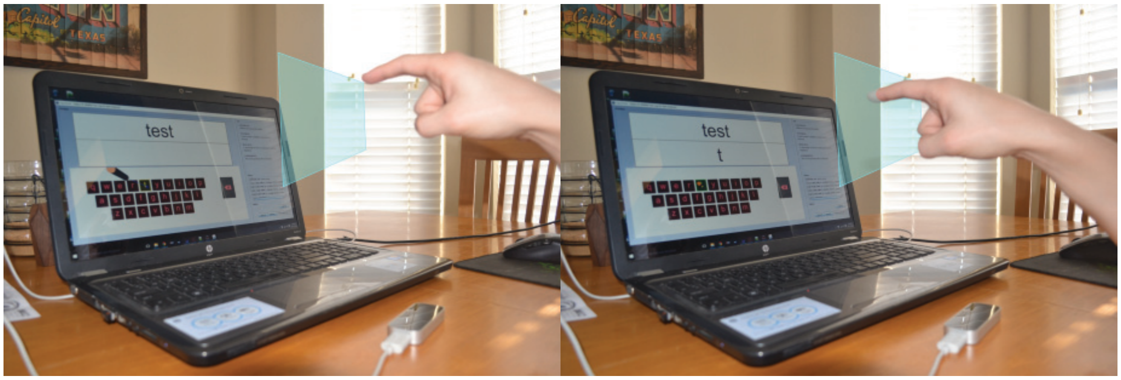


Figure 3.14. A touch was simulated by penetrating the interaction plane.

finger and penetrating the mid-air interaction plane as seen in Figure 3.14. While maintaining the intersection, the pointer finger was used to draw the word-gesture. By pulling the finger away from the mid-air interaction plane, touch was released.

3.2.3.3 Size of the motor space. Figure 3.8c shows the average calibrated motor space for the Leap Static-air Keyboard. The average keyboard was 13.71x10.07 *cm*, with keys that were 0.92x0.92 *cm* and gaps between keys of 0.14 *cm*, the same as the Predictive-air and Bimodal-air keyboards.

3.2.4 Leap Predictive-air Keyboard

3.2.4.1 Interaction method. The Leap Predictive-air Keyboard used the Leap Motion controller to track the pointer finger of either hand for interaction. It was designed to simulate a virtual touch screen by projecting a quadrilateral plane in the air. However, instead of having to interact with a static, unchanging plane, the Predictive-air Keyboard associates the interaction plane with the participant's pointer finger. As the pointer finger moves forward or backward, the plane follows. By analyzing forward and backward hand gestures in the z-direction, the Predictive-air Keyboard tries to move the interaction plane to the pointer finger by predicting when a touch was simulated. Slow gestures serve mostly to move the plane, whereas quick gestures generally snap the plane to the pointer finger. The Leap Motion API provided the predictor values for forward and backward hand gestures.

3.2.4.2 Word separation. Word separation for the Predictive-air Keyboard worked in a similar way as any ordinary touch-based word-gesture keyboard. However, the simulated touch plane was in mid-air. This plane was kept at a consistent distance away from the tracked pointer finger until a forward hand gesture was detected, simulating a touch. The pointer finger could then be used to draw the word-gesture until it was completed. Finally, by making a backward hand gesture away from the interaction plane, the simulated touch was released. The plane interaction was visually similar to Figure 3.14, the Leap Static-air Keyboard interaction.

3.2.4.3 Size of the motor space. Figure 3.8c shows the average calibrated motor space for the Leap Predictive-air Keyboard. The average keyboard was 13.71x10.07 *cm*, with keys that were 0.92x0.92 *cm* and gaps between keys of 0.14 *cm*, the same as the Static-air and Bimodal-air keyboards.

3.2.5 Leap Bimodal-air Keyboard

3.2.5.1 Interaction method. The Leap Bimodal-air Keyboard was designed to utilize two inputs: the Leap Motion controller and a standard keyboard. The Leap Motion controller tracked the pointer finger of either hand by projecting a quadrilateral plane in the air and snapping the movements of the pointer finger to the plane. A touch was simulated by using the secondary input; in this case, a standard keyboard's space bar.

3.2.5.2 Word separation. In order to move from one word to the next for the Leap Bimodal-air Keyboard, the user activated a secondary input: the standard keyboard's space bar. The interaction plane for simulated touch, as seen in Figure 3.15, was still projected in mid-air. Touch was simulated by using either pointer finger to

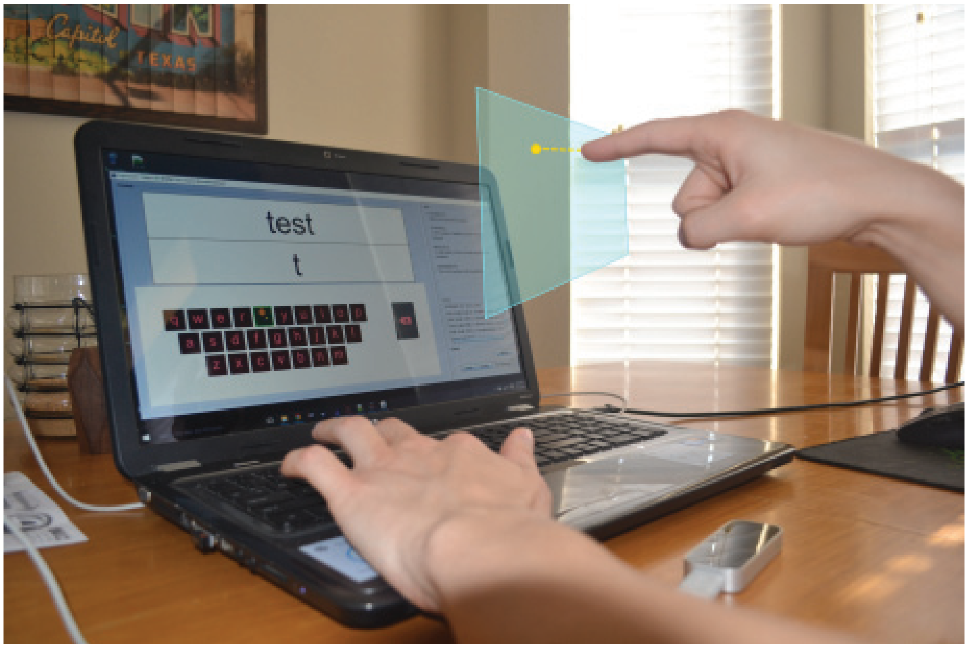


Figure 3.15. A touch was simulated by pressing the space bar on the keyboard.

determine the position over the interaction plane in the x and y directions and then by pressing and holding the space bar. While holding down the space bar, the pointer finger was used to draw the word-gesture and finally the space bar was released to end the touch.

3.2.5.3 Size of the motor space. Figure 3.8c shows the average calibrated motor space for the Leap Bimodal-air Keyboard. The average keyboard was 13.71x10.07 *cm*, with keys that were 0.92x0.92 *cm* and gaps between keys of 0.14 *cm*, the same as the Static-air and Predictive-air keyboards.

3.2.6 Leap Pinch-air Keyboard

3.2.6.1 Interaction method. The Leap Pinch-air Keyboard used the Leap Motion controller to track the palm of either hand for interaction. It was designed to project a quadrilateral plane in mid-air and snapped the palm position to the plane in the z-direction. The hand could then be used to form a pinch-gesture to simulate touch. The Leap Motion API provided the predictor values for recognizing pinching-gestures. It is important to note that unlike in Vulture (Markussen, Jakobsen, and Hornbæk 2014), no glove was required and many different pinch-gestures were recognized.

3.2.6.2 Word separation. A pinching-gesture was used to move to the next word in the sequence for the Leap Pinch-air Keyboard. However, the interaction plane was still projected in mid-air. Touch was simulated by using either hand and then forming and holding a pinching-gesture, as shown in Figure 3.16. While pinching, the word-gesture was drawn and then released to end the "touch."

3.2.6.3 Size of the motor space. Figure 3.8d shows the average calibrated motor space for the Leap Pinch-air Keyboard. The average keyboard was 16.86x9.04 *cm*, with keys that were 1.13x1.13 *cm* and gaps between keys of 0.18 *cm*.

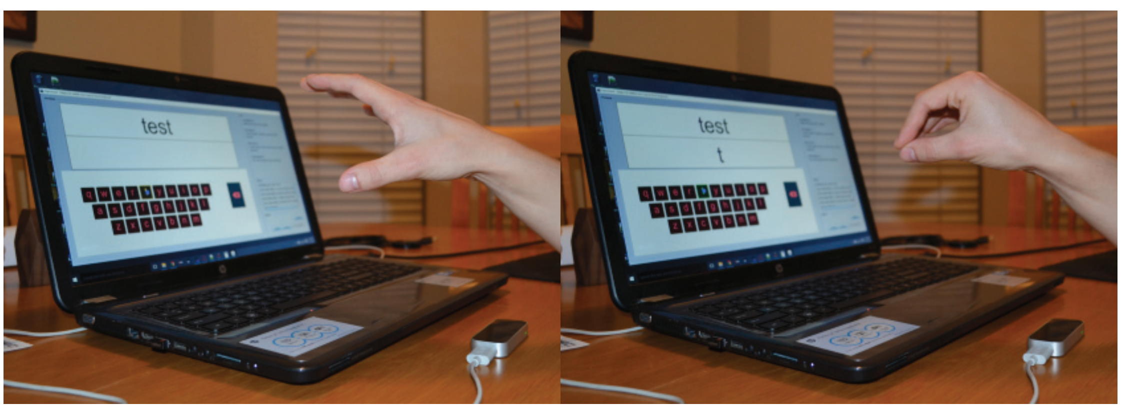


Figure 3.16. A touch was simulated by making a pinching gesture.

CHAPTER FOUR

Pilot Studies

4.1 Pre-pilot Study

The pre-pilot was a very informal study aimed at assessing the functionality of the pseudo word-gesture keyboard implementation. This informal study was used to also test the conceptual feasibility of using 3-dimensions as a means of word separation for simulated touch input. See Chapter 3 for more specific implementation details.

4.1.1 Participants

A sample size of 2 was used in the pre-pilot study. Both participants were male, ages 24 and 25 and both participants had previous experience with gesture-controllers, touch screens, and word-gesture keyboards. Table 4.1 provides the participants' information in detail.

4.1.2 Input Devices and Interaction Styles

The interaction methods used within the pre-pilot study were dependent on the currently active keyboard input forcing the user to only interact with one style of keyboard at a time. All of the keyboards were simulated on a 64-bit Windows 7 work station, with all receivers or controllers connected via USB 2.0. The participants were allowed to recalibrate the active keyboard's interaction plane at any time before starting the main task for a more comfortable experience. Calibration was only possible if applicable to the active keyboard input. Participants were also encouraged to reposition the gesture-controller to best fit their personal preference and were given

Table 4.1. Participant information including age, gender, handedness, computer usage, and previous experiences.

Subject	Gender	Age	Computer Usage per Week (hours)	Handedness	Hand Used in Experiment	Touch-device Experience	Gesture-controller Experience	Word-gesturing Experience	Impairment History
1	m	24	31 - 40	right	right	yes	yes	yes	no
2	m	25	50+	right	right	yes	yes	yes	no

the option to freely rest or raise their elbows during the experiment. More information about the implementation of specific keyboard interactions and calibrations can be found in Chapter 3. Every keyboard was designed for use by either right or left handed participants. Figure 4.1 shows the keyboard layout used for all keyboards other than the Xbox Controller keyboard, whereas Figure 4.2 shows the keyboard used for the Xbox Controller Keyboard.

4.1.2.1 Leap Motion Static-air Keyboard. The Leap Motion Static-air Keyboard used a Leap Motion controller which was placed on the desk in front of the participant. The participant then used a stylus which was tracked by the Leap Motion controller in order to interact with a projected interaction plane. A touch was simulated by the insertion of the stylus into the interaction plane and a release was simulated upon the removal of the stylus. The interaction plane could be calibrated at any time prior to the experiment.

Figure 4.1. The keyboard layout used during the pre-pilot study.

4.1.2.2 Leap Motion Pinch-air Keyboard. The Leap Motion Pinch-air Keyboard also used a Leap Motion controller that was positioned on the desk in front of the participant. The participant then used their bare hand, which was tracked at the center of their palm, to interact with the projected interaction plane. Touch was simulated by having the participant make a pinching gesture with their hand and a release was simulated when the participant released the pinch, opening their hand again. As in the previous Static-air keyboard, the interaction plane could be calibrated at any time prior to the experiment.

4.1.2.3 Leap Motion Surface Keyboard. Again, for the Leap Motion Surface Keyboard, a Leap Motion controller was used for tracking. Unlike the Leap Static-air or Leap Pinch-air keyboards, the gesture controller was placed into a custom designed holder instead of on the desk in front of the participant. The holder was attached to an inclined surface with a printed keyboard fixed on top, as shown in Section 3.2.2. Because identical placement was not guaranteed between uses, the Leap Motion Surface Keyboard required a single calibration after being inserted into the holder to accurately simulate an interaction plane projected onto the printed keyboard. The participant then used a stylus, as before, which was tracked by the Leap Motion controller in order to detect interaction. A touch was simulated by pressing the tip of the stylus against the printed keyboard and a release was simulated when the tip of the stylus was removed from the printed keyboard surface.

4.1.2.4 Xbox Controller Keyboard. The Xbox Controller Keyboard used an Xbox 360 Wireless Controller that transmitted information via the Microsoft Xbox 360 Wireless Receiver for Windows. The participants were required to use the directional sticks or the directional-pad in order to change which key was selected, and then used the 'A' button in order to select the currently highlighted key. The Xbox

Figure 4.2. The keyboard layout used for the Xbox Controller Keyboard.

Controller keyboard only allowed for single-input text entry, functioning in the same way as the default Xbox 360 virtual keyboard.

4.1.3 Task Design

The design for the pre-pilot study included 4 different keyboard interaction styles representing each of the conditions that made up the task. The 4 conditions used were the Leap Motion Static-air, Leap Motion Pinch-air, Leap Motion Surface, and Xbox Controller keyboards.

The task consisted of 30 trials for each of the 4 keyboard interaction styles creating a total of 120 trials per participant. For each trial, a word of 3 to 6 characters in length was chosen at random from the Oxford English Dictionary and displayed on the screen. A blank text-area was positioned directly below the displayed word for the user-generated, transcribed text. Beneath both text areas, there was a virtual representation of the keyboard the participants were using. The participants were then required to use the currently active keyboard interaction style to enter the displayed word using word-gesturing. During the word-gesturing process, the participants were shown real-time updates to the displayed word and transcribed text as well as their movements within the virtual keyboard space. Detected key presses were appended directly to the transcribed text-area with correctly matching letters being colored green and text entry errors being colored red whereas only correctly matching letters were applied to the displayed word. The participants were required to use the active keyboard's backspace key to remove errors. Once a word was correctly entered, the participants were required to press the active keyboard's enter key to move on to the next word.

Small deviations in the above task were required for how the participants interacted with the Xbox Controller keyboard. For this keyboard, there was no word-gesturing feature implemented. Instead, the participant was asked to use single-input text entry using a standard Xbox 360 Controller. The Xbox Controller keyboard was implemented in the same way as a conventional console keyboard. The participants had to press the 'A' button to select a letter, 'Y' button to delete, and the 'start' button to move to the next word.

4.1.4 Experimental Design

The initial pre-pilot study used a within-subjects design without any counter-balancing (Kim 2010). Both participants used every keyboard interaction style in a random order. The strength of a within-subjects design is the overall power increase and reduction in error variance associated with individual differences. The weakness of using the within-subjects design is that it suffers from "carryover effects" (the participation in one condition may affect performance in other conditions) between each keyboard interaction style.

4.1.5 Procedure

There was only a single study visit for each participant in which all tasks were performed. The study visit took between 30 and 45 minutes to complete depending on the number of calibrations. The full set of tasks and their expected durations are detailed in Table 4.2, the pre-pilot schedule of assessments. The participants followed the same process for each of the 4 keyboard interaction styles' tasks.

First, the participants were given a brief verbal explanation and physical demonstration of the active keyboard input. The explanation dialog contained the name of the active keyboard and the method of interaction. The researcher then demonstrated how to enter the word "test" using the active keyboard. The participants were then given the stylus and control of interaction with the keyboard in order to become familiar with the interaction style.

Participants were then instructed to use the keyboard to perform practice words that were randomly chosen from the Oxford English Dictionary between 3 and 6 characters long. No practice words were duplicated and the dictionary was filtered for offensive words. There was no limit placed on how many words could be performed while practicing. The participants were told to continue until they felt they were able to efficiently and comfortably type each word with minimal errors. During the practice phase, the opportunity to optionally recalibrate the interaction-space any number of times was given if applicable to the active keyboard.

Next, the participants performed the task itself. As detailed in Section 4.1.3, the participants were instructed to enter a total of 30 words for the current keyboard interaction style. None of the experiment words were duplicated between themselves or the practice words and again were pulled at random from the Oxford English

Table 4.2. Schedule of Assessments for a single study visit during the pre-pilot study (in minutes).

Step	Controller	Leap Motion Surface	Leap Motion Static-air	Leap Motion Pinch-air	Exit Survey	Total
explain	0.5	0.5	0.5	0.5	0	2
practice	5	5	5	5	0	20
calibrate	0	2	2	2	0	6
task	3	3	3	3	0	12
survey	0	0	0	0	5	5
Total	8.5	10.5	10.5	10.5	5	45

Dictionary between 3 and 6 characters, filtering for swear words. Participants were not allowed to recalibrate the interaction space during the task.

After all tasks were completed for each of the 4 interaction styles, the participants were asked to fill out an exit survey. The exit survey asked the participants for their age, gender, major, and handedness as well as several questions detailing any prior touch, gesture-controller, or word-gesturing experience or impairments that might relate to the study. Finally, the participants were required to fill out the Likert scale relating to difficulty, discomfort, and fatigue experienced when using the interaction styles, as well as rank each interaction style on a numerical scale from best to worst.

4.1.6 Dependent Measures

The pre-pilot study collected qualitative testimonials alongside the exit survey. Participants were encouraged to give feedback of their experience during and after the experiment. The feedback was then used to refine the pseudo word-gesture keyboard implementation and assess the keyboard layout. Additionally, playback data from participants was also recorded. The playback data included detected key presses, the calibrated interaction plane, and the tracking location data. However, no empirical metrics intended for evaluating the keyboards' performances were observed.

4.1.7 Pre-Pilot Summary

The informal pre-pilot study allowed for assessing the functionality of the pseudo word-gesture keyboard implementation through qualitative feedback. The pseudo-implementation was observed to function adequately. In addition, many of the remaining bugs were revealed and corrected. The keyboard design was observed to be too cluttered; as a result, it was redesigned to be sleek and simple. Figure 4.1 shows the keyboard layout used in the pre-pilot study, whereas Figure 3.3 shows the final keyboard layout. A higher sensitivity to errors was observed after an initial error was made because path protection would be disabled; therefore, error correction was modified to be enforced for all later studies. A participant complained of fatigue during the study; for this reason, the number of trials was reduced for the pilot study.

4.2 Pilot Study

The pilot study expanded on what was learned from the pre-pilot. Additional mid-air keyboard interactions were added and the displayed virtual keyboard was redesigned to be simpler and remove obtrusive features.

4.2.1 Participants

A sample size of 7 was used for the pilot study. There were 3 male and 4 female participants, ages ranging from 21 to 24 with a median age of 22. Participants' computer usage ranged from 6 to greater than 50 hours per week with a median usage of 21 to 30 hours per week. All of the participants described their right hand as being dominant and all participants used their right hand during the experiment except for one participant who switched back and forth. All of the participants had previous experience with touch devices. All but one participant had previous experience with gesture-controllers and only 57% had previous experience with word-gesture keyboards. No participants had any impairment that affected their ability to enter text with computers. Table 4.3 provides the participants' information in detail.

Table 4.3. Participant information including age, gender, handedness, computer usage, and previous experiences.

Subject	Gender	Age	Computer Usage per Week (hours)	Handedness	Hand Used in Experiment	Touch-device Experience	Gesture-controller Experience	Word-gesturing Experience	Impairment History
1	male	21	21 - 30	right	right	yes	yes	no	no
2	male	24	41 - 50	right	right	yes	yes	yes	no
3	female	22	50+	right	right	yes	yes	yes	no
4	female	23	50+	right	right	yes	yes	yes	no
5	female	21	6 - 10	right	both	yes	yes	no	no
6	female	24	6 - 10	right	right	yes	no	no	no
7	male	21	21 - 30	right	right	yes	yes	yes	no

4.2.2 *Input Devices and Interaction Styles*

The pilot study saw the introduction of three additional keyboard inputs: the Touch Screen Keyboard, the Leap Motion Bimodal-air Keyboard, and the Leap Motion Predictive-air Keyboard. The Touch Screen Keyboard was added because it is the de facto interaction for modern word-gesture keyboards (Kristensson 2007). The other two keyboards were added as alternative implementations to mid-air, word-gesture keyboards.

As in the pre-pilot, participants interacted with the different keyboard interaction styles one at a time. All of the keyboards except for the Touch Screen Keyboard were simulated on the same 64-bit Windows 7 work station as before. The Touch Screen Keyboard was simulated on the C4667PW, a 3M™ Multi-touch Display, running 64-bit Windows 8. Again, all receivers or controllers were connected through USB 2.0. The participants were again allowed to recalibrate the active keyboard's interaction plane. Again, participants were encouraged to reposition the gesture-controller and were given the option to use either hand and rest or raise their arms during the experiment.

4.2.2.1 Touch Screen Keyboard. The Touch Screen Keyboard was used on a large tabletop touch screen. The participant then used their finger to interact with the virtual keyboard on the screen in the same way as typical touch devices. Touch was simulated when the participants finger touched the screen and release was simulated when the finger was lifted from the surface.

4.2.2.2 Leap Motion Bimodal-air Keyboard. The Leap Motion Bimodal-air Keyboard used a Leap Motion controller that was placed on the desk in front of the participant. The participant then used a stylus which was tracked by the Leap Motion controller in order to determine its location over the projected virtual keyboard. A touch was simulated by pressing the space bar key on a standard QWERTY keyboard and a touch release was simulated upon the release of the space bar. The interaction plane could be calibrated at any time prior to the experiment.

4.2.2.3 Leap Motion Predictive-air Keyboard. The Leap Motion Predictive-air Keyboard used a Leap Motion controller that was, again, placed on the desk in front of the participant. The participant then used a stylus, which was tracked by the Leap Motion controller, in order to interact with a projected interaction plane. A touch was simulated by recognizing and predicting a forward gesture of the stylus toward the interaction plane and a release was simulated by recognizing a backward gesture away from the interaction plane. As before, the interaction plane could be calibrated at any time prior to the experiment.

4.2.3 Task Design

As in the pre-pilot, the conditions of the task were represented by the 7 different keyboard interaction styles. The 7 conditions used were the Leap Motion Static-air, Leap Motion Pinch-air, Leap Motion Surface, and Xbox Controller keyboards as before, with the addition of the Leap Motion Predictive-air, Leap Motion Bimodal-air, and Touch Screen keyboards.

Task profiles were created for each of the 7 keyboard interaction styles. Each task profile consisted of 10 separate trials for a total of 70 trials per participant. The reduction in trials from 30 words to 10 words for each keyboard was due to a complaint of fatigue during the pre-pilot study; one of the participants was unable to finish. The addition of task profiles were an attempt to standardize the data collected rather than randomizing words between uses of the same keyboard. Instead of choosing 10 words at random for each and every keyboard and participant, the task profiles insured that the same 10 words were used across each unique interaction style for all participants. This was handled by generating static, unchanging dictionaries for each keyboard, guaranteeing a total of 70 unique words as opposed to 490 unique words for the 7 participants. The 10 words selected for each dictionary were generated by a custom dissimilarity algorithm that produced the top 10 least dissimilar gesture-shapes across all words in the Oxford English Dictionary for words with 3 to 6 characters in length. This meant that only 10 different gesture-shapes were used by each participant across all interaction styles, ensuring that all participants' experiences with each keyboard were as similar as possible to each other and other participants. The creation of these dictionaries is detailed in Section 3.1.5.

For each trial, a word was chosen at random from the active keyboard's previously constructed dictionary and displayed on the screen. A blank text-area was positioned directly below the displayed word for the participants' transcribed text. Beneath both text areas, the virtual representation of the previously displayed keyboard was updated and simplified. The shift, enter, and number keys were all removed, and the backspace key readjusted. The participants were then required to use the currently active keyboard interaction style to enter the displayed word using word-gesturing as before. During the word-gesturing process, participants were still shown real-time updates to the displayed word and transcribed text as well as their movements within the virtual keyboard space. The participants were required to use the active keyboard's backspace key to remove errors. However, already correctly transcribed characters were protected from being deleted. The change to protect the correctly transcribed characters was because of the high sensitivity and precision required to only delete the erroneous characters. Once a word was correctly entered, the participants were to release the simulated touch by the appropriate means of the active keyboard to move to the next word instead of hitting the enter key.

As before, deviations in the above task were required for how the participants interacted with the Xbox Controller keyboard. For this keyboard, there was still no word-gesturing feature implemented, instead the participant was asked to use single-input text entry using a standard Xbox 360 Controller. The participants had to press the 'A' button to select a letter, 'Y' button to delete, and the 'start' button to move to the next word.

4.2.4 Experimental Design

A within-subjects design was used for the pilot study (Kim 2010). The strength of a within-subjects design is the overall power increase and reduction in error variance associated with individual differences. The weakness of using the within-subjects design is that it suffers from "carryover effects" (the participation in one condition may affect performance in other conditions) between each keyboard interaction style. To account for this weakness, the study was supplemented with a Latin Squares design for counterbalancing (Zhang 2010). Table 4.4 shows how the Latin Squares design was utilized for a sample size of 7 with an equal number of different keyboard inputs.

4.2.5 Procedure

Each subject participated in a single study visit which took between 30 and 70 minutes to complete depending on how many calibrations were performed. The full set of tasks and their expected durations are detailed in Figure 4.5, the pilot study schedule of assessments. The participants followed the same process for each of the 7 keyboard interaction styles' tasks.

First, the participants were given a brief verbal explanation and physical demonstration of the active keyboard input. The explanation dialog contained the

Table 4.4. Latin Squares design for 7 participants and 7 conditions.

participants	conditions						
1	A	B	C	D	E	F	G
2	B	C	D	E	F	G	A
3	C	D	E	F	G	A	B
4	D	E	F	G	A	B	C
5	E	F	G	A	B	C	D
6	F	G	A	B	C	D	E
7	G	A	B	C	D	E	F

Table 4.5. Schedule of Assessments for a single study visit during the pilot study (in minutes).

Step	Controller	Touch Screen	Leap Motion Surface	Leap Motion Static-air	Leap Motion Pinch-air	Leap Motion Predictive-air	Leap Motion Bimodal-air	Exit Survey	Total
explain	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	3.5
calibrate	0	0	2	2	2	2	2	0	10
practice	5	5	5	5	5	5	5	0	35
task	2	2	2	2	2	2	2	0	14
survey	0	0	0	0	0	0	0	5	5
Total	7.5	7.5	9.5	9.5	9.5	9.5	9.5	5	67.5

name of the active keyboard and the method of interaction. The researcher then demonstrated how to enter the word "test" using the active keyboard. The participants were then given control of interaction with the keyboard in order to become familiar with the interaction style.

Participants were then instructed to use the keyboard to perform practice words which were randomly selected from the Oxford English Dictionary with lengths between 3 and 6 characters long. The practice words were filtered for offensive words and against the previously constructed experiment dictionaries so that no participants would be able to see any of the experiment words in advance. There was no limit placed on how many words could be performed while practicing. The participants were told to continue until they felt they were able to efficiently and comfortably gesture each word with minimal errors. During the practice phase, participants were given the opportunity to recalibrate the interaction-space if it was applicable for the currently active keyboard. Recalibration was allowed any number of times.

Next, the participants performed the task itself. As detailed in Section 4.2.3, the participants were instructed to enter a total of 10 words for the current keyboard interaction style. The words selected were pulled at random from the active keyboard's previously constructed dictionary until all 10 words in the dictionary were used. Participants were not allowed to recalibrate the interaction space during the task.

After all tasks were completed for each of the 7 interaction styles, the participants were asked to fill out an exit survey, shown in Appendix B. The exit survey asked the participants for their age, gender, major, and handedness as well as several questions detailing any prior touch, gesture-controller, or word-gesturing experience or impairments that might relate to the study. Finally, the participants were required to fill out the a Likert scale section relating to difficulty, discomfort and fatigue experienced when using the interaction styles as well as rank each style on a numerical scale from best to worst.

4.2.6 Dependent Measures

The design choice to not fully implement word-recognition for the word-gesture keyboards influenced the design of the task and therefore affected the dependent measures. Each individual trial was designed as a single word rather than a phrase that included many words, as in Vulture (Markussen, Jakobsen, and Hornbæk 2014). Therefore, most dependent measures were analyzed on the word-gesture level. It was possible to analyze these values at the phrase level if the combined trials were viewed as a single phrase.

4.2.6.1 Text-entry rate. Typically, text-entry rates were calculated using the standard Words Per Minute formula

$$WPM = \frac{|T-1|}{S} \times 60 \times \frac{1}{5} \tag{4.1}$$

where $|T-1|$ was the length of the transcribed string and S was the amount of time (in seconds) that was taken to transcribe the word from the time the first character was produced (Bi, Chelba, Ouyang, Partridge, and Zhai 2012). When dealing with timing word-gestures, the formula must be modified to:

$$WPM = \frac{|T|}{S} \times 60 \times \frac{1}{5} \tag{4.2}$$

where $|T-1|$ was replaced with $|T|$ and S represents the amount of time (in seconds) that was taken to transcribe the word including the time that it took to produce the first character. The modification was required because the time it takes to produce the first character must be included when timing word-gestures (MacKenzie and Tanaka-Ishii 2007).

4.2.6.2 Error rates. There were several techniques used to measure error rates and find the best representation of keyboard performance and account for the task design.

The first error rate was modeled after the techniques in Vulture (Markussen, Jakobsen, and Hornbæk 2014) and uses the Minimum Word Distance, which was calculated in the same way as the Minimum String Distance (Soukoreff and MacKenzie 2003, MacKenzie and Soukoreff 2002):

$$MWD\ error\ rate = \frac{MWD(P,T)}{\overline{S_P}} \times 100\% \tag{4.3}$$

Minimum Word Distance differentiated from Minimum String Distance in that it was calculated on a per-word level than on a per-character level where P and T were the sets of words in the presented and transcribed strings, and $\overline{S_P}$ was the mean size of the optimal alignments calculated on a per-word level (Markussen, Jakobsen, and Hornbæk 2014). It is important to note that because participants were forced to correctly type in words and there were no errors present in the final transcribed text, words that were considered erroneous had to be defined differently. For this thesis to take advantage of Minimum Word Distance, the word was counted as incorrect if the participant made any errors at all regardless of being forced to correct them. This gives the formula

$$MWD\ error\ rate = \frac{IW}{CW + IW} \times 100\% \tag{4.4}$$

where IW were words where the participant made any mistakes at all regardless of corrections and CW were words where the participant got the word correct on the first attempt.

The next error rate used was the Keystrokes Per Character method, otherwise known as KSPC (Soukoreff and MacKenzie 2003). The Keystrokes Per Character formula

$$KSPC \approx \frac{C + INF + IF + F}{C + INF} \tag{4.5}$$

used Soukoreff and MacKenzie's keystroke taxonomy, where C represented the correct characters in the transcribed text, INF were the incorrect, not fixed characters in the transcribed text, F was used to show the keystrokes which were editing functions (e.g., backspace), and IF were the incorrect but fixed characters in the input stream. In consideration of IF, this did not include selecting backspace on accident. The Keystrokes Per Character method is less than ideal and has many limitations as an error metric (Soukoreff and MacKenzie 2003). Yet, it was still beneficial to be used to analyze the rate of erroneous character production because the word-gesture keyboards were designed with a lack of true word-recognition. It is important to note that INF always equated to zero, reducing Formula 4.5 to

$$KSPC \approx \frac{C + IF + F}{C} \tag{4.6}$$

because the task required participants correctly transcribe each word before moving on to the next.

The final error rate used was the Total Error Rate (Soukoreff and MacKenzie 2003). The Total Error Rate was also described by the previous keystroke taxonomy used in KSPC, giving the formula

$$Total\ Error\ Rate = \frac{INF + IF}{C + INF + F} \times 100\% \tag{4.7}$$

where C represented the correct characters in the transcribed text, INF were the incorrect, not fixed characters in the transcribed text, F was used to show the keystrokes which were editing functions (e.g., backspace), and IF were the incorrect but fixed characters in the input stream. Once again, in consideration of IF, this did not include selecting backspace on accident. Again, Formula 4.7 could be reduced to

$$Total\ Error\ Rate = \frac{IF}{C+F} \times 100\% \tag{4.8}$$

because participants were required to correctly transcribe words.

4.2.6.3 Correctness. Correctness of a single word-gesture was determined by calculating the Fréchet Distance between the expected word-gesture path and the participant's generated word-gesture path. The Fréchet Distance between two curves P and Q, or in this case gesture-shapes, was defined as the minimum length leash needed to walk a dog when the person walks along P and the dog walks along Q (Har-Peled and Raichel 2014). Figure 4.3 shows the recursive implementation of Fréchet Distance used in this thesis where P and Q were the two paths being walked, CA was the matrix that contains all possible distance values for each comparison, and i and j were the indices that were being examined for that particular recursive phase.

4.2.6.4 Distance measures. Distance measures were used to evaluate the movements of the participants' hands in the interaction plane. The two primary distance measures were the distance traveled to complete a word's gesture-shape (recorded in centimeters) and the average velocity of the participant's hand (recorded in centimeters per second).

4.2.6.5 Time-based measures. Time-based measures were used to calculate text-entry rates as well as attempt to evaluate the participant's level of focus. The primary time-based measure recorded was the duration required to complete a word's

```java
/** Recursive implementation of Frechet Distance algorithm.
  * @params P, Q the paths; CA distance matrix; i,j indicies
  * @return calculated Frechet Distance for CA[i][j]
  */
private float frechetRecursive(Vector [] P, Vector [] Q, Float [][] CA,
        int i, int j) {
    float CAij = 0;
    if(CA[i][j] > -1) {
        CAij = CA[i][j];
    } else if(i == 0 && j == 0) {
        CA[i][j] = distance(P[0], Q[0]);
        CAij = CA[i][j];
    } else if(i > 0 && j == 0) {
        CA[i][j] = Math.max(frechetRecursive(P, Q, CA, i - 1, 0),
            distance(P[i], Q[0]));
        CAij = CA[i][j];
    } else if(i == 0 && j > 0) {
        CA[i][j] = Math.max(frechetRecursive(P, Q, CA, 0, j - 1),
            distance(P[0], Q[j]));
        CAij = CA[i][j];
    } else if(i > 0 && j > 0) {
        float min = Math.min(frechetRecursive(P, Q, CA, i - 1, j),
            frechetRecursive(P, Q, CA, i - 1, j - 1));
        min = Math.min(min, frechetRecursive(P, Q, CA, i, j - 1));
        CA[i][j] = Math.max(min, distance(P[i], Q[j]));
        CAij = CA[i][j];
    } else {
        CA[i][j] = Float.POSITIVE_INFINITY;
    }
    return CAij;
}
```

Figure 4.3. A snippet of code showing the recursive implementation of Fréchet distance.

gesture-shape in seconds. To determine a participant's level of focus on the task, the response time to errors (in seconds) was recorded during the experiment. Finally, the duration for participants to first simulate a touch and to correctly enter the first letter were recorded (both in seconds).

4.2.6.6 Additional quantitative measures. There were two additional quantitative measures recorded, the number of practice words completed for each input per participant and the number of times a touch was simulated for each subject per input. To gauge the ease of learning for each keyboard, the number of practice words each participant completed was recorded. The number of times a touch was simulated was anticipated to be related to error rates. The number of touches simulated helped to determine if there were detection errors with the device itself rather than errors generated by the participants.

4.2.6.7 Qualitative measures. The qualitative measures in the pilot study were recorded by utilizing an exit survey once the task for all of the keyboards had been completed. The participants were asked to rate each keyboard used in terms of discomfort, difficulty, and fatigue using a Likert scale with 5 options. Discomfort was defined as the awkwardness of the keyboard and whether it required an uncomfortable position or gesture to use. Difficulty evaluated whether the keyboard interaction style was confusing to understand how to use. Fatigue asked the participant if they had experienced any tiredness or soreness from the keyboard they had used. Lastly, participants ranked the keyboards from 1 to 7, from the most to least preferred keyboards.

4.2.7 Pilot Summary

The pilot study made evident that readjusting the Leap Motion controller's position was, at times, more effective than recalibrating the interaction space. As a result, a default interaction space was calibrated and provided for the full study. In addition, the Xbox Controller Keyboard was determined to be irrelevant to the study and removed from future studies. Section 7.2.9 gives more information on how word-gesture keyboards could be applied to gaming consoles. Since a default calibration was provided and the Xbox Controller Keyboard was removed from the study, the experiment was expected to have additional time. Therefore, to take advantage of the free time, the number of trials were slightly increased in the full study. It was observed that participants were having a hard time remembering which keyboard was which during the exit survey. For this reason, intermittent surveys were introduced so that a survey could be completed for each keyboard directly after finishing a keyboard's task. Finally, to adhere to the "Come As You Are" principle mentioned in Section 2.2 and provide a more natural interaction approach, the stylus was removed for all mid-air keyboards in favor of barehanded tracking. However, the stylus was not removed from the Leap Surface Keyboard due to the orientation of the custom Leap Motion holder, as seen in Section 3.2.2, because the angle of the hand introduced too many detection errors.

CHAPTER FIVE

Methodology

The full study builds on observations in the pilot studies that were discussed in Chapter 4. It explored similar objectives with key differences. As detailed in section 4.2.7, the Xbox Controller Keyboard was removed due to its irrelevance; see Section 7.2.9 for more information on how gaming consoles' virtual keyboards could be improved by moving away from single-input text entry and instead implementing a word-gesture keyboard. In addition, the use of a stylus was removed as an interaction tool from all mid-air keyboards, allowing participants to interact barehanded with the mid-air keyboard inputs. The absence of the stylus aimed to remove the barrier between the participant and the keyboard input during the task. The goal was to make word-gesturing feel more natural (comparable to a touch screen) and adhere to the "Come As You Are" design principle. Section 2.2 gives more insight on the usefulness of this approach. The stylus, however, was not removed from the Leap Surface Keyboard because of the issues associated with tracking that arise from the Leap Motion controller placement when using the custom holder, as described in Section 3.2.2. Finally, the full study of this thesis adhered to a stringent protocol that was approved by the Institutional Review Board (IRB).

5.1 Participants

A sample size of 18 was used in the full study. The justification for this sample size comes from the formula to calculate the sample size for two independent group means using a pooled standard deviation (Dattalo 2008):

$$N = \frac{2(z_{\frac{\alpha}{2}} + z_{1-\beta})^2}{(\frac{\mu_1 - \mu_2}{\sigma_{pooled}})^2} \tag{5.1}$$

where $z_{\frac{\alpha}{2}}$ and $z_{1-\beta}$ were the z-scores for the α and β, respectively, μ_1 and μ_2 were the means of the two populations being compared and σ_{pooled} was the pooled standard deviation of the two populations. A power of $1 - \beta = 0.80$ and a significance level of $\alpha = 0.05$ were used when calculating the sample size. The derived sample size was the average sample size for all relevant variable comparisons based on the study objectives. Outliers requiring a sample size greater than 100 were removed. Furthermore, a sample size of 18 justifies a Replicated Latin Squares design for 6 input methods. The Latin Squares design was chosen for counterbalancing the experimental design and to reduce the effect of participation in one condition affecting performance of other conditions. Further details are explained in Section 5.4.

There were 13 male and 5 female participants with ages ranging from 18 to 24 with a median age of 21. Participants' computer usage ranged from 1 to greater than 50 hours per week with a median usage between 31 to 40 hours per week and 41 to 50 hours per week. All but two of the participants described their right hand as being dominant with one participant describing their left hand and the other claiming to be ambidextrous. Correspondingly, all participants used their right hand during the experiment except for one participant who used their left hand and another who switched back and forth. All of the participants had previous experience with touch devices; 83% of participants had previous experience with gesture-controllers and only 56% had previous experience with word-gesture keyboards. No participants had any impairment that affected their ability to enter text with computers. All participants were required to read and sign an IRB approved consent form before participating in the experiment. Table 5.1 contains more specific details on each participant.

5.2 Input Devices and Interaction Styles

In order to focus the study on only word-gesture keyboards, the full study saw the removal of the Xbox Controller Keyboard. All of the keyboards except for the Touch Screen Keyboard were simulated on the same 64-bit Windows 7 work

Table 5.1. Participant information including age, gender, handedness, computer usage, and previous experiences.

Subject	Gender	Age	Computer Usage per Week (hours)	Handedness	Hand Used in Experiment	Touch-device Experience	Gesture-controller Experience	Word-gesturing Experience	Impairment History
1	female	20	41 - 50	right	right	yes	yes	no	no
2	male	24	31 - 40	right	right	yes	yes	yes	no
3	male	19	1 - 5	both	both	yes	yes	no	no
4	male	23	41 - 50	right	right	yes	yes	yes	no
5	male	21	21 - 30	right	both	yes	yes	no	no
6	female	22	50+	right	right	yes	yes	yes	no
7	male	18	41 - 50	right	right	yes	yes	yes	no
8	female	21	11 - 20	right	right	yes	no	yes	no
9	male	22	50+	left	left	yes	yes	yes	no
10	male	18	21 - 30	right	right	yes	yes	yes	no
11	female	21	50+	right	right	yes	no	no	no
12	male	22	11 - 20	right	right	yes	yes	yes	no
13	male	22	50+	right	right	yes	yes	no	no
14	female	18	50+	right	right	yes	yes	yes	no
15	male	23	50+	right	right	yes	yes	no	no
16	male	18	6 - 10	right	right	yes	no	yes	no
17	male	19	11 - 20	right	right	yes	yes	no	no
18	male	18	31 - 40	right	right	yes	yes	no	no

station. The Touch Screen Keyboard was simulated on the C4667PW, a 3M™ Multi-touch Display, running 64-bit Windows 8. All receivers or controllers were connected through USB 2.0. Up until the start of the task, participants were allowed to recalibrate the active keyboard's interaction plane until they were satisfied by the calibration settings. However, a default interaction space was provided that was unlikely to need recalibration. Participants were encouraged to use the default-calibrated interaction space and reposition the gesture-controller. They were to use the keyboards in whatever way they felt they could perform best.

To adhere to the "Come As You Are" design principle, the stylus was removed from all mid-air keyboards (excluding the Leap Surface Keyboard). This absence aimed to remove the barrier between the participant and the keyboard input during the task. The goal was to make using these keyboards feel more natural by emulating an experience closer to touch-based interactions in mid-air. Section 2.2 gives more insight on the usefulness of this approach.

5.2.1 Touch Screen Keyboard

The Touch Screen Keyboard was used on a large tabletop touch screen. Participants used their finger to interact with the virtual keyboard on the screen in the same way as typical touch devices. Touch was detected when a participant's finger was pressed against the screen, and release was simulated when the finger was lifted from the surface.

5.2.2 *Leap Motion Surface Keyboard*

The Leap Motion Surface Keyboard used the Leap Motion controller for tracking the user's input and movement. Unlike the other Leap-based keyboards, the gesture controller was placed into a custom-designed holder instead of on the desk in front of the participant. The holder was attached to an inclined surface with a printed keyboard fixed on top. As opposed to the pilot study, placement was ensured to be similar between uses, and therefore the Leap Motion Surface Keyboard required only a single calibration for all participants. The participant then used a stylus tracked by the Leap Motion controller in order to detect interaction. A touch was simulated by pressing the tip of the stylus against the printed keyboard and a release was simulated when the tip of the stylus was again removed from the printed keyboard surface.

5.2.3 *Leap Motion Static-air Keyboard*

The Leap Motion Static-air Keyboard also used a Leap Motion controller, now placed on the desk in front of each participant. Participants then used their pointer finger of their dominant hand, which was tracked by the Leap Motion controller, to interact with a projected interaction plane. A touch was simulated by the insertion of their finger into the interaction plane and a release was simulated upon the removal of their finger.

5.2.4 *Leap Motion Pinch-air Keyboard*

Similar to the Static-air Keyboard, the Leap Motion Pinch-air Keyboard also used a Leap Motion controller positioned in front of the participant. The participants used their bare hands, tracked by the center of their palm, to interact with the projected interaction plane. Touch was simulated by having the participant make a pinching gesture with their hand and a release was simulated when the participant released the pinch, opening their hand again.

5.2.5 Leap Motion Bimodal-air Keyboard

Again, the Leap Motion Bimodal-air Keyboard used a Leap Motion controller positioned in front of the participant. The participants used the pointer finger of their dominant hand, tracked by the Leap Motion controller, to determine the location over the projected virtual keyboard. A touch was simulated by pressing the space bar key on a standard QWERTY keyboard and a touch release was simulated upon the release of the space bar.

5.2.6 Leap Motion Predictive-air Keyboard

As the others, the Leap Motion Predictive-air Keyboard saw a Leap Motion controller placed before the participant. Participants then used the pointer finger of their dominant hand, again tracked by the Leap Motion controller, to interact with a projected interaction plane. A touch was simulated by recognizing and predicting a forward gesture of the participant's finger toward the interaction plane and a release was simulated by recognizing a backward gesture away from the interaction plane.

5.3 Task Design

As in the pilot, the conditions of the task were represented by 6 different keyboard interaction styles. The 6 conditions used were the Leap Motion Static-air, Leap Motion Pinch-air, Leap Motion Surface, Leap Motion Predictive-air, Leap Motion Bimodal-air, and Touch Screen keyboards.

Task profiles were created for each of the 6 interaction styles, where each task profile consisted of 15 separate trials for a total of 90 trials per participant. The increase in trials from 10 to 15 words for each keyboard was due to the time added from the removal of the Xbox Controller Keyboard condition. The creation of the task profiles was handled by generating static, unchanging dictionaries for each keyboard, guaranteeing a total of 90 unique words across all participants. The 15 words selected for each dictionary were generated by the same custom dissimilarity algorithm, producing the top 15 least dissimilar gesture-shapes across all words in the Oxford English Dictionary for words 3 to 6 characters in length.

For each trial, a word was chosen at random from the active keyboard's previously constructed dictionary and displayed on the screen. A blank text-area was positioned directly below the displayed word for the participants' transcribed text. Beneath both text areas, the keyboard interaction styles were virtually represented. The participants were then required to use the currently active keyboard interaction style to enter the displayed word using word-gesturing. During the word-gesturing process, participants were shown real-time updates. The participants were required to use the active keyboard interaction's backspace key to remove errors. However, already correct transcribed characters were protected from being deleted. Once a word was correctly entered, the participants released the simulated touch to move to the next word.

5.4 Experimental Design

A within-subjects design was used for the final study (Kim 2010). The strength of a within-subjects design is the overall power increase and reduction in error variance associated with individual differences. The weakness of using the within-subjects design is that it suffers from "carryover effects" (the participation in one condition may affect performance in other conditions) between each keyboard interaction style. To minimize carryover effects, the study was supplemented with a Replicated Latin Squares design for counterbalancing (Mansson and Prescott 2001). Table 5.2 shows how the Replicated Latin Squares design was utilized for 6 different keyboard inputs with a sample size of 18.

Table 5.2. The three replications requires for a Replicated Latin Squares design for 18 participants and 6 conditions.

First Replication

participants	conditions					
1	A	B	C	D	E	F
2	B	C	D	E	F	A
3	C	D	E	F	A	B
4	D	E	F	A	B	C
5	E	F	A	B	C	D
6	F	A	B	C	D	E

Second Replication

participants	conditions					
7	F	A	B	C	D	E
8	A	B	C	D	E	F
9	B	C	D	E	F	A
10	C	D	E	F	A	B
11	D	E	F	A	B	C
12	E	F	A	B	C	D

Third Replication

participants	conditions					
13	E	F	A	B	C	D
14	F	A	B	C	D	E
15	A	B	C	D	E	F
16	B	C	D	E	F	A
17	C	D	E	F	A	B
18	D	E	F	A	B	C

5.5 *Procedure*

Each subject participated in a single study visit that took between 30 and 45 minutes to complete after having read and signed an IRB approved consent form. For the full study, almost no calibrations were performed. The full set of tasks and their expected durations are detailed in the full study schedule of assessments (Table 5.3). The participants followed the same process for each of the 6 keyboard interaction styles' tasks.

First, the participants were given a brief verbal explanation and physical demonstration of the active keyboard input. The explanation dialog contained the

Table 5.3. Schedule of Assessments for a single study visit during the full study (in minutes).

Step	Touch Screen	Leap Motion Surface	Leap Motion Static-air	Leap Motion Pinch-air	Leap Motion Predictive-air	Leap Motion Bimodal-air	Exit Survey	Total
explain	0.5	0.5	0.5	0.5	0.5	0.5	0	3
calibrate	0	0	.5	0	.5	.5	0	1.5
practice	3	3	3	3	3	3	0	18
task	1.5	1.5	1.5	1.5	1.5	1.5	0	9
survey	0.5	0.5	0.5	0.5	0.5	0.5	3	6
Total	5.5	5.5	6	5.5	6	6	3	37.5

name of the active keyboard and the method of interaction. The researcher then demonstrated how to enter the word "test" using the active keyboard. The participants were then given control of interaction with the keyboard in order to become familiar with the interaction style.

Participants were then instructed to use the keyboard to perform the input of practice words. These words were chosen at random from the Oxford English Dictionary with lengths between 3 and 6 characters long. The practice words were filtered for offensive words and against the previously constructed experiment dictionaries. There was no limit placed on how many words could be performed while practicing. The participants were told to continue until they felt they were able to efficiently and comfortably type each word with minimal errors. They were also told the keyboards could be re-calibrated if necessary but were encouraged to learn to use the default calibration. This change was brought about due to participants having a hard time finding calibrations that worked during the pilot study, sometimes calibrating for upwards of 30 minutes and still producing poor results.

Next, the participants performed the task itself. As detailed in Section 5.3, the participants were instructed to enter a total of 15 words for the current keyboard interaction style. The words selected were randomized from the active keyboard's previously constructed dictionary until all 15 words in the dictionary were used.

After the task for the active keyboard was completed, the participants were asked to fill out a small survey section, seen in Figure B.1. The survey asked participants to use the Likert scale to rate each keyboard in terms of difficulty, discomfort and fatigue experienced when using the interaction styles.

Finally, after all tasks were completed for each of the 6 interaction styles, the participants were asked to fill out an exit survey shown in Figure B.2. The exit survey asked the participants for their age, gender, major, and handedness as well as several questions detailing any prior touch, gesture-controller, or word-gesturing experience or impairments that might relate to the study. Lastly, the participants were asked to rank each interaction style on a numerical scale from "most preferred" to "least preferred".

5.6 Dependent Measures

Given that each individual trial was designed as a single word rather than a phrase that included many words, most dependent measures were analyzed on the word-gesture level. If the combined trials were viewed as a single phrase, it would be possible to analyze these values at the phrase level. This combination was used to analyze the Minimum Word Distance from Markussen et al. (2014). Due to the lack of word-recognition in the design of the word-gesture keyboard, the forcing of participants to make corrections to transcribed words, and to help accommodate for device detection errors that were out of the participants' control, some of the dependent measures were modified. These modifications directly affected the data processing for the transcribed word.

The initial modification involved using the shortest form of the transcribed word. Table 5.4 shows how this modification was applied. The shortest-transcribed modification helps to account for device detection errors as well as forced word correction. However, an observed drawback can be seen in Example 4 of Table 5.4. The

Table 5.4. Examples of the shortest-transcribed modification.

Example	Presented text:	Input Stream:	Transcribed text:
1	quick	wquiclk←←←←←←←quick	wquiclk
2	dot	fdot←←←←di←ot	fdot
3	burn	burnm←	burn
4	fire	fuired←←←←←ired←	fuire

intention seemed to be the word “fired” instead of “fire,” but this information was lost with the shortest-transcribed modification.

Since participants were required to correct errors, the second modification included backspace entry from the input stream as part of the presented word when a participant made mistakes. With the modification to the presented text, the transcribed string was now represented by the entire input stream. Table 5.5 shows how this modification was applied. The main motivation behind this modification was to mirror the participants' requirement to correctly enter words, especially for the Fréchet Distance in Section 5.6.3.

5.6.1 Text-entry Rate

Text-entry rates were calculated using modified Words Per Minute, Formula 4.2, where $|T - 1|$ was replaced with $|T|$ and S represents the amount of time (in seconds) that was taken to transcribe the word, including the time taken to produce the first character. The text-entry rate was calculated with and without the shortest-transcribed modification.

Table 5.5. Examples of the backspace-transcribed modification.

Example	Presented text:	Input Stream:	Modified Presented text:	Transcribed text:
1	quick	wquiclk←←←←←←←quick	←←←←←←←quick	wquiclk←←←←←←←quick
2	dot	fdot←←←←di←ot	←←←←d←ot	fdot←←←←di←ot
3	burn	burnm←	burn←	burnm←
4	fire	fuired←←←←←ired←	f←←←←←ire←	fuired←←←←←ired←

5.6.2 Error Rates

There were several techniques used to measure error rates to find the best representation of keyboard performance and account for task design. The first error rate used was the modified Minimum Word Distance, Formula 4.4, where IW was words where the participant made any mistakes regardless of corrections, and CW was words where the participant entered the word correct on the first attempt. The shortest-transcribed modification, though, allows for the original Minimum Word Distance formula, Formula 4.3 from Vulture (Markussen, Jakobsen, and Hornbæk 2014), where P and T were the sets of words in the presented and transcribed strings, and $\overline{S_P}$ was the mean size of the optimal alignments calculated on a per-word level.

Next, due to the addition of the shortest-transcribed modification and the lack of word-recognition implementation of the word-gesture keyboards, the Minimum String Distance error rate was able to be analyzed (Soukoreff and MacKenzie 2003). Using the simplified keystroke taxonomy that was presented before, Minimum String Distance could be defined as the formula

$$MSD\ error\ rate = \frac{INF}{C + INF} \times 100\% \tag{5.2}$$

where C represented the correct characters in the transcribed text and INF was the incorrect, not fixed characters in the transcribed text.

The simplified Keystrokes Per Character formula, Formula 4.6, was used where C represented the correct characters in the transcribed text, INF was the incorrect, not fixed characters in the transcribed text, F was used to show the keystrokes which were editing functions (e.g., backspace), and IF was the incorrect but fixed characters in the input stream. In consideration of IF, this did not include selecting backspace on accident. Additionally, Formula 4.5 was used with the shortest-transcribed modification.

The final error rate was the simplified Total Error Rate from Formula 4.8 where C represented the correct characters in the transcribed text, INF was the incorrect, not fixed characters in the transcribed text, F was used to show the keystrokes which were editing functions (e.g., backspace), and IF was the incorrect but fixed characters in the input stream. Once again, in consideration of IF, this did not include selecting backspace on accident. With the edition of the shortest-transcribed modification, Formula 4.7 was also utilized.

5.6.3 Correctness

As in the pilot study, correctness of a single word-gesture was determined by calculating the Fréchet Distance between the expected word-gesture path and the participant's generated word-gesture path. Figure 4.3 shows the recursive implementation of Fréchet Distance used in this thesis where P and Q were the two paths being walked. CA was the matrix that contains all possible distance values for each comparison, and i and j were the indices that were being examined for that particular recursive phase. The Fréchet Distance was also calculated using both the shortest-transcribed and backspace-transcribed modifications.

5.6.4 Distance Measures

The two primary distance measures were the distance traveled to complete a word's gesture-shape (recorded in centimeters) and the average velocity of the participant's hand (recorded in centimeters per second).

5.6.5 Time-based Measures

The primary time-based measure taken was the duration required to complete a word's gesture-shape in seconds. Additionally, the participant's reaction time to errors, the duration it took for participants to first simulate a touch, and the time it took to correctly enter the first letter (all recorded in seconds).

5.6.6 Additional Quantitative Measures

There were two additional quantitative measures recorded; the number of practice words completed for each input per participant, and the number of times a touch was simulated for each subject per input.

5.6.7 Qualitative Measures

The qualitative measures in the full study were gathered from the intermittent surveys and the final exit survey. In the intermittent surveys, the participants were asked to rate each keyboard that they used in terms of discomfort, difficulty, and fatigue using a Likert scale with 5 options. In the final exit survey, participants ranked the keyboards from 1, most preferred, to 6, the least preferred. See Figure B.2 to view the exit survey.

5.7 Challenges

5.7.1 Word Separation

When it comes to mid-air text-entry, one of the greatest challenges was finding a suitable means of distinguishing between separate words while minimizing complexity. Prior attempts saw selection-based techniques mostly with single-input text-entry (Markussen, Jakobsen, and Hornbæk 2013, Shoemaker, Findlater, Dawson, and Booth 2009), whereas more recent approaches explored gesture-based techniques (Ni, Bowman, and North 2011, Castellucci and MacKenzie 2008, Kristensson, Nicholson, and Quigley 2012) using defined input areas or handwriting techniques (Vikram, Li, and Russell 2013, Amma, Georgi, and Schultz 2012, Schick, Morlock, Amma, Schultz, and Stiefelhagen 2012, Shengli, Zhuxin, Li, and Chung 2008). The limitation to using selection-based techniques and using hand-gestures to interpret letters for text-entry is that these techniques were slow, suffering from low text-entry rates (Markussen, Jakobsen, and Hornbæk 2013, Shoemaker, Findlater, Dawson, and Booth 2009). The most advantageous addition to gesture-based text-entry has been the advent of shape writing (Kristensson 2007, Zhai and Kristensson 2012, Zhai, Kristensson, Gong, Greiner, Peng, Liu, and Dunnigan 2009, Zhai and Kristensson 2004, Zhai and Kristensson 2003), now known as word-gesturing, which was applied for the first time to mid-air by Markussen et al. (2014).

Markussen et al. (2014) used pinching as a means of separating between words by implementing a glove with reflective markers and a large projected display. The pinching gesture was also confined to one specific hand-gesture. Though an effective first look at mid-air word-gesture keyboards, the pinching gesture used in Vulture lacks adaptability between users, adding complexity and limiting word separation. In addition, using a glove with reflective markers, obtaining the required tracking equipment, and using a large projector display are all very inconvenient for the typical user and not easily accessible. The experience needed to be confined down to something that could be used casually with a desktop or laptop computer, while being customizable enough to be used by those who do not have the ability to form pinching gestures.

5.7.2 Motor Space vs Display Space

The decoupling of the motor space and display space contributed to the complexity of mid-air text-entry (Markussen, Jakobsen, and Hornbæk 2014). According to Markussen et al. (2014), having to mentally re-couple these spaces is difficult because of the principle of stimulus-response compatibility (Proctor and Vu 2006). Vulture tried to reduce this problem by using a motor space that was parallel to the display space, yet still only reached 59% the text-entry rate of touch-based inputs.

5.7.3 Fatigue

The last thing to consider when working with mid-air was how fatiguing these gestures could be to produce over long periods of time. In Vulture, the participant was required to stand in front of the display, holding their arms out at hip level to interact (Markussen, Jakobsen, and Hornbæk 2014). Even though this position has been shown to minimize the effects of fatigue while standing (Hincapié-Ramos, Guo, Moghadasian, and Irani 2014), this thesis intended to bring mid-air, word-gesture keyboards away from distant large screen displays and into the personal space of the user. Subsequently, this interaction generally involves sitting. Using an extended arm while sitting and performing gestures is known to cause fatigue, commonly referred to as the "Gorilla Arm Syndrome" (Yoo, Lee, and Ahn 2012, Teixeira 2011), and should be minimized for seated mid-air text-entry.

5.8 Solutions

5.8.1 Word Separation

The Leap Motion controller was used to address the issues presented by the complexity of word separation. This allowed for bare-handed, mid-air gestural interactions with sub-millimeter precision (Weichert, Bachmann, Rudak, and Fisseler 2013, Guna, Jakus, Pogačnik, Tomažič, and Sodnik 2014). It also could be easily obtained by ordinary users and runs on typical desktop computers.

The Leap Motion controller allowed the exploration of various means of word separation in the 3rd-dimension due to the extra degrees of freedom. However, this was less preferred than pinching in Vulture (Markussen, Jakobsen, and Hornbæk 2014). There was no empirical evidence provided to explain this preference, prompting this thesis to explore the 3rd-dimension to discover its full range of problems and to seek possible solutions. The 3rd-dimension was utilized by implementing various approaches that also tracked the z-direction of hand-movement.

A final alternative was to use a bimodal approach for mid-air interaction using a secondary input that replaces the pinching gesture, removing the requirement for gesturing or using extra degrees of freedom in the 3rd-dimension. The aim was to vastly reduce the complexity of word separation in mid-air.

5.8.2 Motor Space vs Display Space

To address the issue of decoupling, the 3rd-dimension was utilized by implementing a default approach with a static plane in mid-air. This was expected to heavily show the effects of decoupling. To study the real limitations and issues that decoupling has, the static interaction plane was projected onto a surface with a printed keyboard beneath it. Additionally, to minimize decoupling between the motor space and display space, the implementation discussed in Section 3.2.4 predicted when users would attempt to touch the mid-air interaction plane.

Additionally, as Vulture did with pinching, the bimodal approach aimed to minimize the effects of decoupling by removing the need to actively make specific hand-gestures, thus reducing the overall complexity of the interaction between the motor space and the displayed screen. In other words, the bimodal approach simplified simulating touch interaction.

5.8.3 Fatigue

This thesis introduced calibration of a custom interaction space, similar to that used in Personal Space (Hari Haran 2014), to allow users the freedom to rest their arms during text-entry and combat the effects of "Gorilla Arm Syndrome" (Guinness 2015, Yoo, Lee, and Ahn 2012, Freeman, Vennelakanti, and Madhvanath 2012, Hincapié-Ramos, Guo, Moghadasian, and Irani 2014). However, to focus on text-entry rates and word separation, reducing fatigue for mid-air gestures was not fully explored despite its benefits. As discussed in Section 7.2.6, implementing more personally defined interaction spaces for resting arms could further improve these calibrations (Hari Haran 2014, Guinness 2015).

CHAPTER SIX

Results and Analysis

The statistical methods used for this thesis were one-way ANOVAs for each set of dependent measures (Wu and Hamada 2009, Neter, Wasserman, and Kutner 1996), followed by a post-hoc analysis using Tukey's Honest Significant Difference (HSD) for multiple-compare (Hochberg and Tamhane 2009). The ranking system from the exit survey used the Friedman's test in conjunction with Tukey's HSD for a post-hoc analysis (Hollander, Wolfe, and Chicken 2013).

In addition, an n-way ANOVA was used on each set of variables for each keyboard with the participants' previous word-gesture experience as a factor. Surprisingly, there was no significant difference in mixed effects when using word-gesture experience as a factor, meaning participants who didn't have word-gesture experience performed as well as participants who did. This implies that interacting with word-gesture keyboards is intuitive (even applied to mid-air).

6.1 Text-entry Rate

Participants reached a mean text-entry rate of 19.5 WPM ($SD = 3.0$) for the Touch Screen Keyboard, 17.1 WPM ($SD = 3.3$) for the Leap Surface Keyboard, 8.6 WPM ($SD = 1.9$) for the Leap Static-air Keyboard, 11.3 WPM ($SD = 2.0$) for the Leap Pinch-air Keyboard, 9.6 WPM ($SD = 2.3$) for the Predictive-air Keyboard, and 15.8 WPM ($SD = 2.2$) for the Leap Bimodal-air Keyboard. Figure 6.1 shows the mean text-entry rates for each keyboard interaction style. It should be noted that the highest Bimodal-air text-entry rate, 20.7 WPM, was on par with the Touch Screen Keyboard text-entry rates. The Pinch-air keyboard, with a mean text-entry rate of 11.3 WPM, was consistent with the results from Vulture ($M = 11.8$) for participants' first session and no training (Markussen, Jakobsen, and Hornbæk 2014).

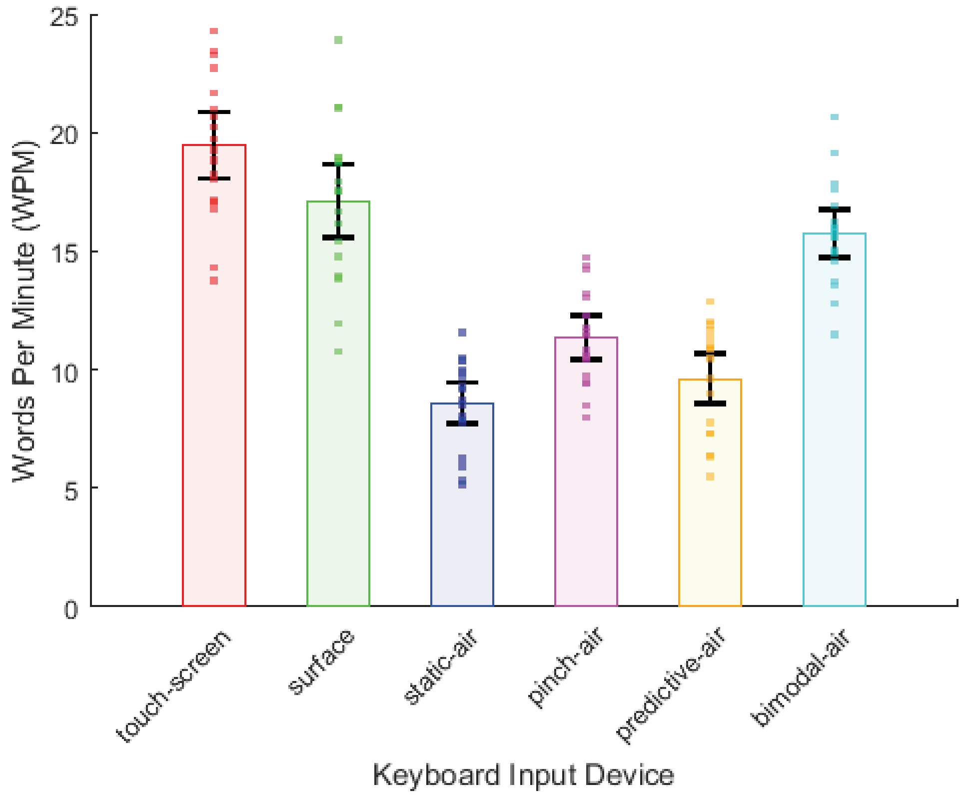


Figure 6.1. Mean Text-entry Rates for each keyboard with error bars showing 95% confidence intervals.

Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 55.5017$, p-value < 0.0001, $SD_{pooled} = 2.5$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.1, revealed significant differences (p-values < 0.001) in text-entry rates between the Touch Screen Keyboard and all mid-air keyboards. There were significant differences (p-values < 0.0001) between the Leap Surface Keyboard and the Static-air, Pinch-air, and Predictive-air keyboards. There were also significant differences (p-values < 0.0001) found between the Leap Bimodal-air and all other mid-air keyboards. Finally, there was a significant difference (p-value $= 0.0170$) found between the Pinch-air and Static-air keyboards.

Unfortunately, no mid-air keyboards were able to achieve text-entry rates as fast as the Touch Screen Keyboard and were significantly slower. However, the Bimodal-air keyboard reached a mean text-entry rate of 15.8 WPM without repeated sessions or training, indicating promising results. With repeated sessions, as in Vulture (Markussen, Jakobsen, and Hornbæk 2014), the Bimodal Keyboard is expected to increase in performance by approximately 75%. The increase is expected to be even greater (approximately 238%) when training for single short phrases, reaching speeds sufficiently high enough to compete with touch-based keyboards and surpass pinching as a means of mid-air word-gesturing.

Both the Static-air and Predictive-air keyboards underperformed, though the Predictive-air keyboard achieved marginally better results. These results were expected since adding the 3rd-dimension as a means of interaction further increased keyboard complexity and the mental coupling required between gestures in the motor-space and feedback on the display (Markussen, Jakobsen, and Hornbæk 2014, Proctor and Vu 2006). It was interesting to see how well the Leap Surface Keyboard performed since it had an identical implementation as the Leap Static-air Keyboard, only differing in where the interaction plane was projected. This implies that visually representing the Leap Static-air keyboard in 3-dimensional space using augmented reality might have the same results due to the decreased decoupling between the motor space and display space.

6.1.1 Text-entry Rate Modified-shortest

Participants reached a mean text-entry rate of 19.7 WPM ($SD = 2.9$) for the Touch Screen Keyboard, 17.3 WPM ($SD = 3.3$) for the Leap Surface Keyboard, 8.8 WPM ($SD = 1.9$) for the Leap Static-air Keyboard, 11.6 WPM ($SD = 2.1$) for the Leap Pinch-air Keyboard, 9.9 WPM ($SD = 2.3$) for the Predictive-air Keyboard, and 15.9 WPM ($SD = 2.2$) for the Leap Bimodal-air Keyboard. Figure 6.2 shows

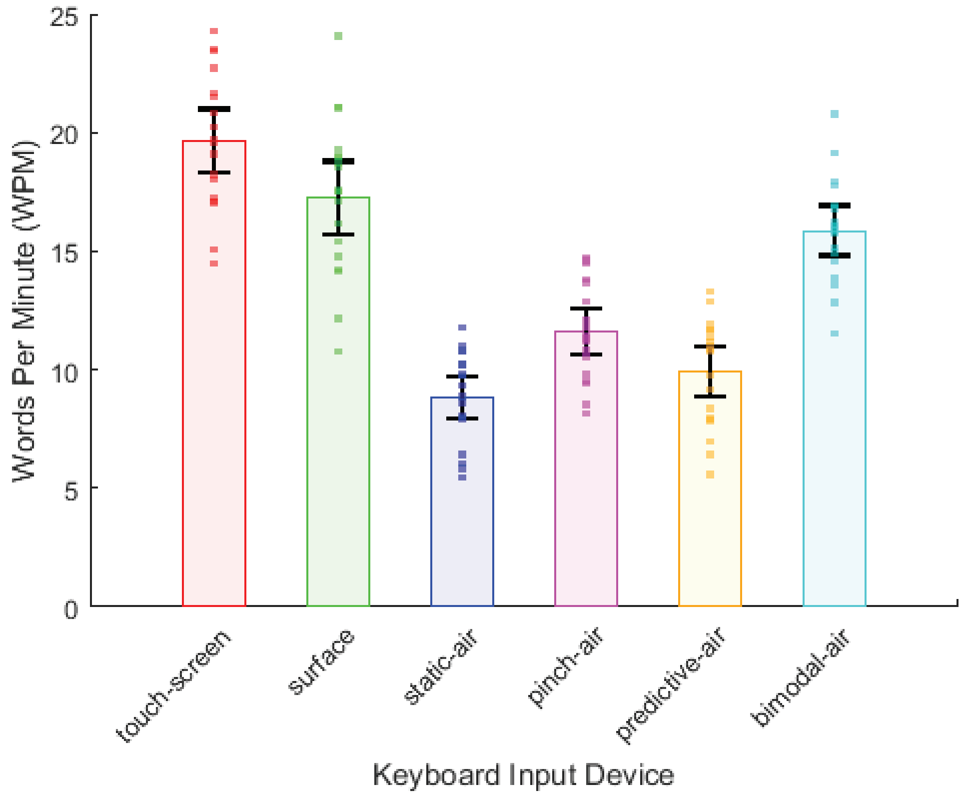


Figure 6.2. Mean Text-entry Rates using the shortest-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

the mean text-entry rates for each keyboard interaction style using the shortest-transcribed modification. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 54.6430$, p-value < 0.0001, $SD_{pooled} = 2.5$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.2, revealed significant differences (p-values < 0.001) in text-entry rates between the Touch Screen Keyboard and all mid-air keyboards. There were significant differences (p-values < 0.0001) between the Leap Surface Keyboard and the Static-air, Pinch-air, and Predictive-air keyboards. There were also significant differences (p-values < 0.0001) found between the Leap Bimodal-air and all other mid-air keyboards. The highest Bimodal-air text-entry rates were on par with the Touch Screen Keyboard. Finally, there was a significant difference (p-value $= 0.0148$) found between the Pinch-air and Static-air keyboards.

The results from using the shortest-transcribed modification were marginally better than those without, implying that errors were more likely corrected while gesturing the words rather than after having completed the gesture. The cases where errors were made while still entering all required letters seem to have had very little impact on the duration it took to complete words.

6.2 Error Rate

6.2.1 Minimum Word Distance

Participants reached a mean MWD of 13.0% ($SD = 13.4$) for the Touch Screen Keyboard, 18.9% ($SD = 14.0$) for the Leap Surface Keyboard, 56.7% ($SD = 19.4$) for the Leap Static-air Keyboard, 34.4% ($SD = 19.4$) for the Leap Pinch-air Keyboard, 48.9% ($SD = 25.2$) for the Predictive-air Keyboard, and 20.4% ($SD = 16.9$) for the Leap Bimodal-air Keyboard. Figure 6.3 shows the mean Minimum Word Distance for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 16.5496$, p-value < 0.0001, $SD_{pooled} = 18.5$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.3, revealed significant differences (p-values < 0.01) in Minimum Word Distance between the Touch Screen Keyboard and the Leap Static-air, Pinch-air, and Predictive-air keyboards. There were significant differences (p-values < 0.0001) between the Leap Surface Keyboard and the Static-air and Predictive-air keyboards. There were also significant differences (p-values < 0.001) found between the Leap Bimodal-air and the Static-air and Predictive-air keyboards. There was finally a significant difference (p-value $= 0.0062$) found between the Leap Pinch-air and Static-air keyboards.

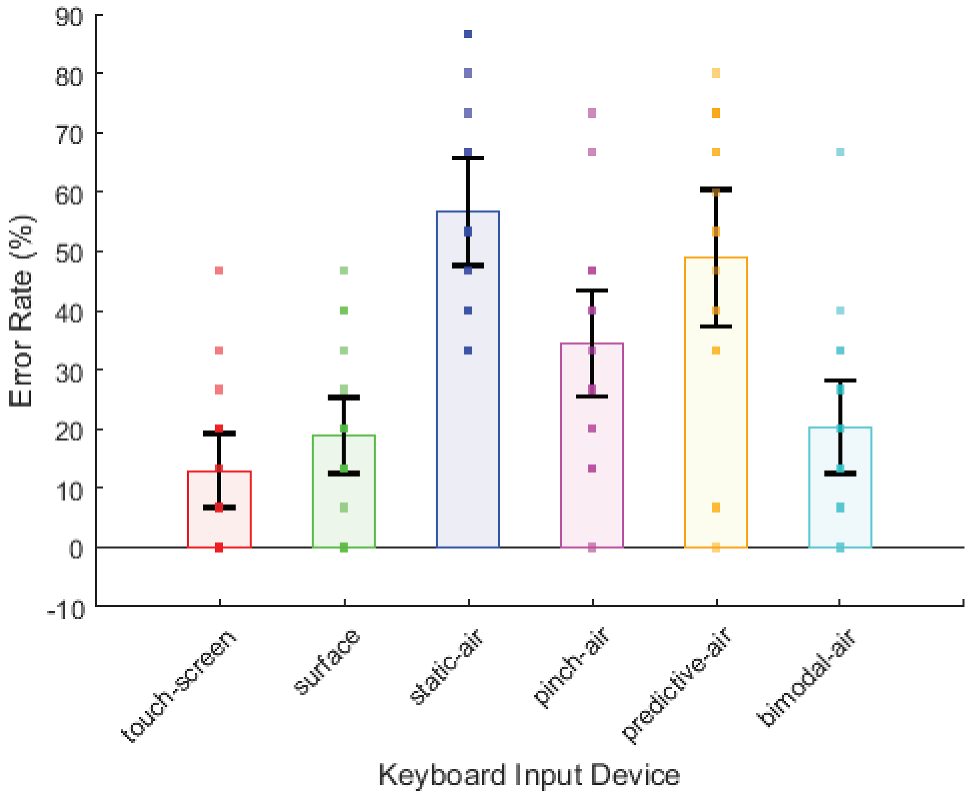


Figure 6.3. Mean Minimum Word Distance for each keyboard with error bars showing 95% confidence intervals.

The error rates here were much higher than those seen in Vulture for Pinch or Touch (Markussen, Jakobsen, and Hornbæk 2014) because this was a modified version of MWD. Since participants were required to fix errors, the MWD from Vulture would have seen error rates of 0%. Instead, if any errors were detected during the typing process, whether it be from device detection issues or the participant going for the wrong letter, the word was counted as erroneous. This representation of MWD showed how many words needed correcting for any reason.

The Static-air and Predictive-air keyboards underperformed, as expected, due to the increased decoupling and the added complexity of working with the 3rd-dimension. The Leap Bimodal-air Keyboard was significantly less erroneous than the Static-air and Predictive-air keyboards and was not significantly different from the Touch Screen Keyboard, implying that removing the 3rd-dimension significantly reduces the effects of decoupling on hand-eye coordination efforts. Again, the Leap Surface was on par with the Touch Screen keyboard, suggesting that with augmented reality the Static-air Keyboard could see a significant improvement in performance in all areas.

6.2.1.1 MWD modified-shortest. Participants reached a mean MWD of 6.3% ($SD = 9.0$) for the Touch Screen Keyboard, 6.3% ($SD = 6.3$) for the Leap Surface Keyboard, 25.6% ($SD = 20.7$) for the Leap Static-air Keyboard, 20.0% ($SD = 11.2$) for the Leap Pinch-air Keyboard, 26.0% ($SD = 16.6$) for the Predictive-air Keyboard, and 10.0% ($SD = 11.7$) for the Leap Bimodal-air Keyboard. Figure 6.4 shows the mean Minimum Word Distance with the shortest-transcribed modification for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 8.5172$, p-value < 0.0001, $SD_{pooled} = 13.5$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.4, revealed significant differences (p-values < 0.05) in Minimum Word Distance with the shortest-transcribed modification between the Touch Screen Keyboard and the Leap Static-air, Pinch-air, and Predictive-air keyboards. There were significant differences (p-values < 0.05) between the Leap Surface Keyboard and the Leap Static-air, Pinch-air, and Predictive-air keyboards. There were also significant differences (p-values < 0.01) found between the Leap Bimodal-air and the Static-air and Predictive-air keyboards.

The substantial reductions in MWD error rates when using the shortest-transcribed modification, nearly 50% reduction for all keyboards, indicated that many participants seemed to be making errors after entering all of the required letters for a gestured word. This may imply situations like Example 4 from Table 5.4, where participants were typing the wrong word, such as "fired" compared to "fire". However,

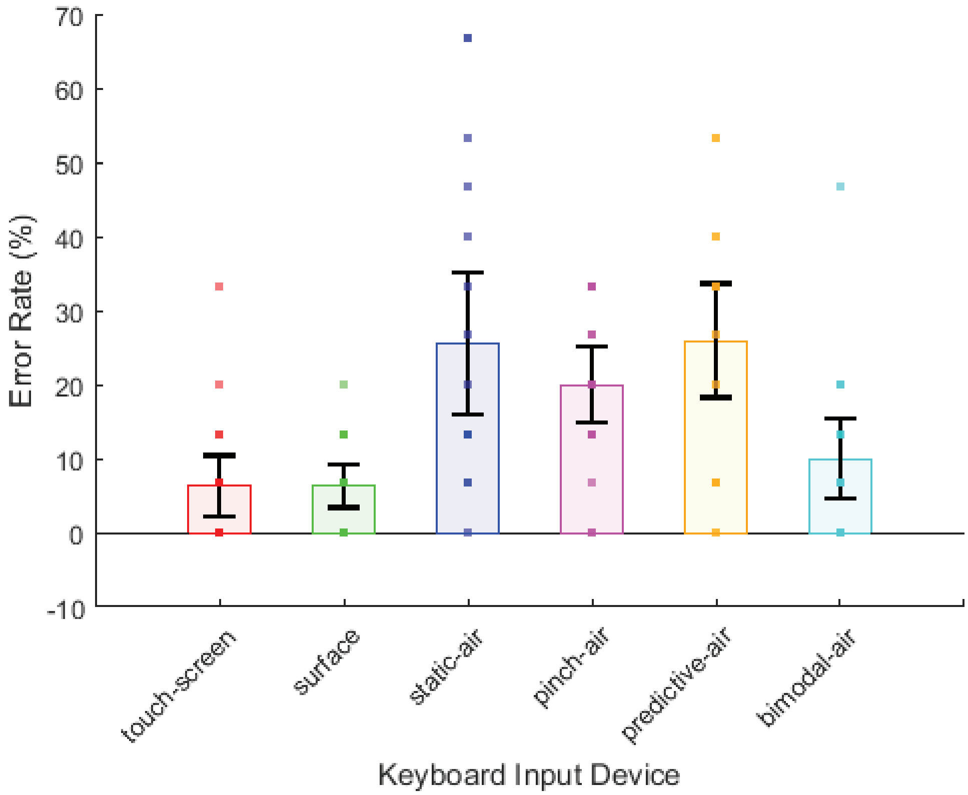
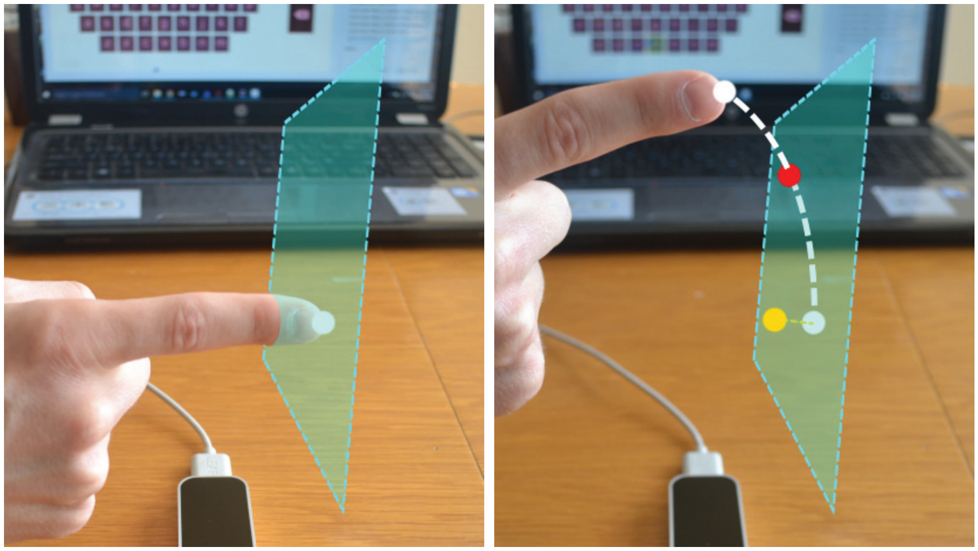


Figure 6.4. Mean Minimum Word Distance using the shortest-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

for all Leap Motion-based keyboards, the more likely implication is that many participants made errors on exiting the interaction plane. This kind of error was observed often and was thought to be heavily influenced by participants pulling their hands away from the interaction plane in an arcing motion, especially for participants who were resting their arms. Figure 6.5 details this arcing motion. It is expected that a word-recognition implementation of the word-gesture keyboards would see even lower error rates closer to those found in Vulture (Markussen, Jakobsen, and Hornbæk 2014), implying that the pseudo-implementation of the word-gesture keyboards was less robust and more sensitive to deviations in a word's gesture-shape than traditional word-gesture keyboards implemented with word-recognition.



(a) Pressing Final Key (b) Removing Hand From Plane

Figure 6.5. Examples of how the natural arcing motion generates erroneous input for 3-dimensional interactions. (a) shows the user pressing the final key; and (b) shows the intended release in *yellow* and the detected release in *red*.

6.2.2 Keystrokes Per Character

Participants reached a mean KSPC of 1.2 keystrokes per character (SD = 0.18) for the Touch Screen Keyboard, 1.4 keystrokes per character (SD = 0.38) for the Leap Surface Keyboard, 2.0 keystrokes per character (SD = 0.79) for the Leap Static-air Keyboard, 1.6 keystrokes per character (SD = 0.42) for the Leap Pinch-air Keyboard, 1.9 keystrokes per character (SD = 0.64) for the Predictive-air Keyboard, and 1.2 keystrokes per character (SD = 0.18) for the Leap Bimodal-air Keyboard. Figure 6.6 shows the mean Keystrokes Per Character for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 9.8827$, p-value < 0.0001, $SD_{pooled} = 0.49$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.5, revealed significant differences (p-values < 0.001) in Keystrokes Per Character between the Touch Screen Keyboard

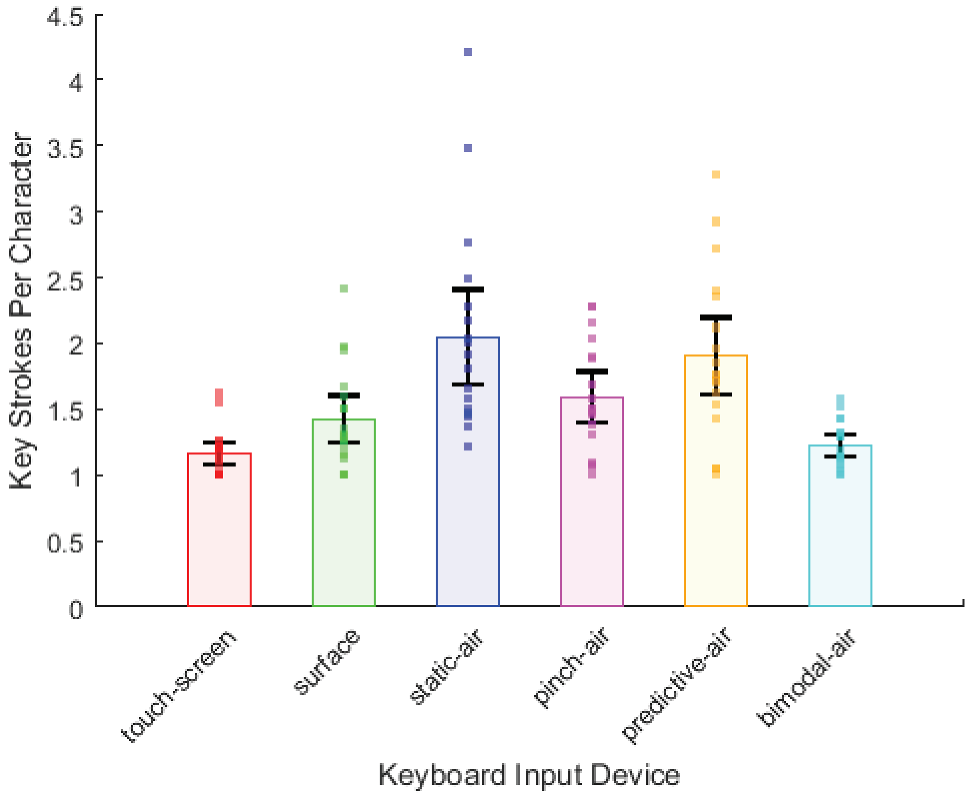


Figure 6.6. Mean Keystrokes Per Character for each keyboard with error bars showing 95% confidence intervals.

and the Leap Static-air and Predictive-air keyboards. There were significant differences (p-values < 0.05) between the Leap Surface Keyboard and the Static-air and Predictive-air keyboards. There were also significant differences (p-values < 0.01) found between the Leap Bimodal-air and the Static-air and Predictive-air keyboards.

Keystrokes Per Character was useful when analyzing the pseudo implementation of the word-gesture keyboards because words were constructed at the character level, rather than at the word level as with word-recognition. With KSPC, if words were input without any detected errors, then a KSPC of 1.0 keystrokes would be expected. This metric helps show the rate at which extra erroneous characters were produced for each keyboard. For example, the KSPC of the Static-air keyboard was

2.0 keystrokes, meaning 100% more interactions were produced than required to gesture the words successfully. As before, the Static-air and Predictive-air keyboard performances were underwhelming, and the Bimodal-air performed on par with the Touch Screen and Leap Surface keyboards. The Pinch-air performed somewhere in between the other Leap Motion-based keyboards. As these means continue to show, a trend was developing in the dependent measures between the keyboards and was expected to hold for all other variables.

6.2.2.1 KSPC modified-shortest. Participants reached a mean KSPC of 1.1 keystrokes per character ($SD = 0.14$) for the Touch Screen Keyboard, 1.3 keystrokes per character ($SD = 0.27$) for the Leap Surface Keyboard, 1.7 keystrokes per character ($SD = 0.48$) for the Leap Static-air Keyboard, 1.4 keystrokes per character ($SD = 0.31$) for the Leap Pinch-air Keyboard, 1.7 keystrokes per character ($SD = 0.43$) for the Predictive-air Keyboard, and 1.1 keystrokes per character ($SD = 0.15$) for the Leap Bimodal-air Keyboard. Figure 6.7 shows the mean Keystrokes Per Character with the shortest-transcribed modification for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 11.5949$, p-value < 0.0001, $SD_{pooled} = 0.32$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.6, revealed significant differences (p-values < 0.0001) in Keystrokes Per Character with the shortest-transcribed modification between the Touch Screen Keyboard and the Leap Static-air and Predictive-air keyboards. There were significant differences (p-values < 0.05) between the Leap Surface Keyboard and the Leap Static-air and Predictive-air keyboards. There were also significant differences (p-values < 0.001) found between the Leap Bimodal-air Keyboard and the Static-air and Predictive-air keyboards. Finally there was a significant difference (p-value $= 0.0072$) between the Pinch-air and Static-air keyboards.

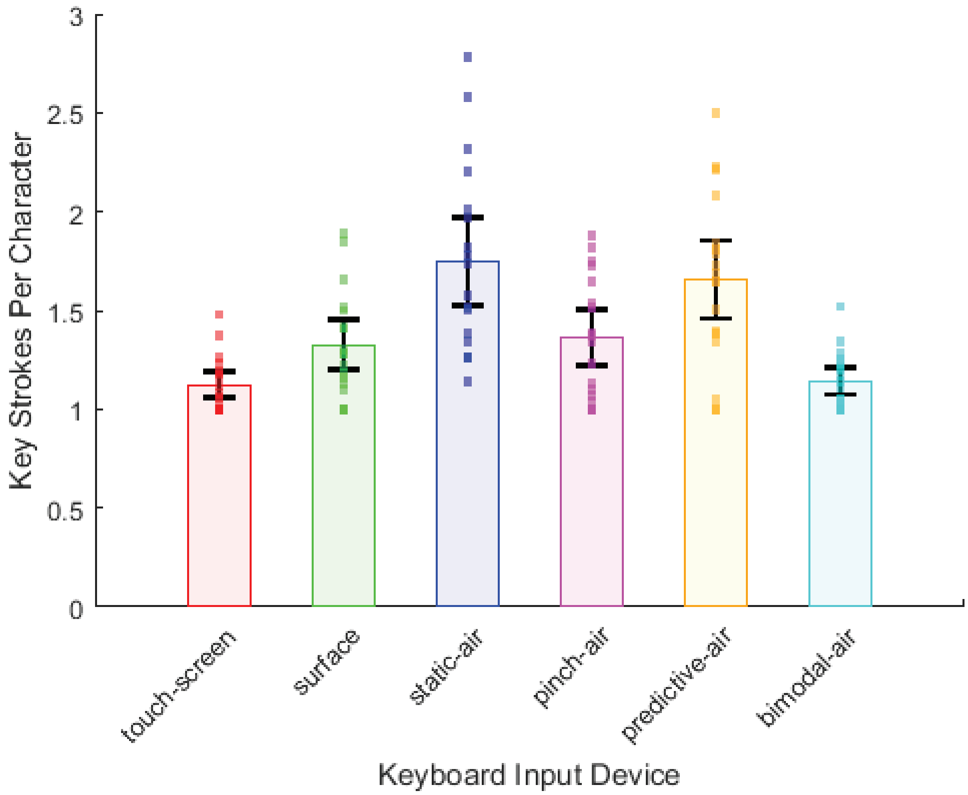


Figure 6.7. Mean Keystrokes Per Character using the shortest-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

As with MWD, KSPC saw huge improvements with the shortest-transcribed modification, especially for Leap Motion-based keyboards. This reaffirmed that a large number of errors were produced during exiting the interaction plane.

6.2.3 Minimum String Distance

6.2.3.1 MSD modified-shortest. Participants reached a mean MSD of 1.6% (SD = 2.4) for the Touch Screen Keyboard, 2.2% (SD = 2.4) for the Leap Surface Keyboard, 6.4% (SD = 5.0) for the Leap Static-air Keyboard, 5.1% (SD = 3.1) for the Leap Pinch-air Keyboard, 5.4% (SD = 3.3) for the Predictive-air Keyboard, and 2.5% (SD = 2.6) for the Leap Bimodal-air Keyboard. Figure 6.8 shows the mean Minimum String Distance with the shortest-transcribed modification for each

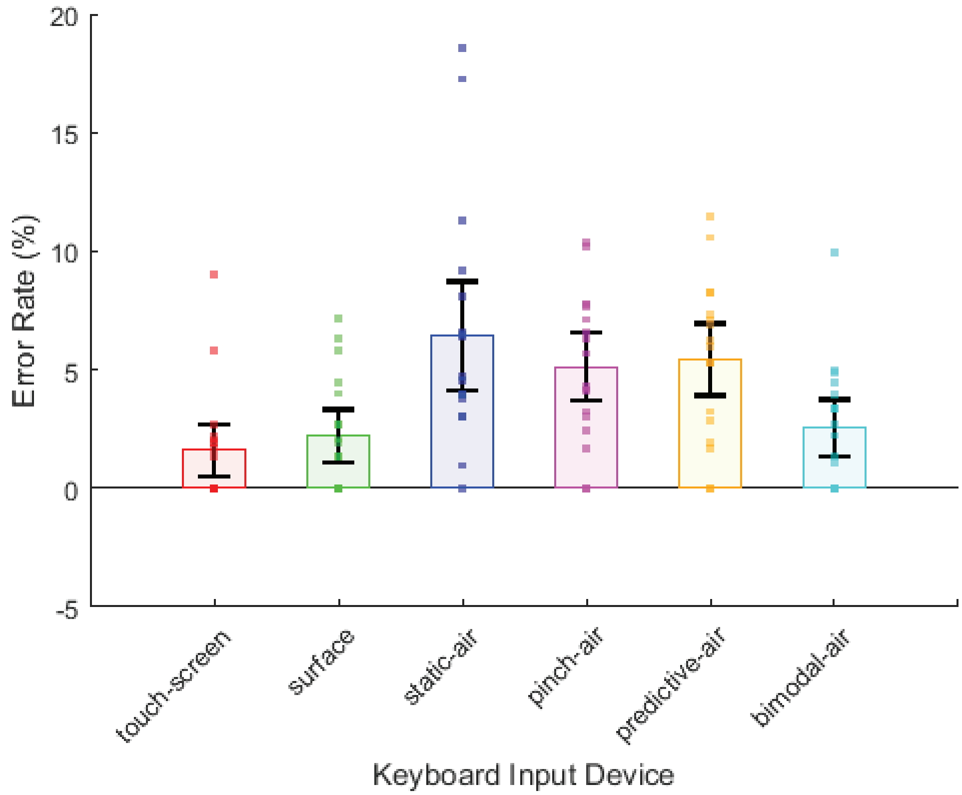


Figure 6.8. Mean Minimum String Distance using the shortest-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 6.8688$, p-value < 0.0001, $SD_{pooled} = 3.3$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.7, revealed significant differences (p-values < 0.05) in Minimum String Distance with the shortest-transcribed modification between the Touch Screen Keyboard and the Leap Static-air, Pinch-air, and Predictive-air keyboards. There were significant differences (p-values < 0.05) between the Leap Surface Keyboard and the Leap Static-air and Predictive-air keyboards both. There was also a significant difference (p-value $= 0.0067$) found between the Leap Bimodal-air and the Static-air keyboards.

The error rates seen for the Pinch-air and Touch Screen keyboards for MSD with the shortest-transcribed modification were similar to those found in Vulture for their MWD (Markussen, Jakobsen, and Hornbæk 2014). It is expected that with practice or repeated sessions, these error rates would become even less. The keyboards again follow the same pattern as other measures, providing more evidence of the effects of decoupling on performance and complexity when using 3-dimensions versus a tangible surface. Notably, the Leap Bimodal-air seems to be mostly unaffected by decoupling between the motor space and display space and seems to be only minimally affected by complexity, only having been significantly different from the Touch Screen Keyboard for text-entry rates.

6.2.4 Total Error Rate

Participants reached a mean Total Error Rate of 4.9% ($SD = 5.5$) for the Touch Screen Keyboard, 9.3% ($SD = 7.1$) for the Leap Surface Keyboard, 23.6% ($SD = 12.1$) for the Leap Static-air Keyboard, 14.1% ($SD = 9.0$) for the Leap Pinch-air Keyboard, 20.1% ($SD = 11.4$) for the Predictive-air Keyboard, and 6.8% ($SD = 5.5$) for the Leap Bimodal-air Keyboard. Figure 6.9 shows the mean Total Error Rate for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 12.9381$, p-value $<$ 0.0001, $SD_{pooled} = 8.8$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.8, revealed significant differences (p-values < 0.0001) in Total Error Rate between the Touch Screen Keyboard and the Leap Static-air and Predictive-air keyboards. There were significant differences (p-values < 0.01) between the Leap Surface Keyboard and the Static-air and Predictive-air keyboards. There were also significant differences (p-values < 0.01) found between the Leap Bimodal-air and the Static-air and Predictive-air keyboards. Finally, there

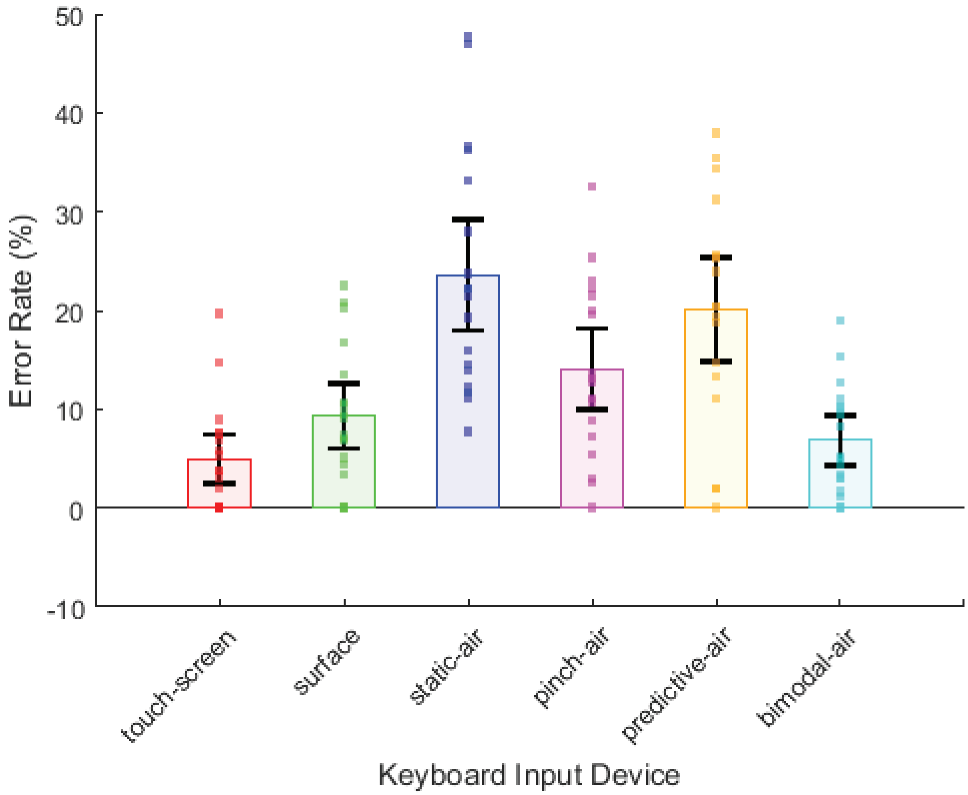


Figure 6.9. Mean Total Error Rates for each keyboard with error bars showing 95% confidence intervals.

was a significant difference (p-value = 0.0191) between the Pinch-air and the Static-air keyboards.

Again, the keyboards follow the same pattern. The Bimodal-air and Leap Surface keyboards perform approximately as well as the Touch Screen Keyboard while the Static-air and Predictive-air keyboards continue to fall short. The Pinch-air keyboard falls somewhere in the middle for Total Error Rate, showing no significant differences with any keyboards.

6.2.4.1 Total error rate modified-shortest. Participants reached a mean Total Error Rate of 4.9% ($SD = 5.4$) for the Touch Screen Keyboard, 9.2% ($SD = 7.0$) for the Leap Surface Keyboard, 23.3% ($SD = 11.8$) for the Leap Static-air Keyboard, 13.5% ($SD = 8.9$) for the Leap Pinch-air Keyboard, 19.4% ($SD = 10.7$) for the Predictive-air Keyboard, and 6.7% ($SD = 5.3$) for the Leap Bimodal-air Keyboard. Figure 6.10 shows the mean Total Error Rate with the shortest-transcribed modification for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 13.0717$, p-value $<$ 0.0001, $SD_{pooled} = 8.6$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.9, revealed significant differences (p-values < 0.05) in Total Error Rate with the shortest-transcribed modification between the Touch Screen Keyboard and the Leap Static-air, Pinch-air, and Predictive-air keyboards. There were significant differences (p-values < 0.01) between the Leap

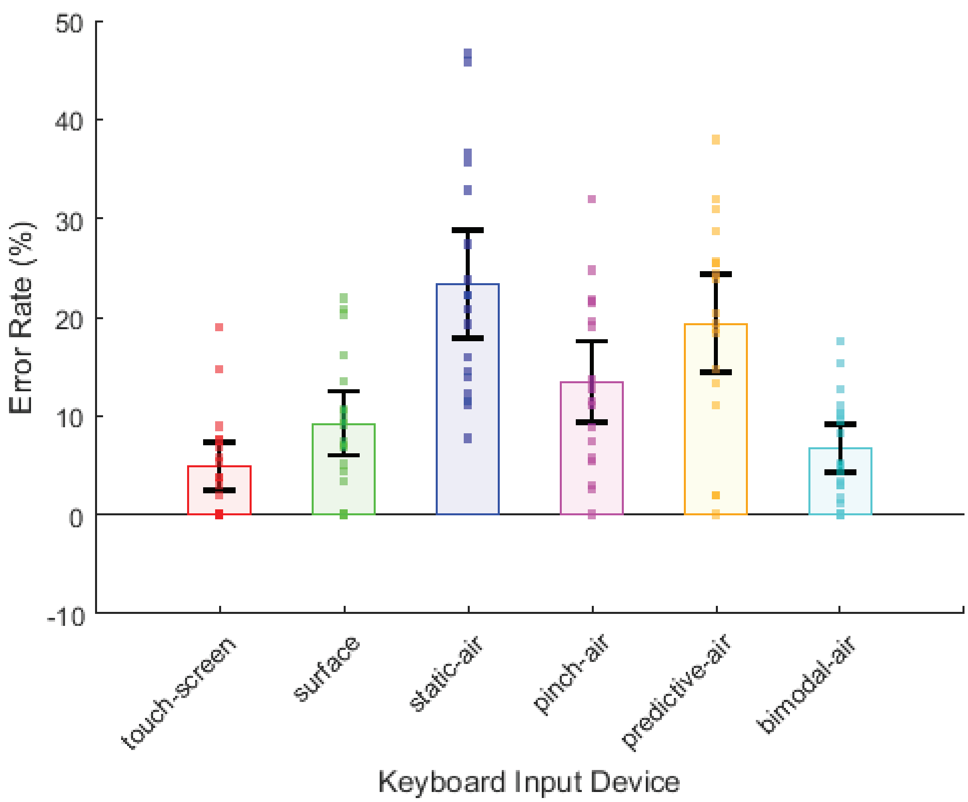


Figure 6.10. Mean Total Error Rate using the shortest-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

Surface Keyboard and the Leap Static-air and Predictive-air keyboards. There were also significant differences found between the Leap Bimodal-air and the Static-air and Predictive-air keyboards with a p-value < 0.001. Finally, there was a significant difference (p-value $= 0.0101$) between the Pinch-air and Static-air keyboards.

Surprisingly, unlike KSPC and MWD, the Total Error Rate did not follow suit by showing large decreases in error rate with the shortest-transcribed modification. However, these observations continue to support the findings between the text-entry rates with and without modification, which were also relatively unaffected by the shortest-transcribed modification. This further implied that errors made by participants were more likely corrected throughout gesturing words rather than after having completed the word-gesture. This was a major downside of lacking word-recognition because gestures were interrupted and corrected rather than being completed with errors, as is natural to traditional word-gesture keyboards.

6.3 Correctness

6.3.1 Fréchet Distance

Participants reached a mean Fréchet Distance of 231.2 pixels ($SD = 26.7$) for the Touch Screen Keyboard, 242.8 pixels ($SD = 30.4$) for the Leap Surface Keyboard, 364.3 pixels ($SD = 86.5$) for the Leap Static-air Keyboard, 501.2 pixels ($SD = 42.8$) for the Leap Pinch-air Keyboard, 350.6 pixels ($SD = 86.8$) for the Predictive-air Keyboard, and 396.5 pixels ($SD = 102.1$) for the Leap Bimodal-air Keyboard. Figure 6.11 shows the mean Fréchet Distance for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 37.9416$, p-value < 0.0001, $SD_{pooled} = 69.4$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.10, revealed significant differences (p-values < 0.0001) in Fréchet Distance between the Touch Screen Keyboard and

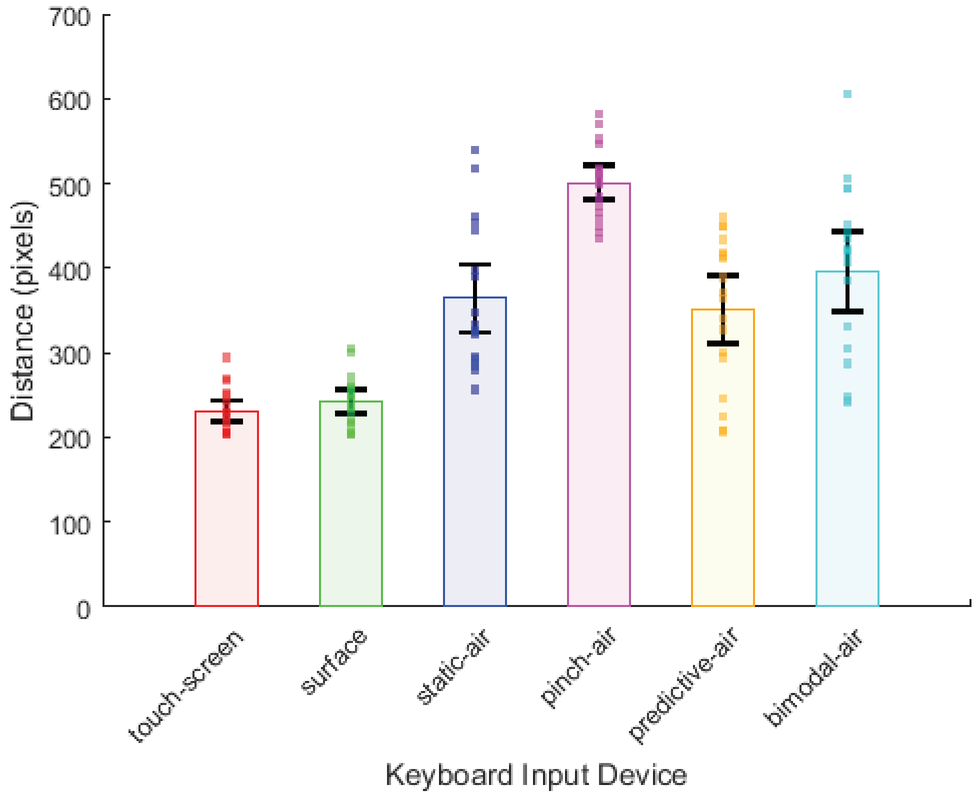


Figure 6.11. Mean Fréchet Distance for each keyboard with error bars showing 95% confidence intervals.

all of the mid-air keyboards. There were significant differences (p-values < 0.001) between the Leap Surface Keyboard and all of the mid-air keyboards. There were also significant differences (p-values < 0.001) found between the Leap Pinch-air and all of the other mid-air keyboards.

The Fréchet Distance was the only place where the trends were broken and the Bimodal-air keyboard performed significantly worse than the Touch Screen and Leap Surface keyboards. The Pinch-air Keyboard also performed significantly worse than all other keyboards. These results were assumed to be caused by a combination of the word-gesture velocity seen in Section 6.4.2 and the effects of either increased decoupling or complexity from working with the 3rd-dimension. Appendix C emphasizes these assumptions by illustrating the participant-generated word-gesture paths for all word sets and their corresponding keyboards.

The Leap Bimodal-air seemed to suffer in correctness due to its very high word-gesture velocities, whereas the Static-air and Predictive-air benefit in correctness due to their very slow word-gesture velocities. Participants used explicitly slower, more precise gestures to compensate for these keyboards' increased decoupling. The Leap Surface and Touch Screen keyboards were relatively unaffected in correctness by their high word-gesturing velocities due to their decreased decoupling of the word-gesture motor space from the displayed screen.

The Pinch-air Keyboard suffered from the combination of high word-gesture velocities and effects of increased decoupling. However, the Pinch-air Keyboard also seemed to be heavily affected by the deviation in its tracking method, which increased the overall distance traveled for each word-gesture, as seen in Section 6.4.1. All other mid-air keyboards tracked participants' pointer fingers, whereas the Pinch-air keyboard tracked participants' palms. Palms were tracked instead because participants' fingers moved drastically when creating a pinching gesture. This difference in tracking combined with high word-gesture velocities were the assumed factors for this decrease in performance.

6.3.1.1 Fréchet distance modified-shortest. Participants reached a mean Fréchet Distance of 221.0 pixels (SD = 18.0) for the Touch Screen Keyboard, 231.0 pixels (SD = 22.7) for the Leap Surface Keyboard, 306.4 pixels (SD = 53.2) for the Leap Static-air Keyboard, 459.2 pixels (SD = 20.1) for the Leap Pinch-air Keyboard, 281.6 pixels (SD = 49.9) for the Predictive-air Keyboard, and 368.4 pixels (SD = 82.4) for the Leap Bimodal-air Keyboard. Figure 6.12 shows the mean Fréchet Distance with the shortest-transcribed modification for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between

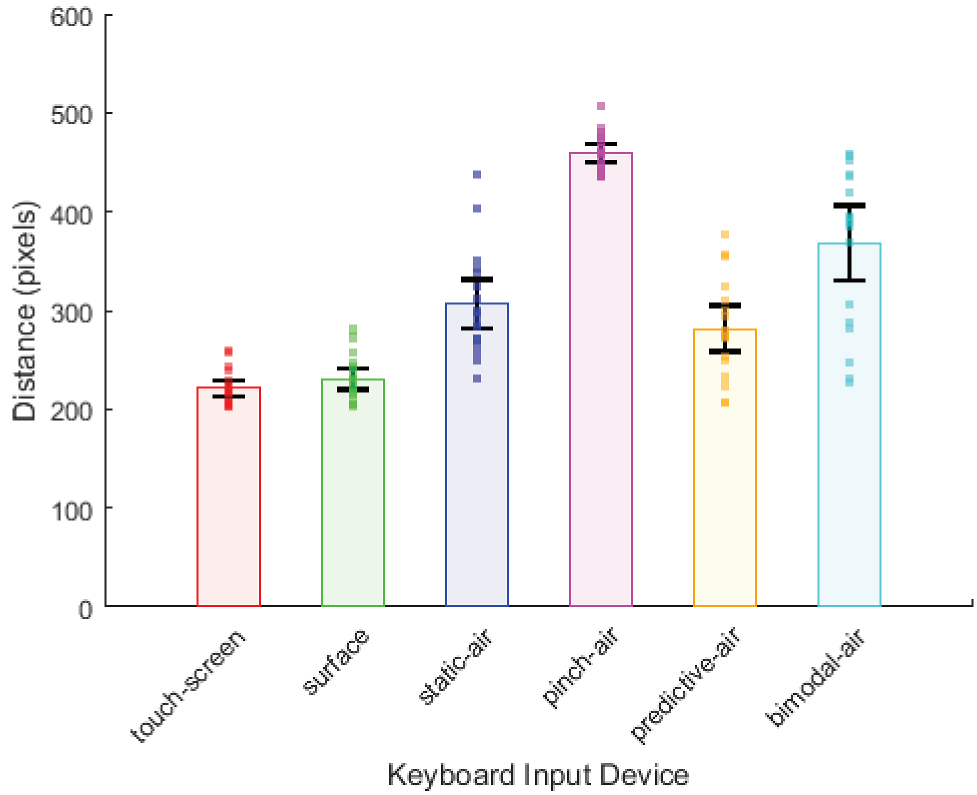


Figure 6.12. Mean Fréchet Distance using the shortest-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

keyboard means ($F(5, 102) = 65.7440$, p-value < 0.0001, $SD_{pooled} = 47.2$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.11, revealed significant differences (p-values < 0.01) in Fréchet Distance with the shortest-transcribed modification between the Touch Screen Keyboard and all of the mid-air keyboards. There were significant differences (p-values < 0.05) between the Leap Surface Keyboard and all of the mid-air keyboards. There were also significant differences (p-values < 0.01) found between the Leap Bimodal-air Keyboard and the other mid-air keyboards. Finally, there were significant differences (p-values < 0.0001) found between the Leap Pinch-air Keyboard and the Static-air and Predictive-air keyboards.

The increased correctness for all keyboards, especially the mid-air keyboards, when using the shortest-transcribed modification implies that many errors were being produced when exiting the interaction plane. This was especially evident for the Static-air, Pinch-air, and Predictive-air keyboards, which all saw the largest improvements to correctness. For the Static-air and Predictive-air keyboards, these errors were caused by the addition of the 3rd-dimension and participants having trouble entering and exiting the interaction plane accurately. The Pinch-air Keyboard saw a combination of high word-gesture velocity and a different method of tracking that was assumed to produce these kinds of errors.

6.3.1.2 Fréchet distance modified-backspace. Participants reached a mean Fréchet Distance of 212.3 pixels ($SD = 9.3$) for the Touch Screen Keyboard, 223.5 pixels ($SD = 18.5$) for the Leap Surface Keyboard, 258.9 pixels ($SD = 33.5$) for the Leap Static-air Keyboard, 444.7 pixels ($SD = 8.0$) for the Leap Pinch-air Keyboard, 237.7 pixels ($SD = 24.8$) for the Predictive-air Keyboard, and 356.8 pixels ($SD = 85.0$) for the Leap Bimodal-air Keyboard. Figure 6.13 shows the mean Fréchet Distance with the backspace-transcribed modification for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 97.1196$, p-value < 0.0001, $SD_{pooled} = 39.7$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.12, revealed significant differences (p-values < 0.01) in Fréchet Distance with the backspace-transcribed modification between the Touch Screen Keyboard and the Leap Static-air, Pinch-air, and Bimodal-air keyboards. There were significant differences (p-values < 0.0001) between the Leap Surface Keyboard and the Leap Pinch-air and Bimodal-air keyboards. There were also significant differences (p-values < 0.0001) found between the Leap Bimodal-air and all other mid-air keyboards. Finally, there were significant differences (p-values

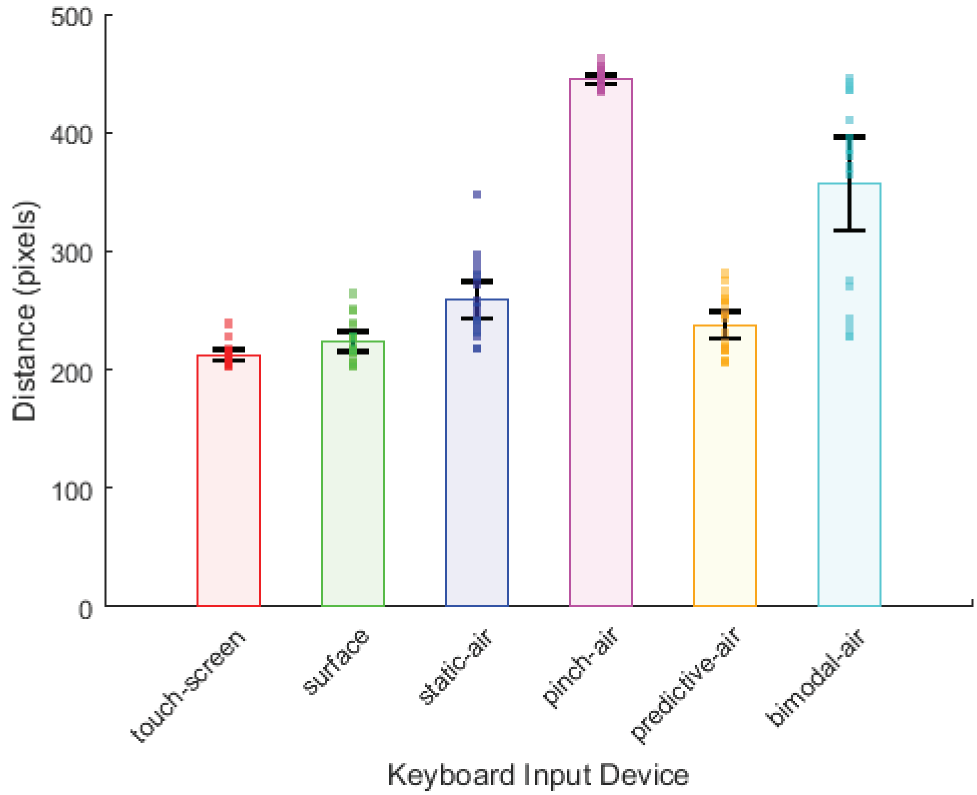


Figure 6.13. Mean Fréchet Distance using the backspace-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

< 0.0001) between the Leap Pinch-air Keyboard and the Static-air and Predictive-air keyboards.

If the backspace key was included in the path considered to be correct for Fréchet Distance, since participants were required to correct words, a boost in correctness was observed. The Bimodal-air and Pinch-air keyboards still suffered from the previously mentioned problems in Section 6.3.1. The Static-air and Predictive-air, however, saw huge improvements in correctness because of the low word-gesture velocities and high error rates. The Predictive-air Keyboard, in this case, was not significantly different from the Touch Screen and Leap Surface keyboards, implying that participants were relatively attentive to correcting errors for it.

6.4 Distance Measures

6.4.1 Word-gesture Distance

Participants reached a mean Word-gesture Distance of 1087.5 pixels ($SD = 85.9$) for the Touch Screen Keyboard, 1423.9 pixels ($SD = 246.0$) for the Leap Surface Keyboard, 1590.9 pixels ($SD = 508.2$) for the Leap Static-air Keyboard, 2165.3 pixels ($SD = 540.5$) for the Leap Pinch-air Keyboard, 1634.7 pixels ($SD = 481.4$) for the Predictive-air Keyboard, and 1627.7 pixels ($SD = 321.3$) for the Leap Bimodal-air Keyboard. Figure 6.14 shows the mean Word-gesture Distance for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 13.9225$, p-value < 0.0001, $SD_{pooled} = 398.6$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.13, revealed significant differences (p-values < 0.01) in Word-gesture Distance between the Touch Screen Keyboard and all of the mid-air keyboards. There were also significant differences (p-values < 0.01) between the Leap Pinch-air and all other Leap Motion-based keyboards.

As expected, all mid-air keyboards required longer word-gesture distances than the Touch Screen Keyboard due to the effect of decoupling between the word-gesture motor space and the keyboard display. It was surprising, however, that the Leap Surface Keyboard required significantly more word-gesture distance than the Touch Screen Keyboard. This could be because of the requirement to use a stylus rather than the participants' bare hands.

The Pinch-air keyboard required a significantly longer word-gesture distance than all other keyboards. This was assumed to be caused by the different tracking method as previously mentioned. Tracking the palm required broader, less controlled movements than tracking fingers or a stylus.

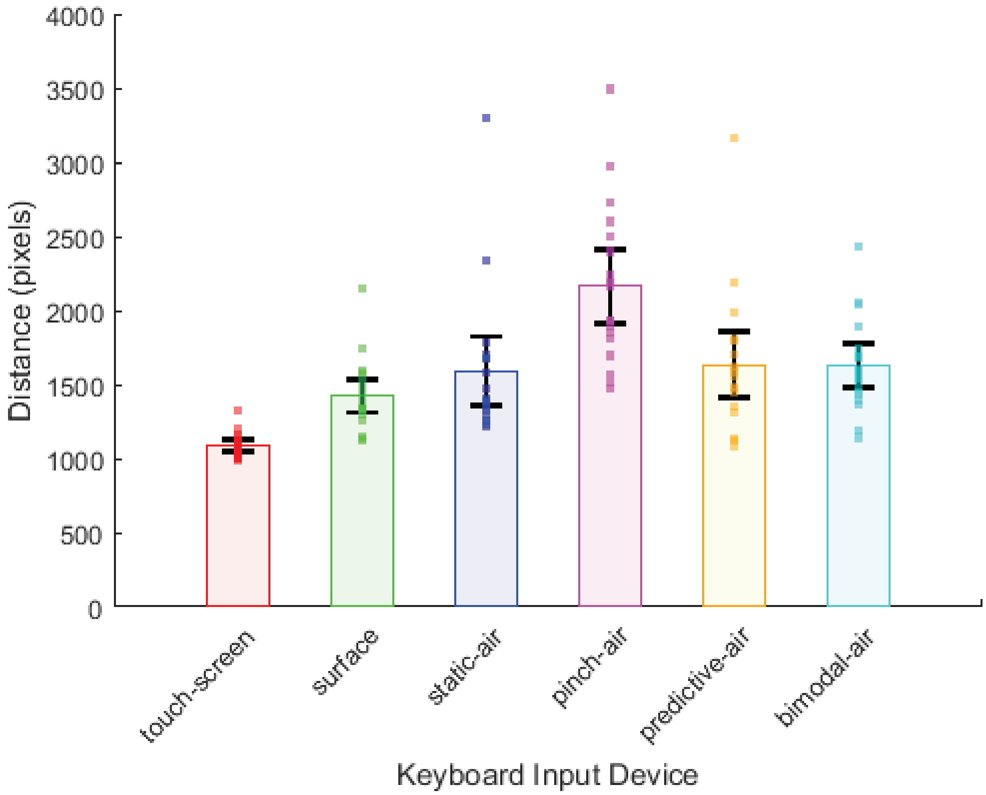


Figure 6.14. Mean Word-gesture Distance for each keyboard with error bars showing 95% confidence intervals.

6.4.1.1 Word-gesture distance modified-shortest. Participants reached a mean Word-gesture Distance of 1081.3 ($SD = 84.6$) for the Touch Screen Keyboard, 1396.5 ($SD = 226.5$) for the Leap Surface Keyboard, 1524.8 ($SD = 412.5$) for the Leap Static-air Keyboard, 1981.4 ($SD = 470.6$) for the Leap Pinch-air Keyboard, 1536.7 ($SD = 346.1$) for the Predictive-air Keyboard, and 1566.5 ($SD = 253.6$) for the Leap Bimodal-air Keyboard. Figure 6.15 shows the mean Word-gesture Distance with the shortest-transcribed modification for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 14.403$, p-value < 0.0001, $SD_{pooled} = 325.1$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

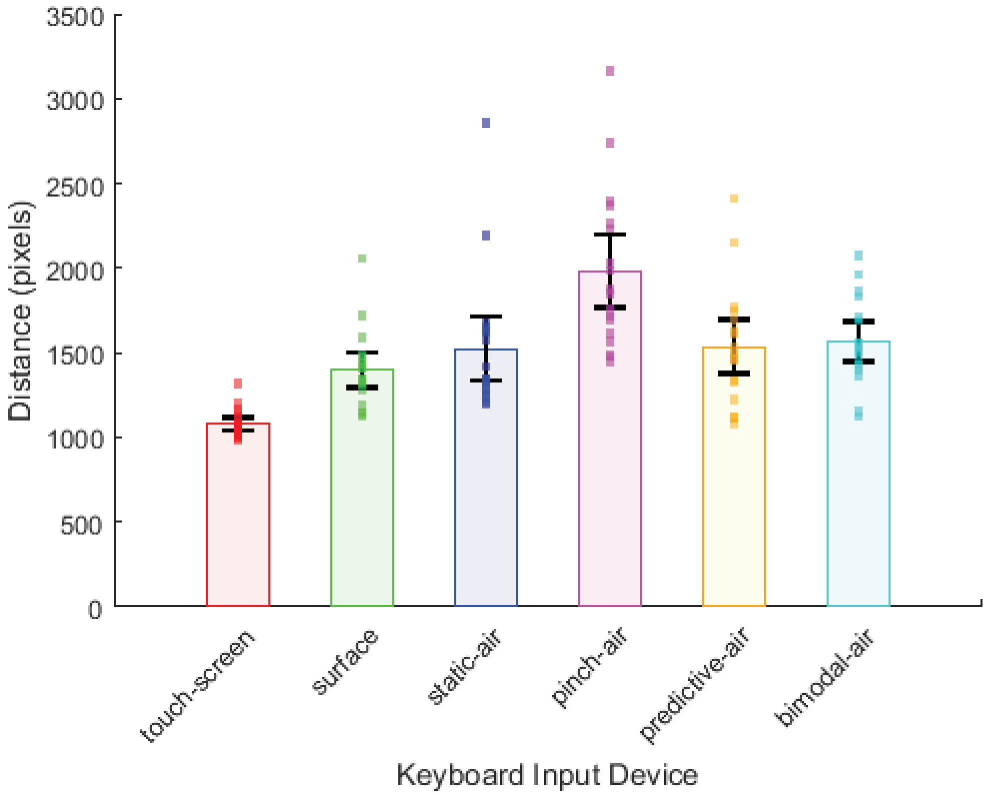


Figure 6.15. Mean Word-gesture Distance using the shortest-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

The multiple comparisons, seen in Table D.14, revealed significant differences (p-values < 0.05) in Word-gesture Distance with the shortest-transcribed modification between the Touch Screen Keyboard and all other keyboards. There were significant differences (p-values < 0.01) between the Leap Surface and Pinch-air keyboards and all other Leap Motion-based keyboards.

Again, the small difference between the word-gesture distance and the word-gesture distance using the shortest-transcribed modification implied that most errors were corrected by participants while in the middle of typing words.

6.4.2 *Word-gesture Velocity*

Participants reached a mean Word-gesture Velocity of 673.3 $pixels/s$ ($SD = 126.4$) for the Touch Screen Keyboard, 641.1 $pixels/s$ ($SD = 165.5$) for the Leap Surface Keyboard, 380.9 $pixels/s$ ($SD = 78.2$) for the Leap Static-air Keyboard, 643.8 $pixels/s$ ($SD = 204.7$) for the Leap Pinch-air Keyboard, 405.6 $pixels/s$ ($SD = 87.6$) for the Predictive-air Keyboard, and 777.6 $pixels/s$ ($SD = 240.9$) for the Leap Bimodal-air Keyboard. Figure 6.16 shows the mean Word-gesture Velocity for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 17.233$, p-value < 0.0001, $SD_{pooled} = 161.8$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

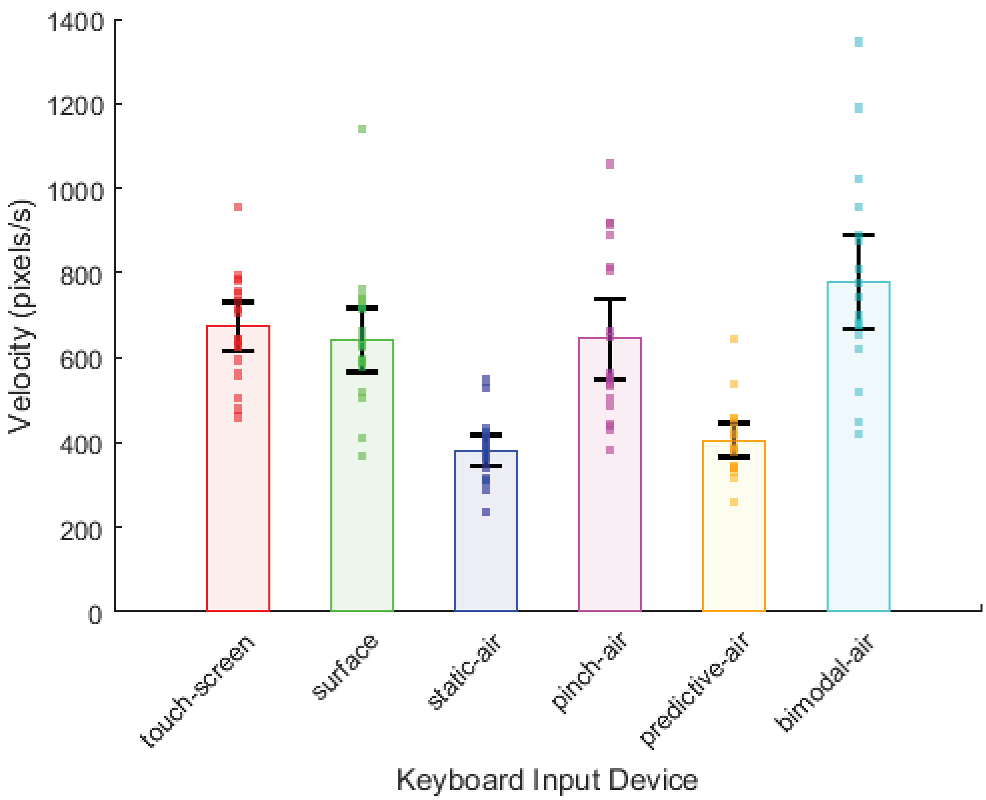


Figure 6.16. Mean Word-gesture Velocity for each keyboard with error bars showing 95% confidence intervals.

The multiple comparisons, seen in Table D.15, revealed significant differences (p-values < 0.0001) in Word-gesture Velocity between the Touch Screen Keyboard and the Leap Static-air and Predictive-air keyboards. There were significant differences (p-values < 0.001) between the Leap Surface Keyboard and the Static-air and Predictive-air keyboards. There were also significant differences (p-values < 0.0001) found between the Leap Bimodal-air and the Static-air and Predictive-air keyboards. Finally, there were significant differences (p-values < 0.001) found between the Leap Pinch-air and the Static-air and Predictive Air keyboards.

The Static-air and Predictive-air keyboards were significantly slower than all of the other keyboards for word-gesture velocity. This was expected due to the enhanced precision required when working with a 3rd-dimension in mid-air. It is interesting that these two keyboards also had the highest error rates across all error measures, implying a very high degree of difficulty, even with slower, more precise movements.

6.4.3 Hand Velocity

Participants reached a mean Hand Velocity of 35.7 cm/s ($SD = 6.7$) for the Touch Screen Keyboard, 15.0 cm/s ($SD = 3.9$) for the Leap Surface Keyboard, 6.9 cm/s ($SD = 1.4$) for the Leap Static-air Keyboard, 10.5 cm/s ($SD = 3.2$) for the Leap Pinch-air Keyboard, 7.6 cm/s ($SD = 1.7$) for the Predictive-air Keyboard, and 15.2 cm/s ($SD = 5.1$) for the Leap Bimodal-air Keyboard. Figure 6.17 shows the mean Hand Velocity for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 121.9826$, p-value < 0.0001, $SD_{pooled} = 4.1$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.16, revealed significant differences (p-values < 0.0001) in Hand Velocity between the Touch Screen Keyboard and all other keyboards. There were significant differences (p-values < 0.05) between the Leap Surface Keyboard and the Static-air, Pinch-air, and Predictive-air keyboards.

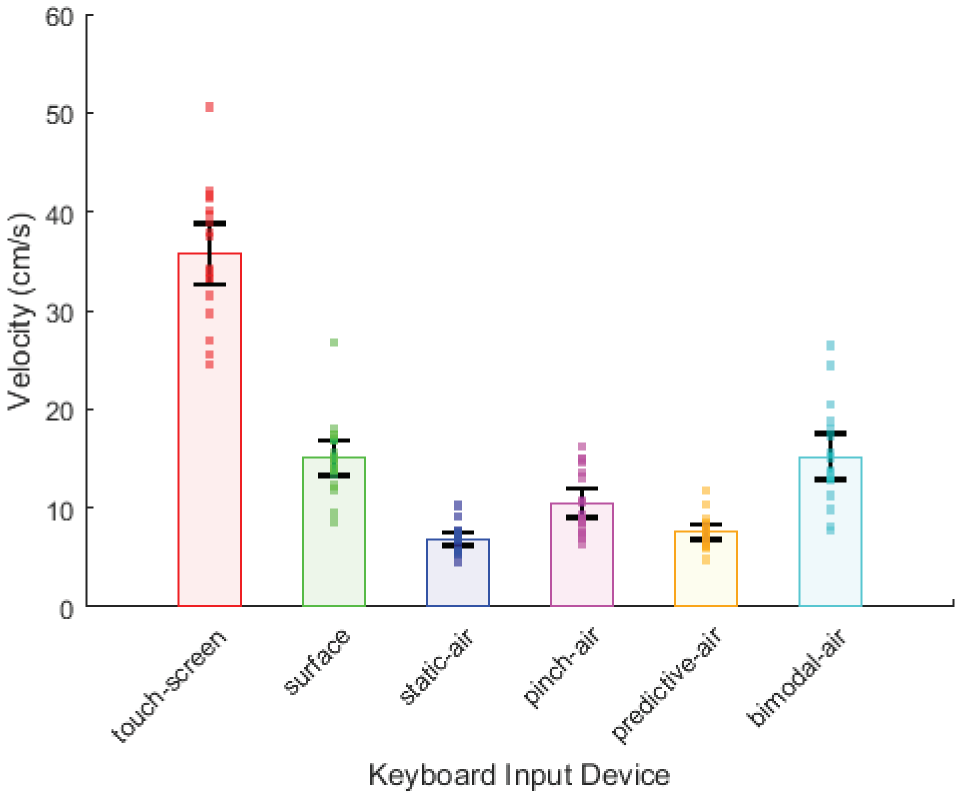


Figure 6.17. Mean Hand Velocity for each keyboard with error bars showing 95% confidence intervals.

There were also significant differences (p-values < 0.05) found between the Leap Bimodal-air and all other mid-air keyboards.

The hand velocities were all similar to the word-gesture velocities in Section 6.4.2, except for the Touch Screen Keyboard. This result was simply caused by the physical size difference between the interaction planes and the size of the Touch Screen, as mentioned in Chapter 3.

6.5 Time-based Measures

6.5.1 Word-gesture Duration

Participants reached a mean Word-gesture Duration of 1.65 s ($SD = 0.46$) for the Touch Screen Keyboard, 2.03 s ($SD = 1.17$) for the Leap Surface Keyboard, 4.01 s (SD = 1.17) for the Leap Static-air Keyboard, 3.36 s (SD = 0.76) for the Leap Pinch-air Keyboard, 3.87 s (SD = 1.18) for the Predictive-air Keyboard, and 2.18 s (SD = 0.36) for the Leap Bimodal-air Keyboard. Figure 6.18 shows the mean Word-gesture Duration for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 28.1205$, p-value < 0.0001, $SD_{pooled} = 0.81$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.17, revealed significant differences (p-values < 0.0001) in Word-gesture Duration between the Touch Screen Keyboard and the Leap Static-air, Pinch-air, and Predictive-air keyboards. There were significant differences (p-values < 0.001) between the Leap Surface Keyboard and the

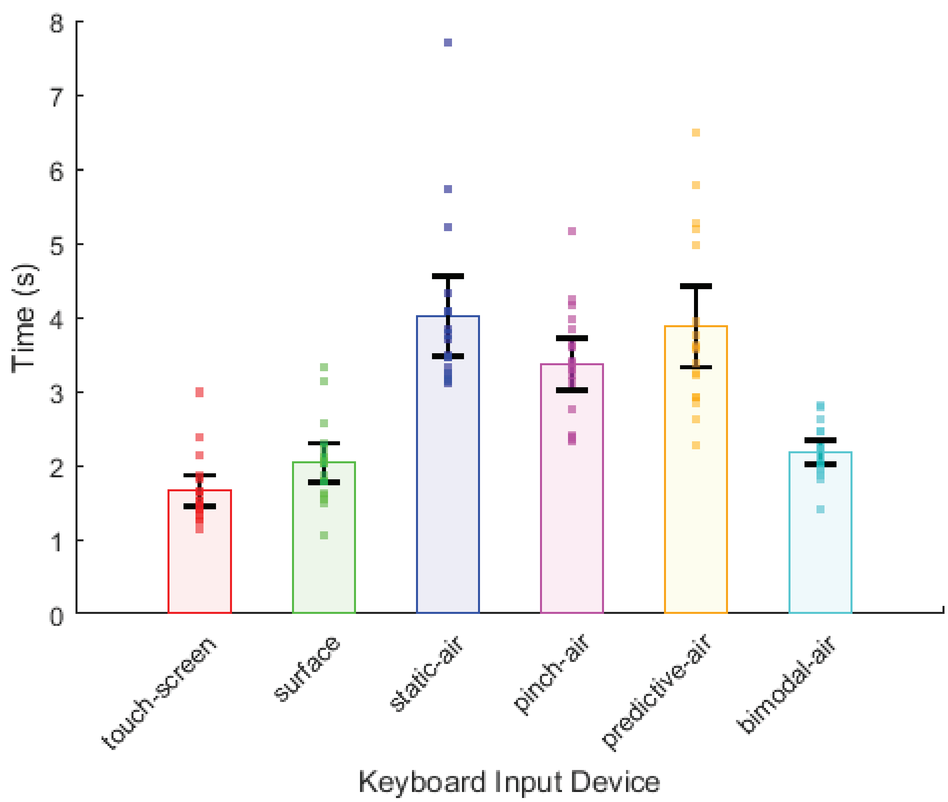


Figure 6.18. Mean Word-gesture Duration for each keyboard with error bars showing 95% confidence intervals.

Static-air, Pinch-air, and Predictive-air keyboards. There were also significant differences (p-values < 0.001) found between the Leap Bimodal-air and all of the other mid-air keyboards.

As expected, the Static-air, Pinch-air, and Predictive-air keyboards had significantly longer word-gesture durations than the other keyboards. For the Static-air and Predictive-air, lower velocities and higher decoupling meant that word-gestures take longer to perform, whereas for the Pinch-air Keyboard, the longer distances traveled influenced the longer durations. The Bimodal-air keyboard performed as well as the Touch Screen and Leap Surface keyboards, as expected from its prior trends.

6.5.1.1 Word-gesture duration modified-shortest. Participants reached a mean Word-gesture Duration of 1.60 s ($SD = 0.40$) for the Touch Screen Keyboard, 2.00 s ($SD = 0.55$) for the Leap Surface Keyboard, 3.75 s ($SD = 0.93$) for the Leap Static-air Keyboard, 3.17 s ($SD = 0.73$) for the Leap Pinch-air Keyboard, 3.56 s ($SD = 0.86$) for the Predictive-air Keyboard, and 2.16 s ($SD = 0.36$) for the Leap Bimodal-air Keyboard. Figure 6.19 shows the mean Word-gesture Duration with the shortest-transcribed modification for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 32.0355$, p-value < 0.0001, $SD_{pooled} = 0.67$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.18, revealed significant differences (p-values < 0.0001) in Word-gesture Duration with the shortest-transcribed modification between the Touch Screen Keyboard and the Leap Static-air, Pinch-air, and Predictive-air keyboards. There were significant differences (p-values < 0.0001) between the Leap Surface Keyboard and the Leap Static-air, Pinch-air, and Predictive-air keyboards. There were also significant differences (p-values < 0.001) found between the Leap Bimodal-air and all of the other mid-air keyboards.

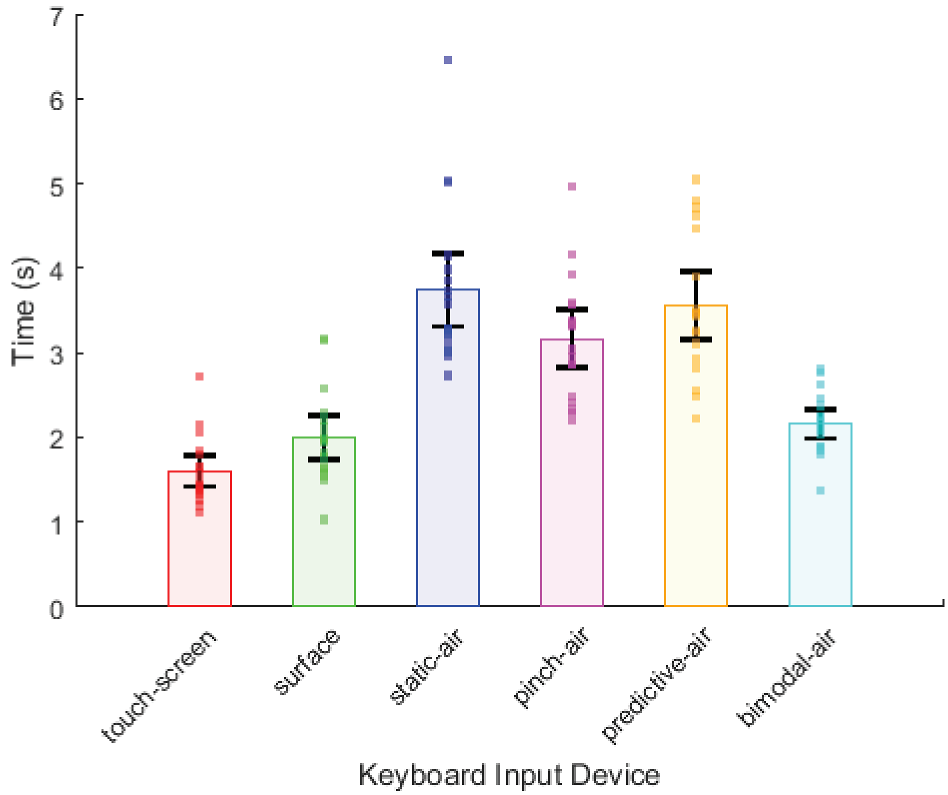


Figure 6.19. Mean Word-gesture Dur using the shortest-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

As seen before, the shortest-transcribed modification for word-gesture duration had little effect on the total length of word-gesture durations. This again supported the idea that participants mostly corrected errors as they were being made during transcription.

6.5.2 Reaction Time to Errors

Participants reached a mean Reaction Time to Errors of 0.22 s ($SD = 0.23$) for the Touch Screen Keyboard, 0.40 s ($SD = 0.29$) for the Leap Surface Keyboard, 1.02 s ($SD = 0.43$) for the Leap Static-air Keyboard, 0.59 s ($SD = 0.30$) for the Leap Pinch-air Keyboard, 0.78 s ($SD = 0.40$) for the Predictive-air Keyboard, and 0.29 s ($SD = 0.24$) for the Leap Bimodal-air Keyboard. Figure 6.20 shows the

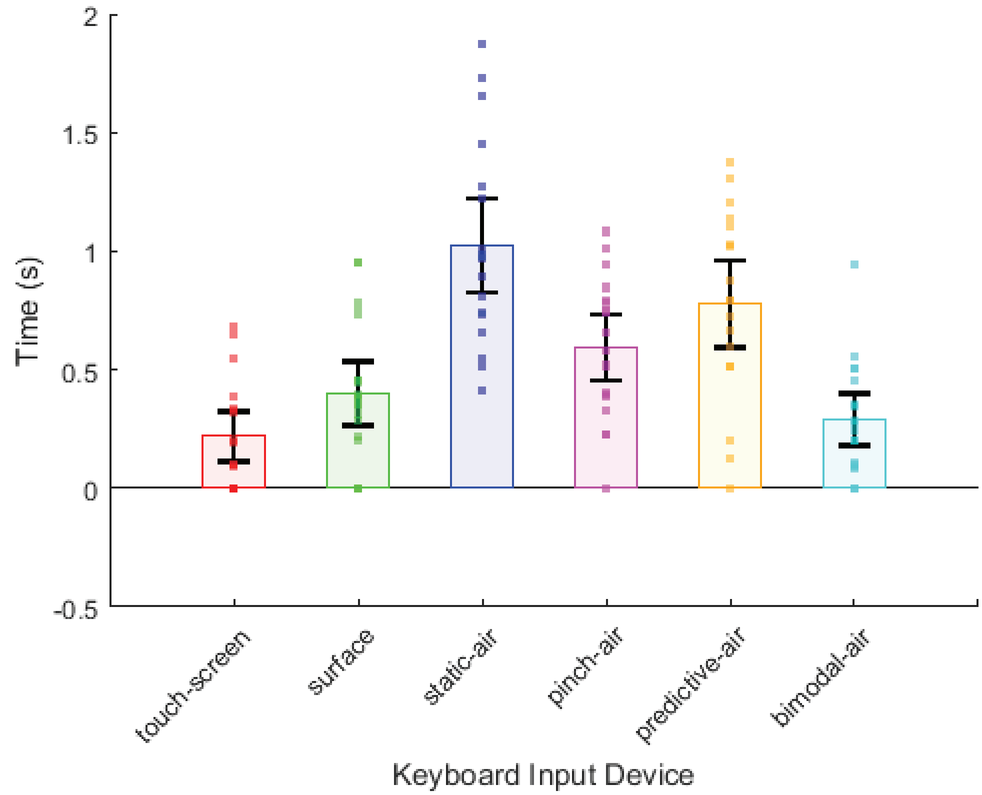


Figure 6.20. Mean Reaction Time to Errors for each keyboard with error bars showing 95% confidence intervals.

mean Reaction Time to Errors for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 16.3385$, p-value < 0.0001, $SD_{pooled} = 0.32$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.19, revealed significant differences (p-values < 0.01) in Reaction Time to Errors between the Touch Screen Keyboard and the Leap Static-air, Pinch-air, and Predictive-air keyboards. There were significant differences (p-values < 0.01) between the Leap Surface Keyboard and the Static-air and Predictive-air keyboards. There were also significant differences (p-values < 0.001) found between the Leap Bimodal-air and the Static-air and Predictive-air keyboards. Finally, there was a significant difference (p-value $= 0.0017$) found between the Pinch-air and Static-air keyboards.

As expected, keyboards involving the 3rd-dimension had significantly slower response times to errors than keyboards that did not. This implied that these keyboards required a higher degree of focus due to increased decoupling between the motor space and display.

6.5.3 Reaction Time to Simulate Touch

Participants reached a mean Reaction Time to Simulate Touch of 1.24 s ($SD = 0.21$) for the Touch Screen Keyboard, 1.22 s ($SD = 0.40$) for the Leap Surface Keyboard, 2.60 s ($SD = 0.82$) for the Leap Static-air Keyboard, 1.71 s ($SD = 0.37$) for the Leap Pinch-air Keyboard, 2.23 s ($SD = 0.83$) for the Predictive-air Keyboard, and 1.41 s ($SD = 0.23$) for the Leap Bimodal-air Keyboard. Figure 6.21 shows the mean Reaction Time to Simulate Touch for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 19.6476$, p-value < 0.0001, $SD_{pooled} = 0.54$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.20, revealed significant differences (p-values < 0.0001) in Reaction Time to Simulate Touch between the Touch Screen Keyboard and the Leap Static-air and Predictive-air keyboards. There were significant differences (p-values < 0.0001) between the Leap Surface Keyboard and the Static-air and Predictive-air keyboards. There were also significant differences (p-values < 0.001) found between the Leap Bimodal-air and the Static-air and Predictive-air keyboards. Finally, a significant difference (p-value $= 0.0001$) was detected between the Pinch-air and Static-air keyboards.

Again, the reaction time to simulate a touch was significantly slower for the Static-air and Predictive-air keyboards due to the introduction of the 3rd-dimension and increased decoupling between the word-gesture motor space and the display.

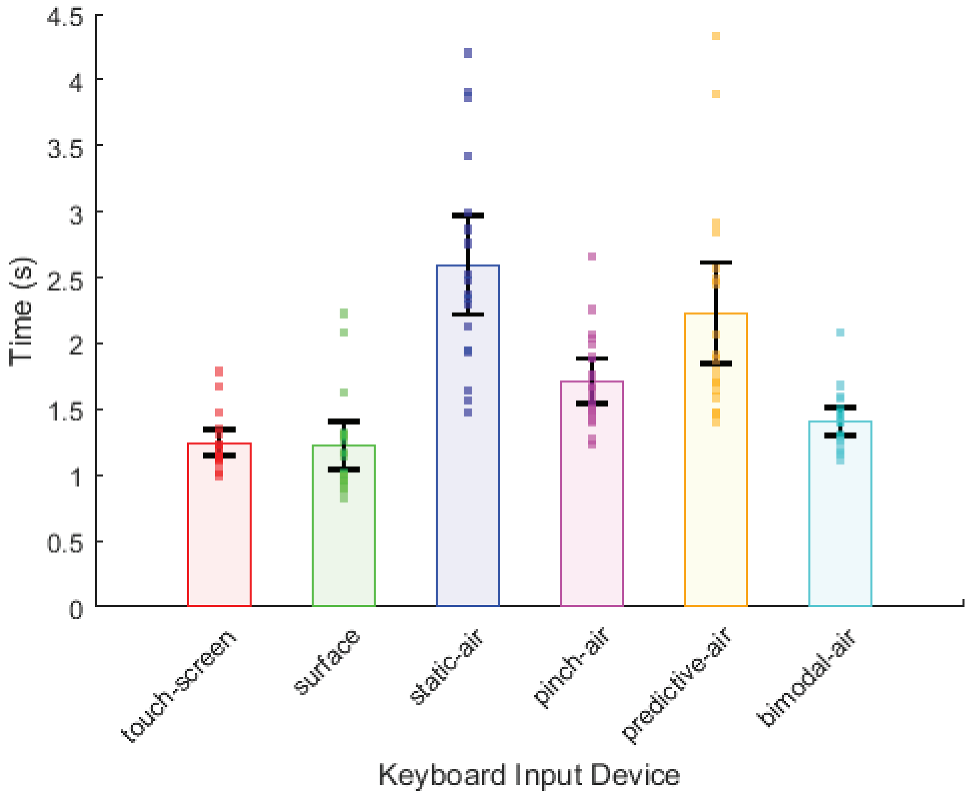


Figure 6.21. Mean Reaction Time to Simulate Touch for each keyboard with error bars showing 95% confidence intervals.

6.5.4 Reaction Time for First Correct Letter

Participants reached a mean Reaction Time for First Correct Letter of 1.44 s ($SD = 0.33$) for the Touch Screen Keyboard, 1.92 s ($SD = 0.70$) for the Leap Surface Keyboard, 3.68 s ($SD = 1.52$) for the Leap Static-air Keyboard, 2.34 s ($SD = 0.68$) for the Leap Pinch-air Keyboard, 3.40 s ($SD = 1.40$) for the Predictive-air Keyboard, and 1.49 s ($SD = 0.31$) for the Leap Bimodal-air Keyboard. Figure 6.22 shows the mean Reaction Time for First Correct Letter for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 18.3416$, p-value < 0.0001, $SD_{pooled} = 0.95$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

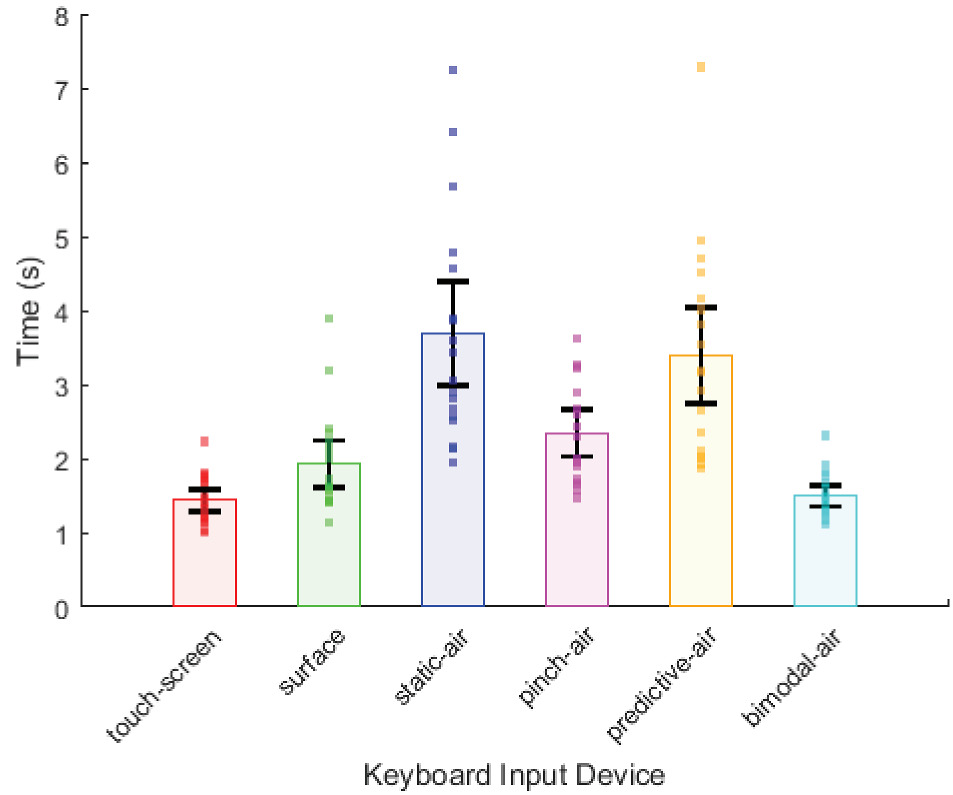


Figure 6.22. Mean Reaction Time for First Correct Letter for each keyboard with error bars showing 95% confidence intervals.

The multiple comparisons, seen in Table D.21, revealed significant differences (p-values < 0.0001) in Reaction Time for First Correct Letter between the Touch Screen Keyboard and the Leap Static-air and Predictive-air keyboards. There were significant differences (p-values < 0.001) between the Leap Surface Keyboard and the Static-air and Predictive-air keyboards. There were also significant differences (p-values < 0.0001) found between the Leap Bimodal-air and the Static-air and Predictive-air keyboards. Finally, there were significant differences (p-values < 0.05) found between the Leap Pinch-air and the Static-air and Predictive-air keyboards.

The Leap Bimodal-air Keyboard performed at almost exactly the same rate as the Touch Screen Keyboard for entering the first correct character of each word. Although not a significant difference, it should also be noted that the Leap Surface

Keyboard performed slightly slower than both the Bimodal-air and Touch Screen keyboards because participants had a tendency to look down at the printed keyboard. If participants had looked at the on-screen keyboard instead, there would have been a major increase in decoupling that would have further slowed the use of the Leap Surface Keyboard.

Again, the reaction time to simulate a touch was significantly slower for the Static-air and Predictive-air keyboards due to the introduction of the 3rd-dimension and increased decoupling between the word-gesture motor space and the display.

6.6 *Additional Quantitative Measures*

6.6.1 *Number of Touches Simulated*

Participants reached a mean Number of Touches Simulated of 1.58 simulations ($SD = 0.79$) for the Touch Screen Keyboard, 1.62 simulations ($SD = 0.55$) for the Leap Surface Keyboard, 3.77 simulations ($SD = 2.21$) for the Leap Static-air Keyboard, 2.42 simulations ($SD = 1.05$) for the Leap Pinch-air Keyboard, 3.33 simulations ($SD = 1.62$) for the Predictive-air Keyboard, and 1.67 simulations ($SD = 0.53$) for the Leap Bimodal-air Keyboard. Figure 6.23 shows the mean Number of Touches Simulated for each keyboard interaction style. Using a one-way ANOVA, a significant difference was detected between keyboard means ($F(5, 102) = 10.0290$, p-value < 0.0001, $SD_{pooled} = 1.28$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.22, revealed significant differences (p-values < 0.01) in Number of Touches Simulated between the Touch Screen Keyboard and the Leap Static-air and Predictive-air keyboards. There were significant differences (p-values < 0.01) between the Leap Surface Keyboard and the Static-air and Predictive-air keyboards. There were also significant differences (p-values < 0.01) found between the Leap Bimodal-air and the Static-air and Predictive-air keyboards.

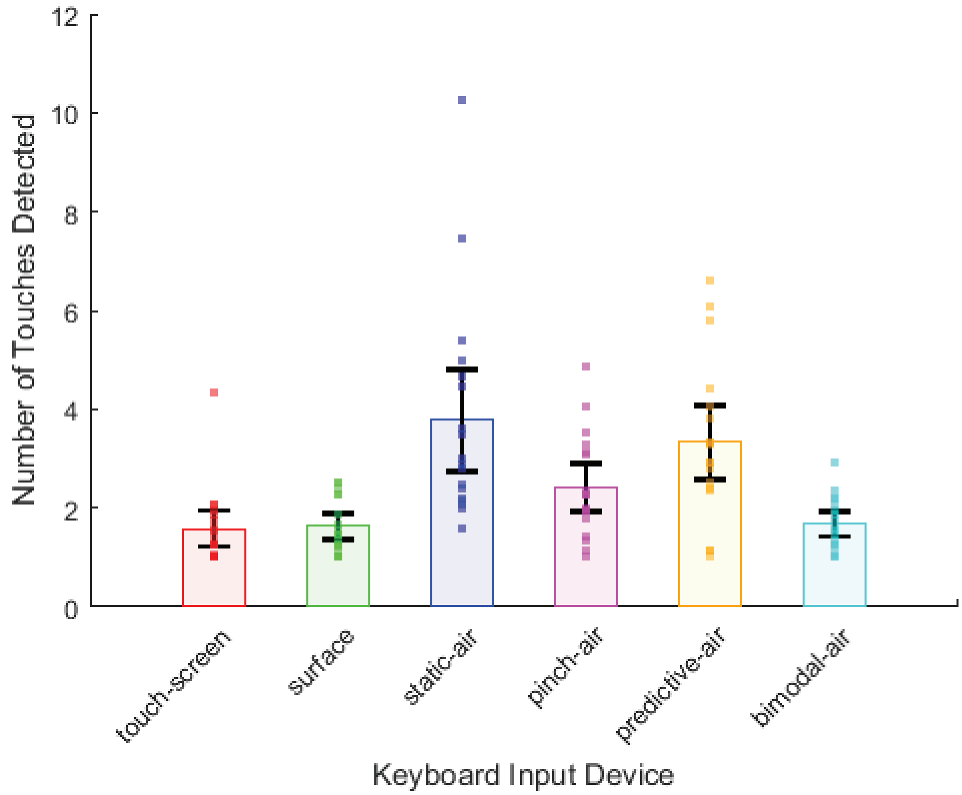


Figure 6.23. Mean Number of Touches Simulated for each keyboard with error bars showing 95% confidence intervals.

Finally, there was a significant difference (p-value = 0.0245) between the Pinch-air keyboard and the Static-air keyboard.

Again, the same trends as before were seen. The Leap Surface and Bimodal-air keyboards performed on par with the Touch Screen Keyboard, the Static-air and Predictive-air keyboards saw significantly more simulated touches, and the Pinch-air keyboard was somewhere in the middle. It was observed that the Static-air keyboard suffered from a "skimming" issue. The "skimming" issue occurred when participants would only reach far enough to barely simulate a touch on the interaction plane, and then when they would move their hand from one side of the keyboard to the other, the natural arcing motion of the participants' hands would cause them to pull away from the interaction plane as they moved. This issue was assumed to also be a culprit for some of the increased error rates. Figure 6.5 and Figure 7.1 both show how the skimming issue worked in detail.

6.6.2 *Number of Practice Words*

Participants reached a mean Number of Practice Words of 5.4 words ($SD =$ 4.3) for the Touch Screen Keyboard, 7.7 words ($SD = 3.9$) for the Leap Surface Keyboard, 9.0 words ($SD = 6.7$) for the Leap Static-air Keyboard, 8.1 words ($SD = 5.3$) for the Leap Pinch-air Keyboard, 8.9 words ($SD = 6.8$) for the Predictive-air Keyboard, and 10.2 words ($SD = 3.9$) for the Leap Bimodal-air Keyboard. Figure 6.24 shows the mean Number of Practice Words for each keyboard interaction style. Using a one-way ANOVA, no significant differences were detected between keyboard means ($F(5, 102) = 1.6828$, p-value $= 0.1454$, $SD_{pooled} = 5.3$), therefore no additional analysis was made. Typically, as observed by the researcher, there were two categories of participants when it came to performing practice words. The first category contained participants who would perform a full set of 15 words for each keyboard. Rarely, these participants felt the need to perform more than one practice set's worth of words. The second category contained participants who progressively performed less practice words for each consecutive keyboard; this is a primary example of why a Replicated Latin Squares design for counterbalancing was used.

6.7 *Qualitative Measures*

6.7.1 *Level of Discomfort*

Keyboards reached a median Level of Discomfort of "Strongly Disagree" for the Touch Screen Keyboard, "Strongly Disagree" for the Leap Surface Keyboard, "Neutral" for the Leap Static-air Keyboard, "Disagree" for the Leap Pinch-air Keyboard, between "Disagree" and "Neutral" for the Predictive-air Keyboard, and "Disagree" for the Leap Bimodal-air Keyboard. Figure 6.25 shows the median Level of Discomfort for each keyboard interaction style rated by the participants. Using a one-way

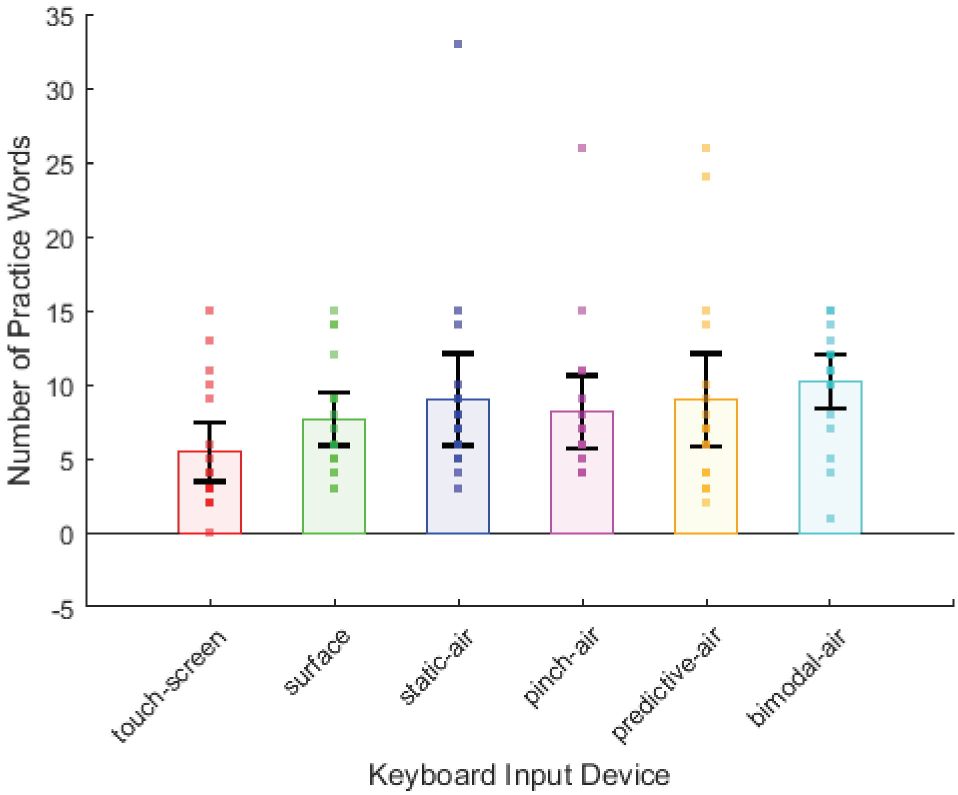


Figure 6.24. Mean Number of Practice Words for each keyboard with error bars showing 95% confidence intervals.

ANOVA, a significant difference was detected between keyboard means for Level of Discomfort ($F(5, 102) = 7.5000$, p-value < 0.0001, $SD_{pooled} = 0.94$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.23, revealed significant differences (p-values < 0.05) in Level of Discomfort between the Touch Screen Keyboard and the Leap Static-air, Pinch-air, and Predictive-air keyboards. There were significant differences (p-values < 0.01) between the Leap Surface Keyboard and the Static-air and Predictive-air keyboards. There was also a significant difference (p-value $=$ 0.0232) found between the Leap Bimodal-air and the Static-air keyboards with.

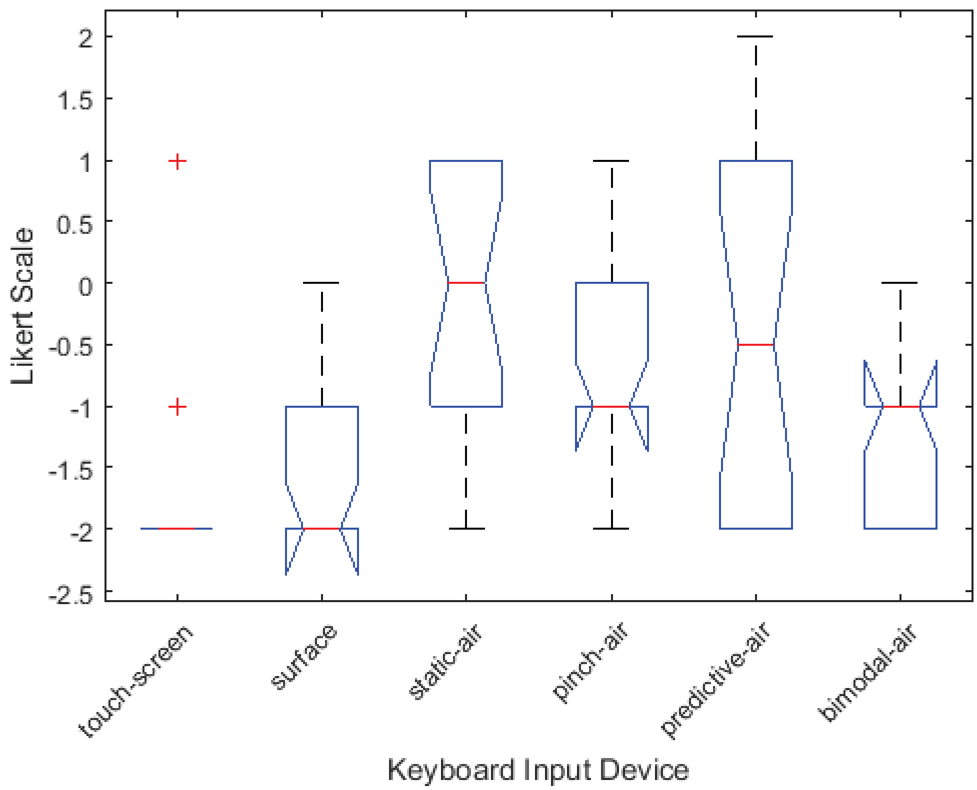


Figure 6.25. Median Level of Discomfort for each keyboard showing the 25th and 75th percentiles.

The level of discomfort experienced by participants, not surprisingly, was very similar to the trend seen in most other dependent measures. Discomfort was experienced more for the mid-air keyboards, and more extremely by the keyboards that utilized the extra degree of freedom available in the 3rd-dimension.

6.7.2 Level of Fatigue

Keyboards reached a median Level of Fatigue of "Strongly Disagree" for the Touch Screen Keyboard, "Strongly Disagree" for the Leap Surface Keyboard, "Agree" for the Leap Static-air Keyboard, between "Neutral" and "Agree" for the Leap Pinch-air Keyboard, "Neutral" for the Predictive-air Keyboard, and "Disagree" for the Leap Bimodal-air Keyboard. Figure 6.26 shows the median Level of Fatigue for each

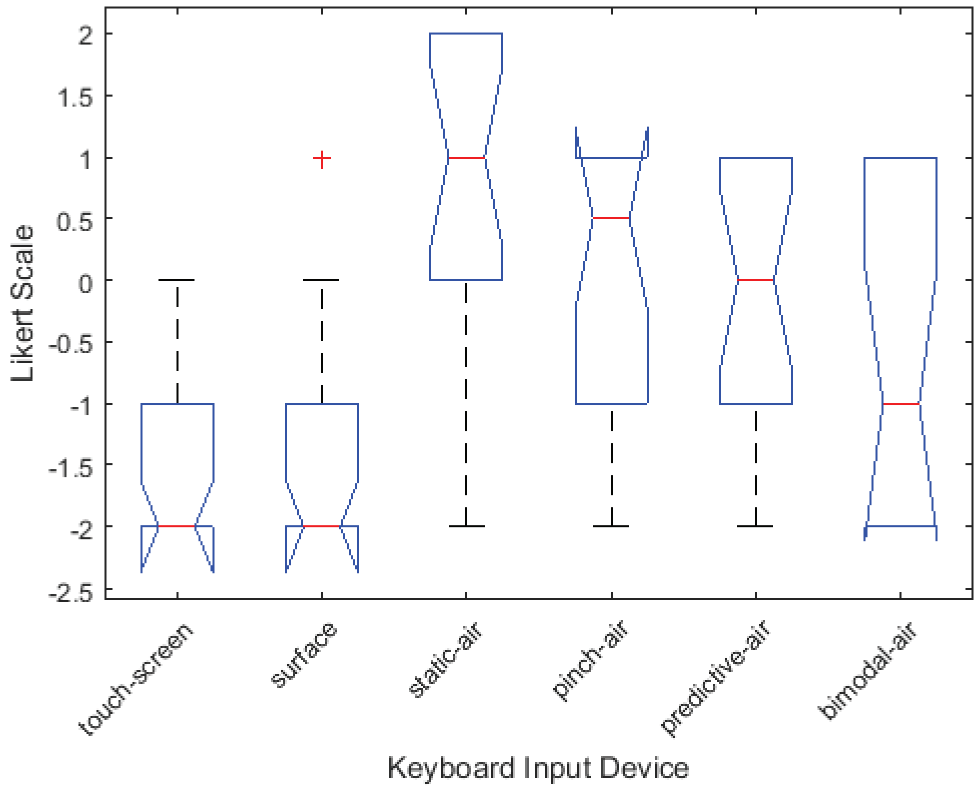


Figure 6.26. Median Level of Fatigue for each keyboard showing the 25th and 75th percentiles.

keyboard interaction style rated by the participants. Using a one-way ANOVA, a significant difference was detected between keyboard means for Level of Discomfort ($F(5, 102) = 10.9910$, p-value < 0.0001, $SD_{pooled} = 1.12$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.24, revealed significant differences (p-values < 0.01) in Level of Fatigue between the Touch Screen Keyboard and the Leap Static-air, Pinch-air, and Predictive-air keyboards. There were significant differences (p-values < 0.05) between the Leap Surface Keyboard and the Static-air, Pinch-air, and Predictive-air keyboards. There was also a significant difference (p-value $= 0.0043$) found between the Leap Bimodal-air and the Static-air keyboards. As expected, the fatigue levels experienced for mid-air keyboards were typically higher than those that were not mid-air.

6.7.3 Level of Difficulty

Keyboards reached a median Level of Difficulty of "Strongly Disagree" for the Touch Screen Keyboard, "Strongly Disagree" for the Leap Surface Keyboard, between "Disagree" and "Neutral" for the Leap Static-air Keyboard, "Disagree" for the Leap Pinch-air Keyboard, "Disagree" for the Predictive-air Keyboard, and "Disagree" for the Leap Bimodal-air Keyboard. Figure 6.27 shows the median Level of Difficulty for each keyboard interaction style rated by the participants. Using a one-way ANOVA, a significant difference was detected between keyboard means for Level of Discomfort ($F(5, 102) = 6.0351$, p-value < 0.0001, $SD_{pooled} = 0.98$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.25, revealed significant differences (p-values < 0.05) in Level of Difficulty between the Touch Screen Keyboard and the Leap Static-air, Pinch-air, and Predictive-air keyboards. There was significant difference (p-values $= 0.0042$) between the Leap Surface and the Static-air keyboards. There was also a significant difference (p-value $= 0.0209$) found between the Leap Bimodal-air and the Static-air keyboards. As anticipated, the mid-air keyboards had a higher level of difficulty and were harder to use or understand than the non-mid-air keyboards.

6.7.4 Preference Ranking

Participants reached a mean Preference Ranking of 1.89 ($SD = 1.18$) for the Touch Screen Keyboard, 2.22 ($SD = 1.22$) for the Leap Surface Keyboard, 5.33 ($SD = 0.84$) for the Leap Static-air Keyboard, 4.22 ($SD = 1.06$) for the Leap Pinch-air Keyboard, 4.50 ($SD = 1.42$) for the Predictive-air Keyboard, and 2.83 ($SD = 1.29$) for the Leap Bimodal-air Keyboard. Figure 6.28 shows the mean Preference

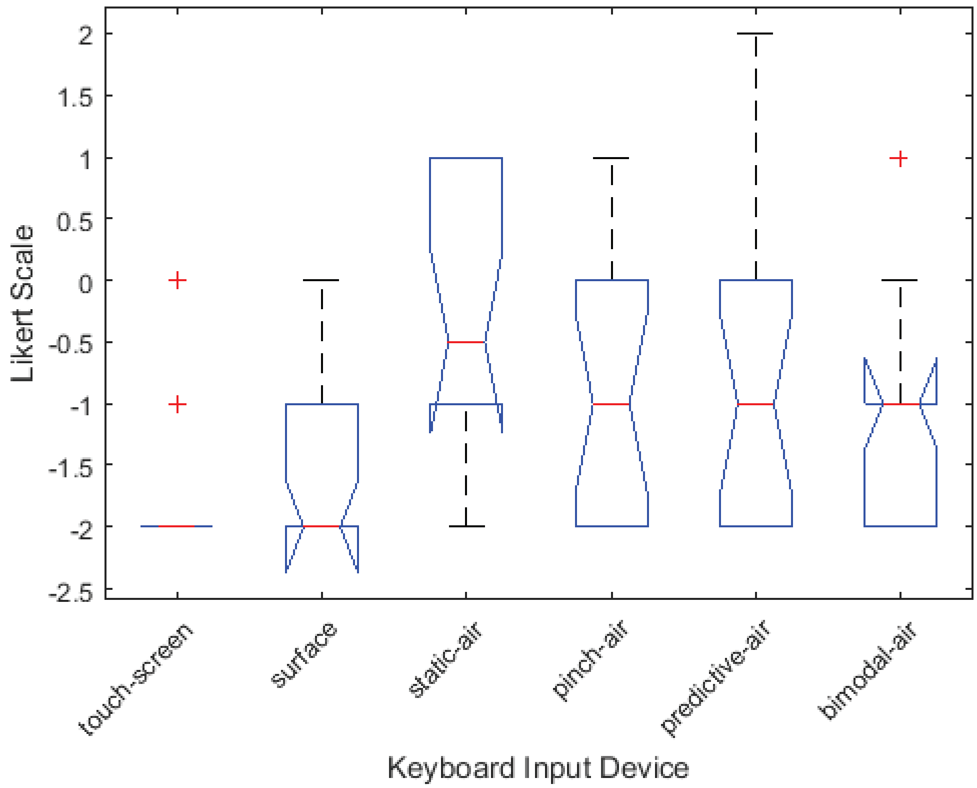


Figure 6.27. Median Level of Difficulty for each keyboard showing the 25th and 75th percentiles.

Ranking for each keyboard interaction style. Using Friedman's rank-test, ANOVA, a significant difference was detected between keyboard means, $\chi^2(5, 85) = 49.1429$, p-value < 0.0001, $SD_{pooled} = 1.87$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table D.26, revealed significant differences (p-values < 0.01) in Preference Ranking between the Touch Screen Keyboard and the Leap Static-air, Pinch-air, and Predictive-air keyboards. There were significant differences (p-values < 0.05) between the Leap Surface Keyboard and the Static-air, Pinch-air, and Predictive-air keyboards. There was also a significant difference (p-value $= 0.0009$) found between the Leap Bimodal-air and the Static-air keyboards.

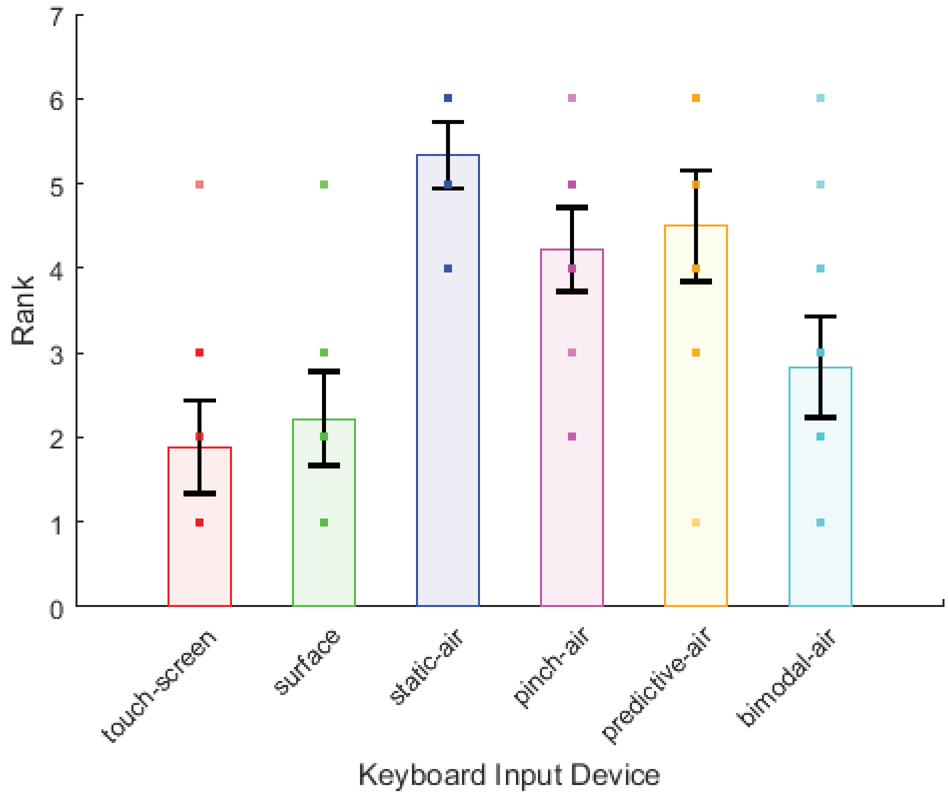


Figure 6.28. Mean Preference Rankings for each keyboard with error bars showing 95% confidence intervals.

The preference ranking of all of the keyboards very strongly reflected the trends seen in most of the other dependent measures. The Touch Screen Keyboard was the most preferred, as expected, followed by the Leap Surface in a close second. The Leap Bimodal-air was the most preferred mid-air keyboard, whereas the Static-air Keyboard was the least preferred.

CHAPTER SEVEN

Discussion, Future Work, and Conclusion

7.1 Discussion

Vulture has shown that word-gesture keyboards are beneficial to mid-air text-entry by surpassing earlier work and reaching a text-entry rate of 28.1 WPM. However, these keyboards utilized pinching as a means of word separation (Markussen, Jakobsen, and Hornbæk 2014). This thesis aimed to find alternative solutions to word separation and interaction with mid-air, word-gesture keyboards. To seek improvements superior to pinching, this thesis focused on the 3rd-dimension in addition to implementing a bimodal approach. Inconvenient glove-use was replaced with a natural, barehanded gestural interaction.

This thesis suffered from some of the same issues that were experienced in previous work, such as text-entry rates being much slower for mid-air keyboards than touch-based. One reason for this was that participants were required to recouple the gestures in motor space with those on the display (Markussen, Jakobsen, and Hornbæk 2014). This effect was described in detail for many of the results in Chapter 6. Another reason, as seen in Vulture and in other studies (Markussen, Jakobsen, and Hornbæk 2014, Witt, Lawo, and Drugge 2008), was that participants relied heavily on displayed feedback. This was seen in the small amount of latency introduced by the Leap Motion controller when tracking and displaying hand motions on the screen. Some participants reported having to slow hand movement to ensure a more accurate display representation. If movements were made too quickly, the visual display would lag slightly, causing participants to "overshoot" characters. A final similarity to Vulture was that the keyboard alternatives forced explicit delimiting of words, adding to the complexity of text-entry.

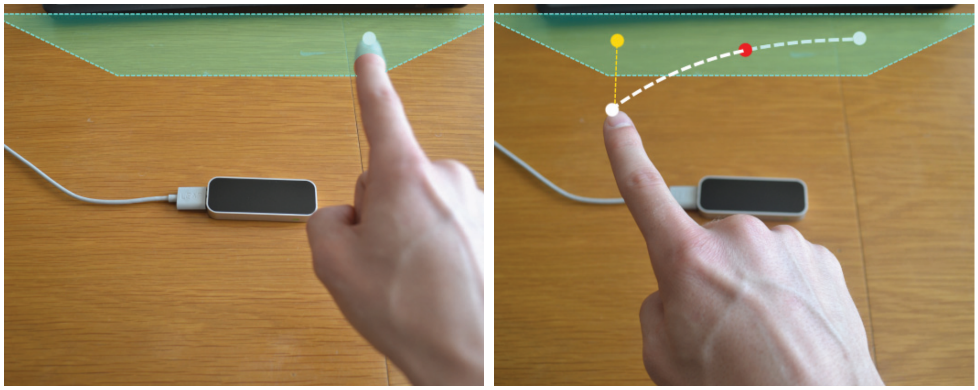
7.1.1 Pinching Interaction

This thesis implemented a pinching method using the Leap Motion controller as a metric for comparison against Vulture's pinching method. This was done to compare the pseudo-implementation of word-gesturing versus Vulture's traditional word-gesture keyboard applied to mid-air. The text-entry rate for pinching with a single session was 11.3 WPM, which was consistent with the mean text-entry rate using Vulture ($M = 11.8$ WPM) for a single session (Markussen, Jakobsen, and Hornbæk 2014). Additionally, pinching was 58% of the text-entry rate of direct touch input, which was proportional to Vulture at 59% of the text-entry rate of direct touch input (Markussen, Jakobsen, and Hornbæk 2014). These consistent measures provided a baseline for this thesis's mid-air and bimodal keyboard alternatives.

7.1.2 3-dimensional Interaction

This thesis showed initial text-entry rates for utilizing the 3rd-dimension as a means of word separation at 8.6 WPM for the Static-air approach and 9.6 WPM for the Predictive-air approach. These approaches underperformed because of difficulty in mentally coupling word-gestures in motor space with those displayed on the screen, as well as having extra degrees of freedom in the 3rd-dimension when delimiting words via direct interaction with an invisible plane.

The Predictive-air Keyboard was not significantly worse than the Pinch-air Keyboard and warrants further investigation with a repeated-measures, multiple session study. Both the Static-air and Predictive-air suffered from an issue referred to as "skimming," demonstrated in Figure 6.5 and Figure 7.1. The "skimming" issue occurred when the natural arcing of the participant's hand during the gesturing motion caused them to lose the simulated touch. This "skimming" occurred both during word-gestures and when removing the hand from the plane after having completed a word-gesture. This was more prevalent if a participant's elbow was rested.



(a) Pressing Key

(b) Moving to next Letter across Keyboard

Figure 7.1. Example of how the natural arcing motion of the arm loses touch as moving across the interaction plane. (a) shows the user pressing a key; and (b) shows the intended destination in *yellow* and the accidental release in *red*.

The Leap Surface Keyboard utilized the Static-air implementation to project a mid-air keyboard onto flat surfaces, simulating a touch screen. It's interesting to note that the Leap Surface was essentially indistinguishable from the Touch Screen Keyboard in terms of performance and text-entry rate, reaching 17.1 WPM for a single session. The Leap Surface Keyboard handily proves how essential it is to avoid decoupling the motor space and display space. This implies that the Static-air Keyboard may substantially benefit from an input plane displayed using augmented reality, re-coupling the motor space and display space.

7.1.3 Bimodal Interaction

Bimodal mid-air interactions were shown to be very promising in this thesis and it should be noted that any other secondary input could be used as the source for touch simulation. The Bimodal-air Keyboard reached a text-entry rate of 15.8 WPM for a single session, which was a significant improvement over using pinching or utilizing the 3rd-dimension. The Bimodal-air Keyboard reached 81% of the text-entry rate of direct touch input and was also often indistinguishable from the Touch Screen Keyboard for many other dependent measures, as detailed in Chapter 6. The Bimodal-air keyboard warrants further investigation with a repeated-measures, multiple session study on a traditional word-gesture keyboard implemented with word-recognition.

The Bimodal-air was not without some of its own drawbacks. As with all other mid-air keyboards implemented with the Leap Motion controller, there was an associated input plane that word-gestures were projected onto as the participants moved their hands. This plane, if calibrated or oriented incorrectly, could lead to higher error rates and less precision overall, as seen in Section 7.2.6.

7.1.4 Lacking Word-recognition

There were some glaring limitations to using a pseudo-implementation of word-gesturing as opposed to using a full word-recognition implementation. A major limitation was that the pseudo-implementation analyzed gestures as they were being created, allowing participants to see real-time updates to path and character production. The software attempted to analyze the current character being pressed based on the deviations in the gesture-path being created and then compared those deviations with the current expected characters. The consequence of this limitation was that participants often interrupted the gesturing process in favor of correcting the erroneous characters, which is absent in traditional word-gesture keyboards that utilize word-recognition. An additional limitation to not analyzing a gesture-shape against a known compendium of common words is the software's increased sensitivity to detecting errors while gesturing, which caused more frequent interruptions of gesture-shapes. This limitation also meant that non-words (e.g., words not in any dictionary) could be produced. Another hindrance was that once an error was made, the likelihood of detecting more errors increased because the path protection would be disabled.

Due to the differences and trade-offs between pseudo-implementation and traditional implementation, it was expected that text-entry rates might have been different than those produced by a traditional word-gesture keyboard applied in mid-air.

However, this was not the case. Text-entry rates for a single session with no training were consistent with those produced in Vulture. The Pinch-air keyboard reached a mean text-entry rate of 11.3 WPM as opposed to the mean text-entry rate of 11.8 WPM seen in Vulture (Markussen, Jakobsen, and Hornbæk 2014). This is important because the Pinch-air keyboard was designed to mimic the Vulture technique for word separation.

7.1.5 Enforcing Error Correction

Error correction was enforced due to the higher level of sensitivity to producing errors for the pseudo-implemented word-gesture keyboards. Once one error had been produced, the path protection described in Section 3.1.2 would be disabled since it could no longer be known what the next action of the participant would be. By enforcing correction, the goal was to ensure that the software could always know the next expected key "press." However, requiring the participants to correct erroneous transcriptions presented its own limitations. By requiring error correction, it was much more difficult to track and interpret results for error rates. It is important to note that because participants were required to correct the erroneous transcriptions, the text-entry rates reported in Section 6.1 were achieved with a 0% error rate for Vulture's Minimum Word Distance formula (Markussen, Jakobsen, and Hornbæk 2014).

7.2 Future Work

7.2.1 Word-recognition

A future path of this work should use a traditional word-gesture keyboard implementation with word-recognition. This single change will give improved and more standardized results for all measures. Traditional word-recognition would eliminate this thesis's need for enforced error correction, which resulted in modified variables and character-level error production. Because participants would not be distracted with error production mid-gesture, an increase in text-entry rates is expected.

7.2.2 Task Redesign

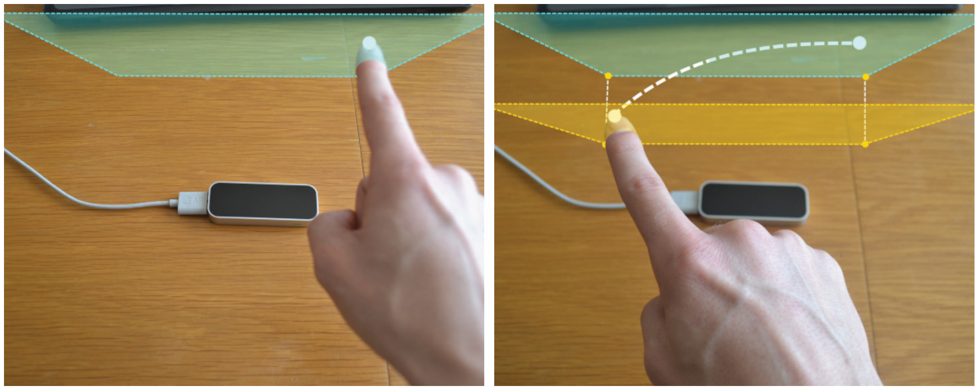
The task needs to be substantially redesigned to fit a word-recognition implementation of the word-gesture keyboards. First, it would be beneficial to build phrases containing similar gesture-shapes instead of single words to better track text-entry and error rates. Next, there should not be any requirement to correct erroneous words; it should feel natural for the participant and they should fix errors where they feel it is necessary. Finally, the number of trials needs to be increased by a significant amount and extended for repeated sessions so that the true potential of the keyboards' (especially the Bimodal-air's) text-entry rates can analyzed. If implemented on a traditional word-gesture keyboard in mid-air, with repeated sessions and training, the Bimodal-air keyboard is expected to achieve text-entry rates superior to those found in Vulture.

7.2.3 Augmented Reality

Augmented reality would decrease the mental coupling between the gesture motor space and the keyboard display space, and therefore would be a huge boon for 3-dimensional mid-air implementations. This is theorized to benefit the Static-air Keyboard most due to the success of the Leap Surface Keyboard, which is simply the Static-air keyboard projected onto a surface. Participants being able to directly see the mid-air keyboard they are using would be expected to substantially increase text-entry rates.

7.2.4 The "Skimming" Issue

The simplest way to solve the "skimming" issue is to increase the interaction plane's release threshold. As shown in Figure 7.2, once a simulated touch has been made, increasing the release threshold toward the participant so that the participant's



(a) Intersecting Plane to hit First Key (b) Increased Release Threshold

Figure 7.2. An example of how the "skimming" issue could be fixed by increasing the release threshold after touching. (a) shows the user pressing the first key; and (b) shows the increased release threshold for the interaction plane as the user moves across to the next key.

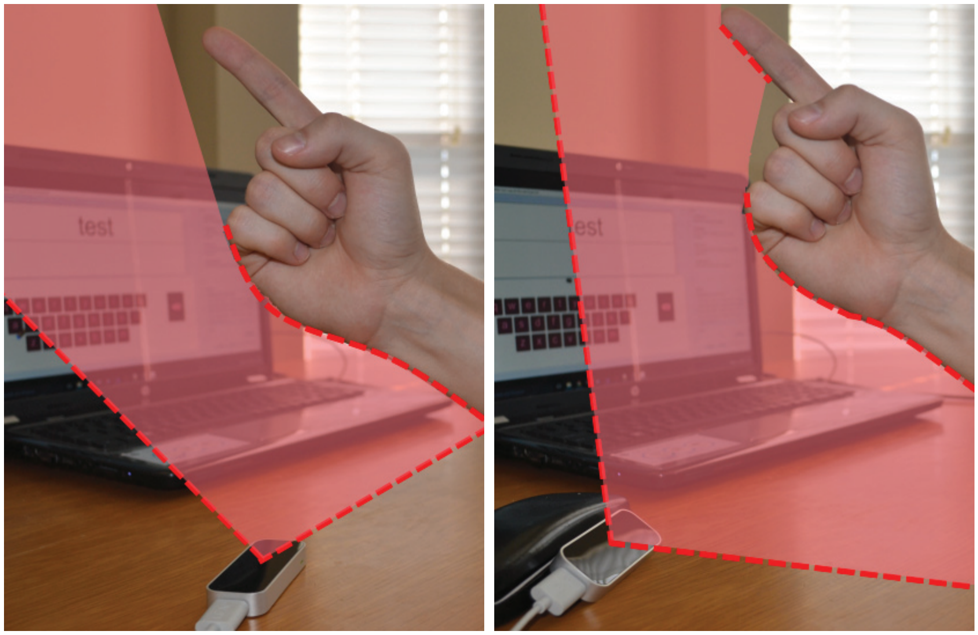
hand does not leave the interaction plane when moving on the x and y axes. This change is expected to reduce accidental releases and improve results for the Static-air Keyboard. A similar approach could also be used for the Predictive-air Keyboard because quick side-to-side movements were sometimes seen as a touch being released.

7.2.5 Improved Tracking

During the study, the Leap Motion controller often had issues detecting participants' pointer fingers or palms. A major factor of these detection issues was due to the positioning of the Leap Motion controller itself, as well as some participants holding their hand in an upward position, as shown in Figure 7.3a. A simple solution to this issue is to angle the Leap Motion controller so that it is facing the participant at an angle providing a wider range of view, as seen in Figure 7.3b.

7.2.6 Alternative Interaction Plane

As participants moved across the interaction plane in mid-air, sometimes the side-to-side movements and up and down movements were not represented as expected, as seen in Figure 7.4. Better approaches to calibration and plane-creation,



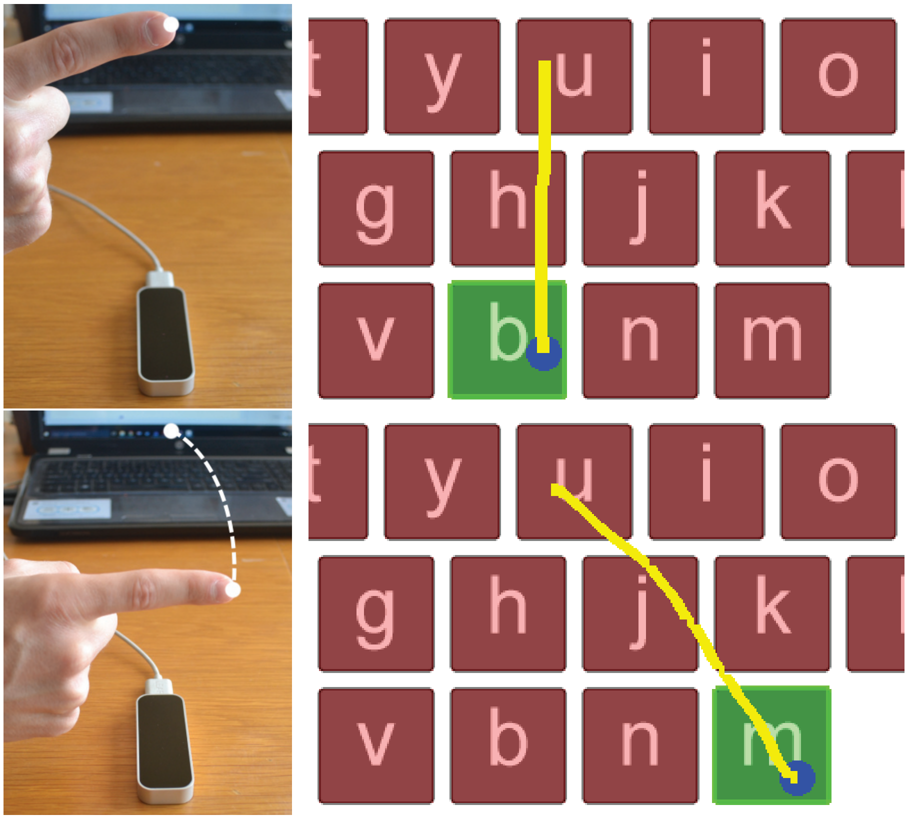
(a) Hand Blocking Finger (b) Angle Reveals Finger

Figure 7.3. Examples of how some participants blocked their finger from being tracked. (a) shows the user blocking the view of their finger with their hand; (b) shows how an angled Leap Motion controller might solve the problem.

such as those discussed in Personal Space and elsewhere (Hari Haran 2014, Guinness 2015), need to be implemented. A spherical or even curved interaction plane designed for each individual participant would greatly benefit all of the mid-air keyboards especially so that participants would be able to rest their arms (Guinness 2015). With a flat, quadrilateral plane, many participants suffered the same issues as mentioned in (Hari Haran 2014), especially when moving to the extreme edges of the projected keyboard.

7.2.7 Multi-functional Space

Since mid-air word-gesturing keyboards utilize the 3rd-dimension or implement a bimodal approach, hand gestures could be used for alternative interactions when not simulating a touch (e.g., control a cursor, make a selection, etc.). This would



(a) Movement (b) Intended Path (*Top*) vs. Actual Path (*Bottom*)

Figure 7.4. Example showing how a bad calibration affects movement. (a) shows a user trying to make a vertical movement; and (b) shows how the actual path was misaligned because of the calibration and orientation of the Leap Motion controller.

allow users to fully interact with screens of any kind, including projections, and not limit the user to a keyboard and mouse.

7.2.8 Image Processing

The Leap Surface Keyboard's projected virtual keyboard revealed an area for possible expansion. While the hard-coded software keyboard functioned adequately, image processing could make the feature much more versatile. Being able to print out and use any keyboard for word gesturing or otherwise could be highly beneficial.

7.2.9 Gaming Console Keyboards

Using the Xbox Controller Keyboard in the pre-pilot and the pilot study confirmed the inconvenience of single-input text-entry. Therefore, it would be beneficial to apply what has been learned from this thesis (and word-gesturing in general) to gaming console keyboards. Gaming consoles could implement mid-air word-gesture keyboards using the Xbox Kinect, PlayStation Move, or using Wii Remotes. Alternatively, standard console keyboards could transition to a word-gesture keyboard using standard console controllers. The word-gesture keyboard would most likely have to be bimodal. The user would hold a button, simulating touch, and use a thumb stick to move a cursor around for word-gesturing. Either method would be an improvement over single character text-entry currently seen on modern gaming consoles.

7.2.10 Accessibility

A major motivation for this thesis was the possible application of mid-air word-gesture keyboards for amputees or those with disabilities that affect performance using a standard keyboard. A proper study in accessibility should be performed to utilize the Bimodal-air keyboard for users with disabilities.

7.3 Conclusion

Word-gesture keyboards provide the means to more efficient mid-air text-entry as compared to the slower, single-input text-entry that had been previously used (Markussen, Jakobsen, and Hornbæk 2014). This thesis demonstrated alternative ways to delimit the separation of words for word-gesture keyboards applied to mid-air. It was shown that utilizing the 3rd-dimension as a means of word separation is too complex to be beneficial when paired with a decoupled gesture-space and display space. As demonstrated by projecting a Static-air plane onto a surface with the Leap Surface Keyboard, better results may be achieved when using the 3rd-dimension as a means of word-separation if they are complemented by an augmented reality display to recouple the gesture-space and display space. Another method to reduce the effects of a decoupled gesture-space and display space is to use new techniques to better calibrate the interaction plane for each individual user (Hari Haran 2014, Guinness 2015). Finally, the empirical results from this thesis showed that using a bimodal technique as a means of word separation would greatly benefit word-gesture keyboards in mid-air. Text-entry rates for the Bimodal-air Keyboard ($M = 15.8$ WPM) reached 81% of those observed for direct touch input. This was a 37% improvement to Vulture, which reached only 59% of the text-entry rate of direct touch input (Markussen, Jakobsen, and Hornbæk 2014). In addition, the bimodal method was nearly indistinguishable from the touch screen for almost all other dependent measures and was better than pinching for nearly all measures. This thesis necessitates a follow-up study with re-designed trials using a traditional word-gesture keyboard implementation utilizing repeated measures and reoccurring sessions to further investigate bimodal techniques for mid-air text-entry. With further research, a bimodal approach shows the potential to be the future of mid-air text-entry for word-gesture keyboards.

APPENDICES

APPENDIX A

Dictionary Word Sets

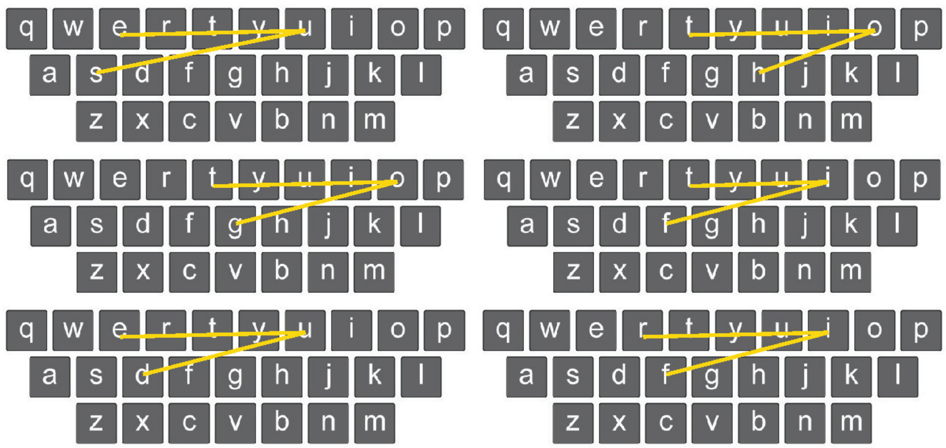


Figure A.1. The similar gesture-shapes of the word set: ‘sue’, ‘hot’, ‘got’, ‘fit’, ‘due’, ‘fir’. These gesture-shapes were generated by minimizing dissimilarity using the custom dissimilarity algorithm in Section 3.1.5.2.

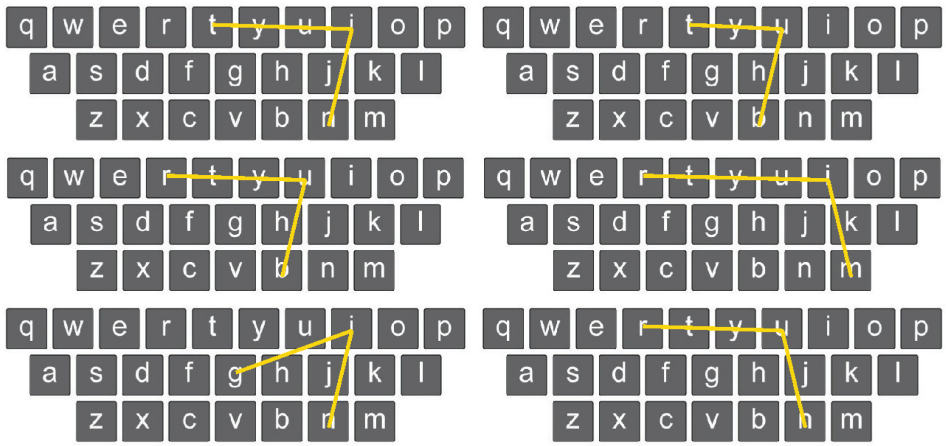


Figure A.2. The similar gesture-shapes of the word set: 'tin', 'tub', 'rub', 'rim', 'gin', 'run'. These gesture-shapes were generated by minimizing dissimilarity using the custom dissimilarity algorithm in Section 3.1.5.2.

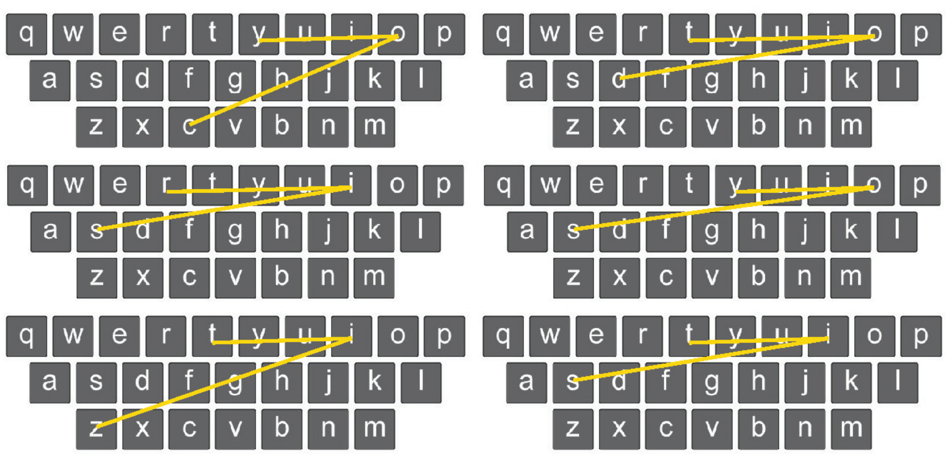


Figure A.3. The similar gesture-shapes of the word set: 'coy', 'dot', 'sir', 'soy', 'zit', 'sit'. These gesture-shapes were generated by minimizing dissimilarity using the custom dissimilarity algorithm in Section 3.1.5.2.

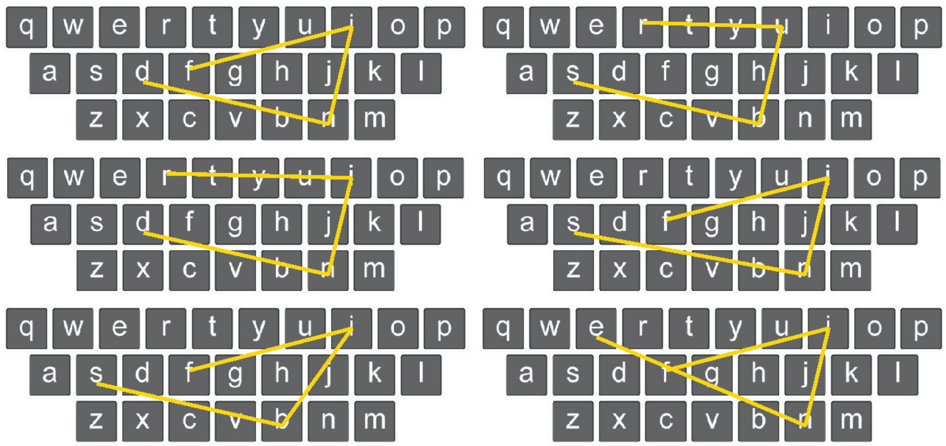


Figure A.4. The similar gesture-shapes of the word set: ‘find’, ‘rubs’, ‘rind’, ‘fins’, ‘fibs’, ‘fine’. These gesture-shapes were generated by minimizing dissimilarity using the custom dissimilarity algorithm in Section 3.1.5.2.

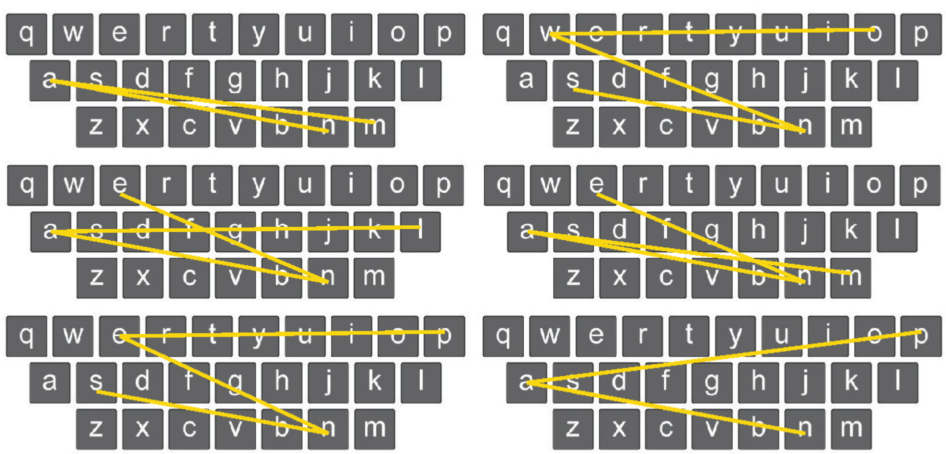


Figure A.5. The similar gesture-shapes of the word set: ‘mans’, ‘owns’, ‘lane’, ‘mane’, ‘pens’, ‘pans’. These gesture-shapes were generated by minimizing dissimilarity using the custom dissimilarity algorithm in Section 3.1.5.2.

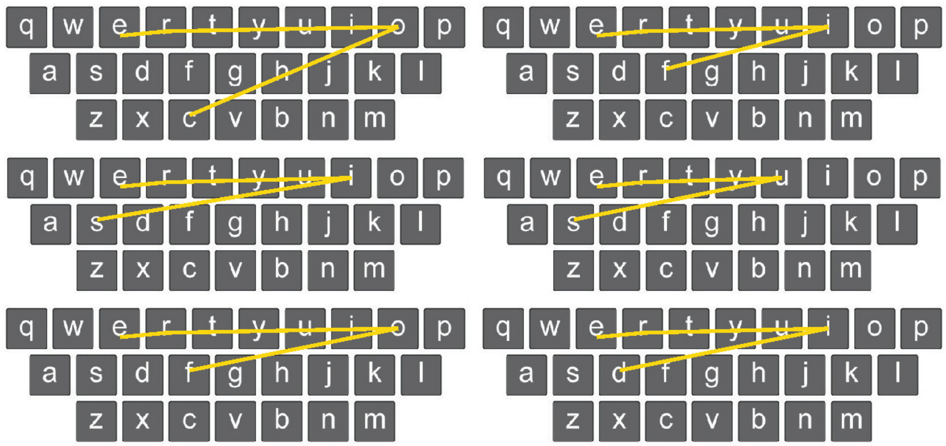


Figure A.6. The similar gesture-shapes of the word set: 'core', 'fire', 'sire', 'sure', 'fore', 'dire'. These gesture-shapes were generated by minimizing dissimilarity using the custom dissimilarity algorithm in Section 3.1.5.2.

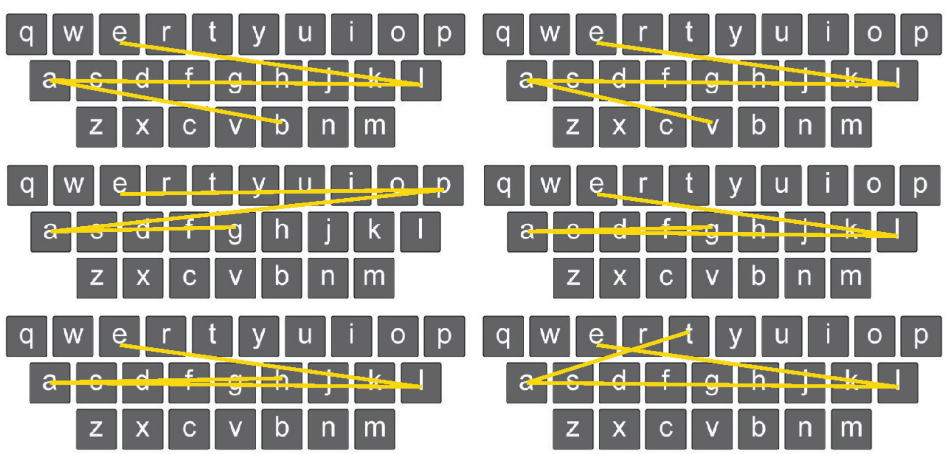


Figure A.7. The similar gesture-shapes of the word set: 'bale', 'vale', 'gape', 'gale', 'hale', 'tale'. These gesture-shapes were generated by minimizing dissimilarity using the custom dissimilarity algorithm in Section 3.1.5.2.

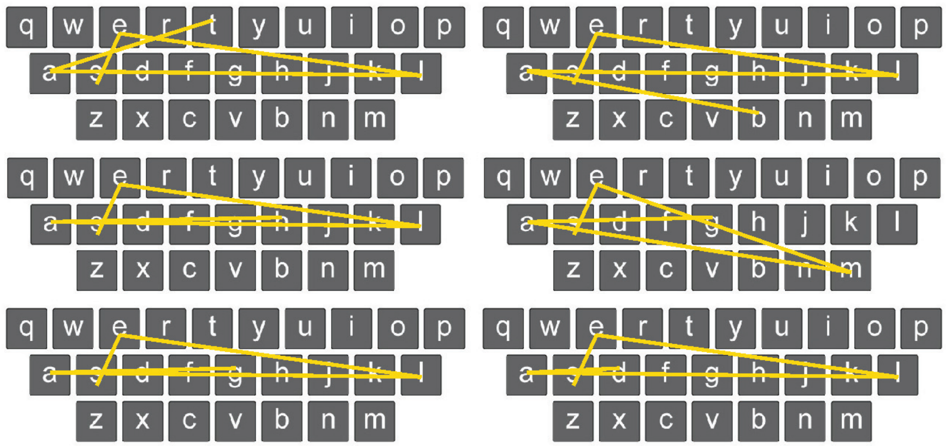


Figure A.8. The similar gesture-shapes of the word set: ‘tales’, ‘bales’, ‘hales’, ‘games’, ‘gales’, ‘dales’. These gesture-shapes were generated by minimizing dissimilarity using the custom dissimilarity algorithm in Section 3.1.5.2.

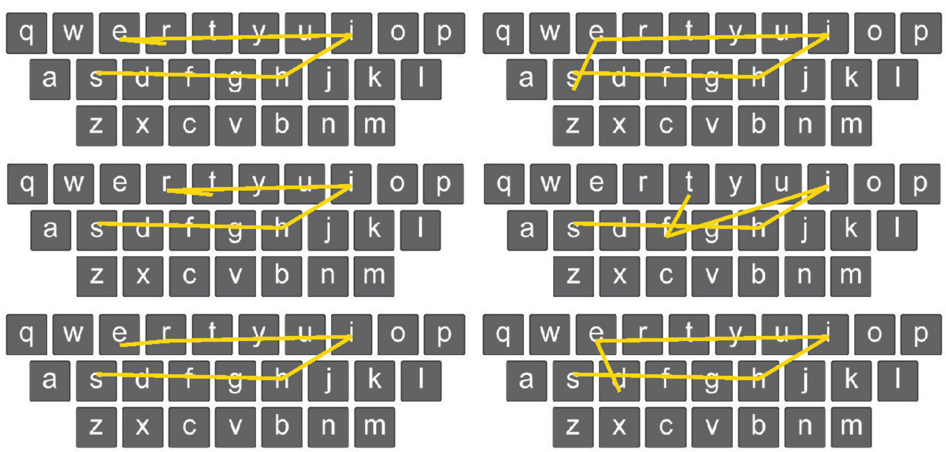


Figure A.9. The similar gesture-shapes of the word set: ‘shier’, ‘shies’, ‘shirt’, ‘shift’, ‘shire’, ‘shied’. These gesture-shapes were generated by minimizing dissimilarity using the custom dissimilarity algorithm in Section 3.1.5.2.

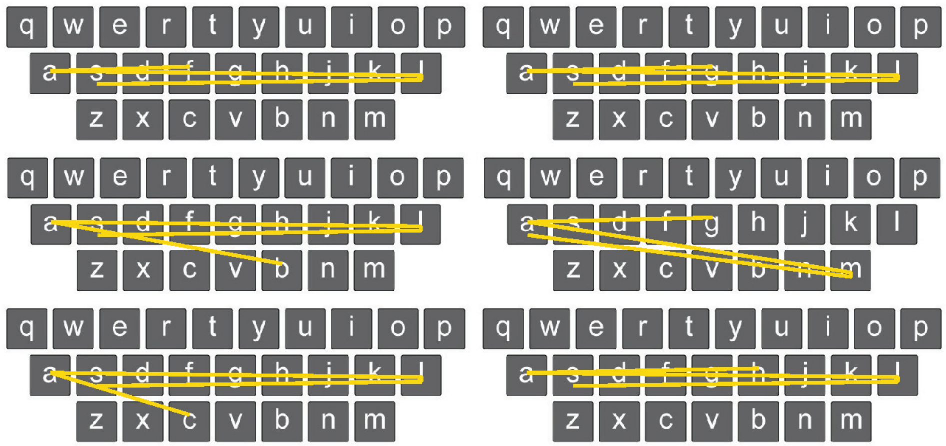


Figure A.10. The similar gesture-shapes of the word set: ‘falls’, ‘galls’, ‘balls’, ‘gamma’, ‘calls’, ‘halls’. These gesture-shapes were generated by minimizing dissimilarity using the custom dissimilarity algorithm in Section 3.1.5.2.

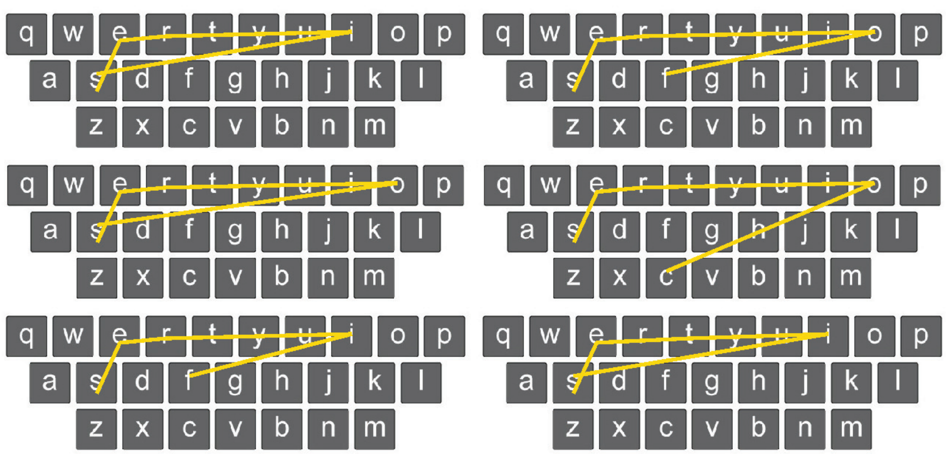


Figure A.11. The similar gesture-shapes of the word set: ‘sires’, ‘fores’, ‘sores’, ‘cores’, ‘fires’, ‘sites’. These gesture-shapes were generated by minimizing dissimilarity using the custom dissimilarity algorithm in Section 3.1.5.2.

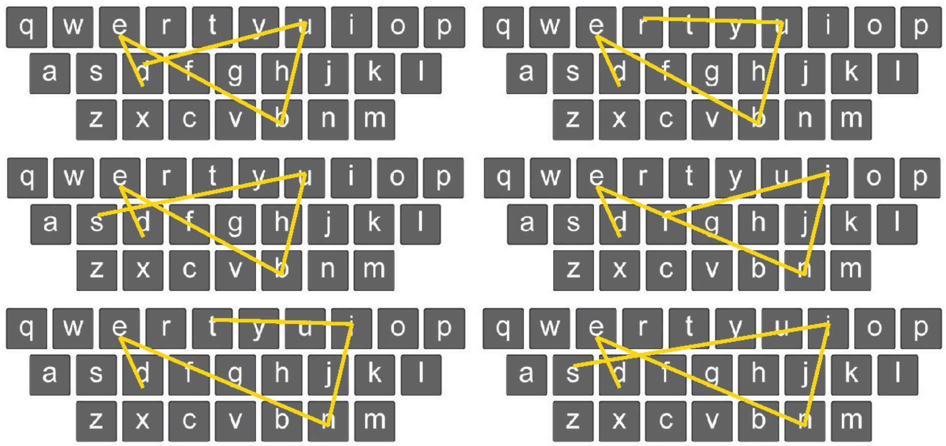


Figure A.12. The similar gesture-shapes of the word set: ‘dubbed’, ‘rubbed’, ‘subbed’, ‘finned’, ‘tinned’, ‘sinned’. These gesture-shapes were generated by minimizing dissimilarity using the custom dissimilarity algorithm in Section 3.1.5.2.

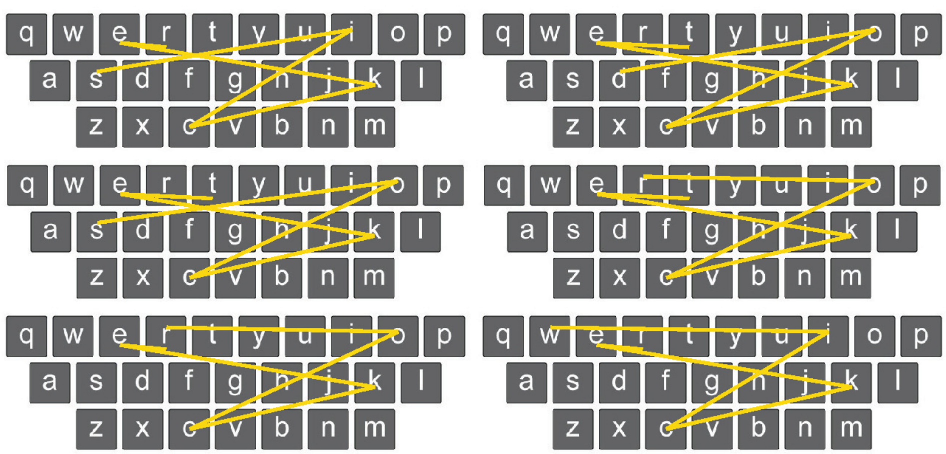


Figure A.13. The similar gesture-shapes of the word set: ‘sicker’, ‘docket’, ‘socket’, ‘rocket’, ‘rocker’, ‘wicker’. These gesture-shapes were generated by minimizing dissimilarity using the custom dissimilarity algorithm in Section 3.1.5.2.

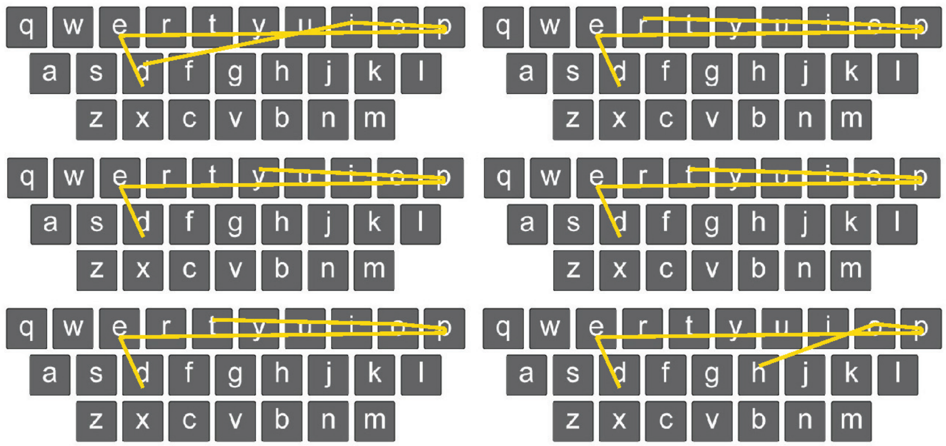


Figure A.14. The similar gesture-shapes of the word set: ‘dipped’, ‘ripped’, ‘yipped’, ‘tipped’, ‘topped’, ‘hopped’. These gesture-shapes were generated by minimizing dissimilarity using the custom dissimilarity algorithm in Section 3.1.5.2.

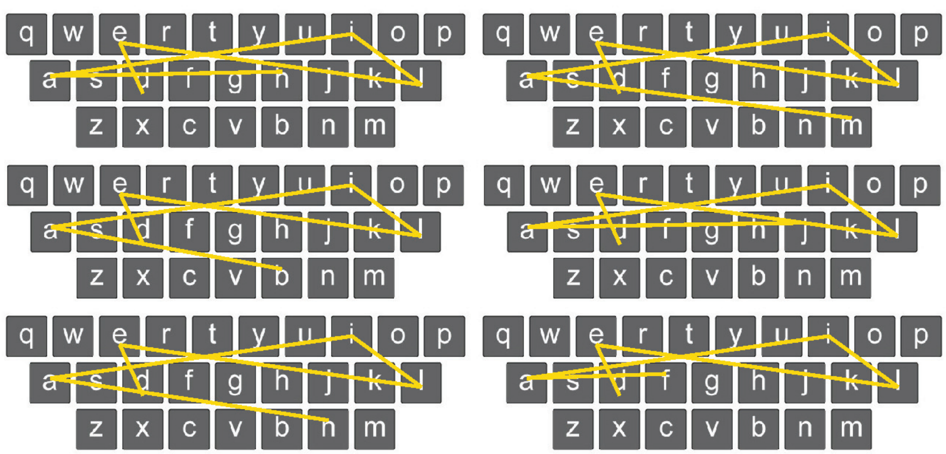


Figure A.15. The similar gesture-shapes of the word set: ‘hailed’, ‘mailed’, ‘bailed’, ‘jailed’, ‘nailed’, ‘failed’. These gesture-shapes were generated by minimizing dissimilarity using the custom dissimilarity algorithm in Section 3.1.5.2.

APPENDIX B

Surveys

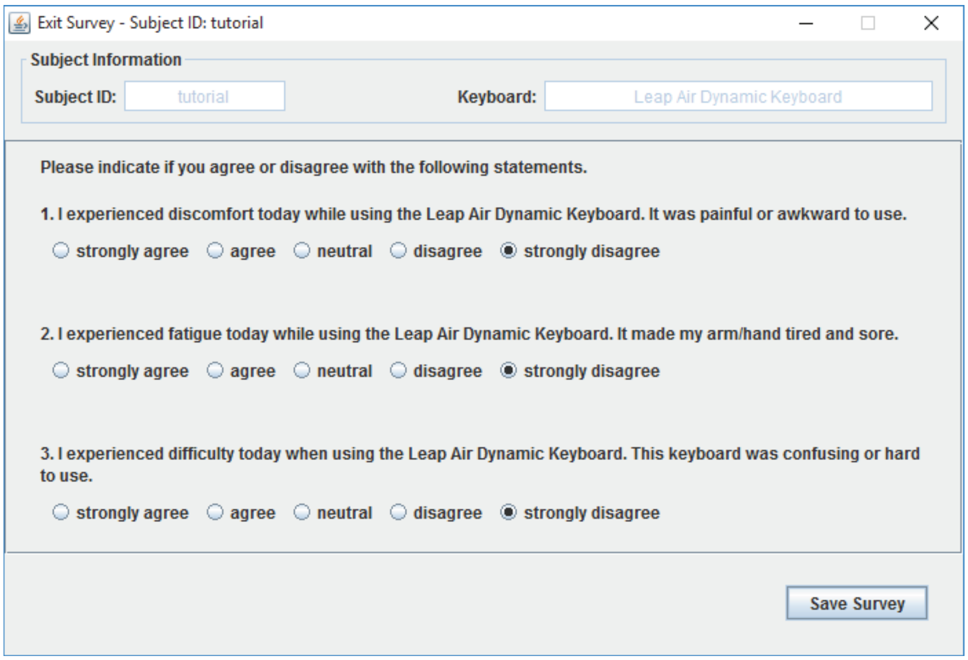
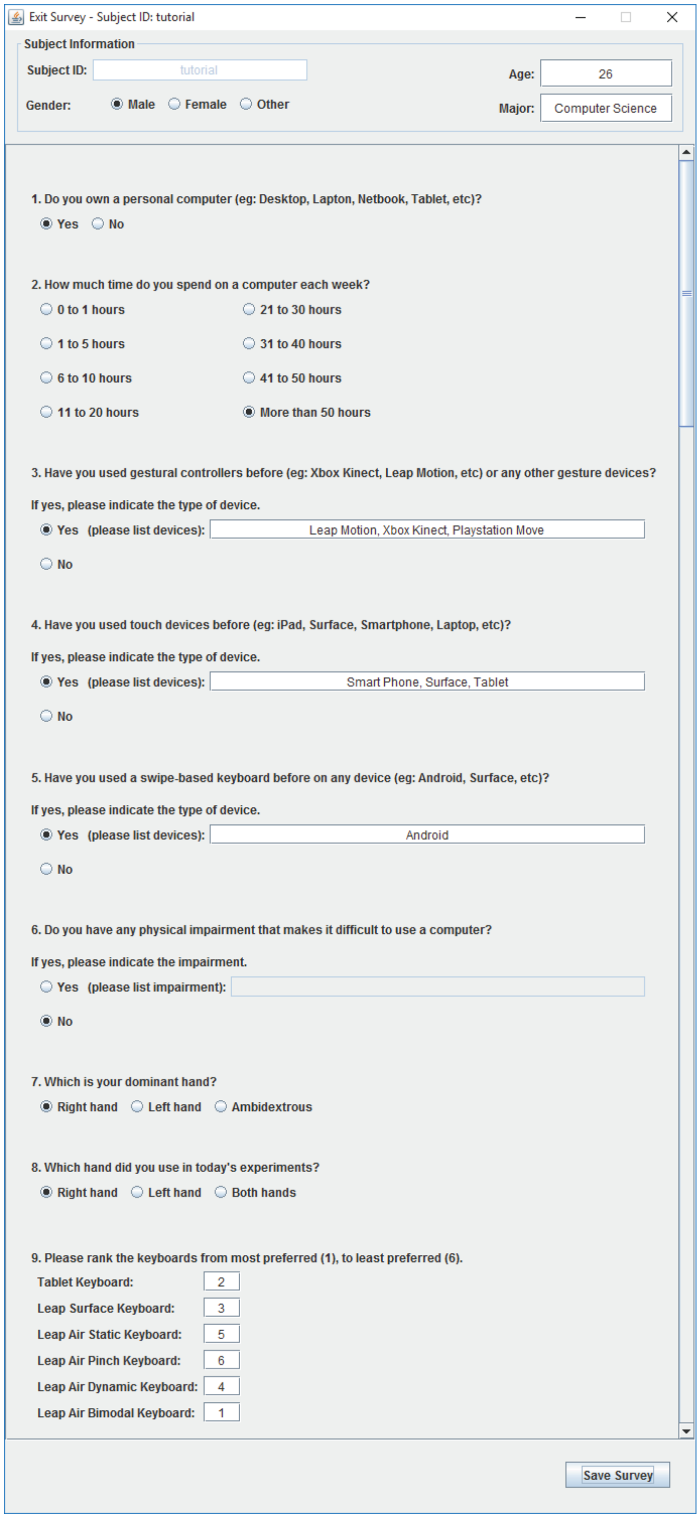


Figure B.1. Example of an intermittent survey that participants were required to fill out after finishing a task with a single keyboard.



Exit Survey - Subject ID: tutorial

Subject Information

Subject ID: tutorial

Age: 26

Gender: (•) Male ( ) Female ( ) Other

Major: Computer Science

1. Do you own a personal computer (eg: Desktop, Lapton, Netbook, Tablet, etc)?

(•) Yes ( ) No

2. How much time do you spend on a computer each week?

( ) 0 to 1 hours ( ) 21 to 30 hours

( ) 1 to 5 hours ( ) 31 to 40 hours

( ) 6 to 10 hours ( ) 41 to 50 hours

( ) 11 to 20 hours (•) More than 50 hours

3. Have you used gestural controllers before (eg: Xbox Kinect, Leap Motion, etc) or any other gesture devices?

If yes, please indicate the type of device.

(•) Yes (please list devices): Leap Motion, Xbox Kinect, Playstation Move

( ) No

4. Have you used touch devices before (eg: iPad, Surface, Smartphone, Laptop, etc)?

If yes, please indicate the type of device.

(•) Yes (please list devices): Smart Phone, Surface, Tablet

( ) No

5. Have you used a swipe-based keyboard before on any device (eg: Android, Surface, etc)?

If yes, please indicate the type of device.

(•) Yes (please list devices): Android

( ) No

6. Do you have any physical impairment that makes it difficult to use a computer?

If yes, please indicate the impairment.

( ) Yes (please list impairment):

(•) No

7. Which is your dominant hand?

(•) Right hand ( ) Left hand ( ) Ambidextrous

8. Which hand did you use in today's experiments?

(•) Right hand ( ) Left hand ( ) Both hands

9. Please rank the keyboards from most preferred (1), to least preferred (6).

Tablet Keyboard:	2
Leap Surface Keyboard:	3
Leap Air Static Keyboard:	5
Leap Air Pinch Keyboard:	6
Leap Air Dynamic Keyboard:	4
Leap Air Bimodal Keyboard:	1

Save Survey

Figure B.2. The electronic exit survey that participants were required to fill out after completing all tasks.

APPENDIX C

Participant Generated Paths

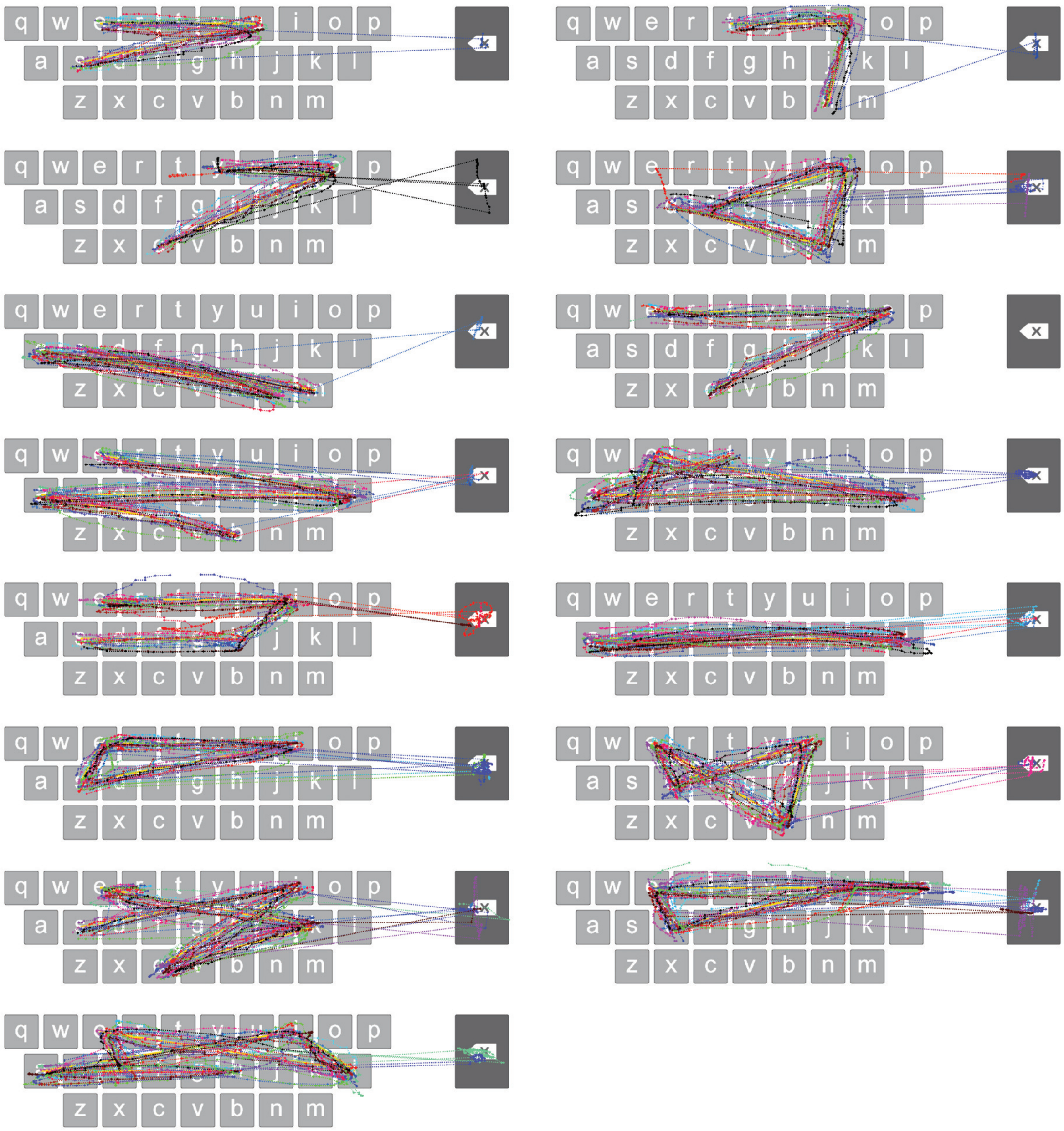


Figure C.1. The participant-generated word-gesture paths using the Touch Screen Keyboard for the dictionary words: ‘sue’, ‘tin’, ‘coy’, ‘find’, ‘mans’, ‘core’, ‘bale’, ‘tales’, ‘shier’, ‘falls’, ‘sires’, ‘dubbed’, ‘sicker’, ‘dipped’, ‘hailed’.

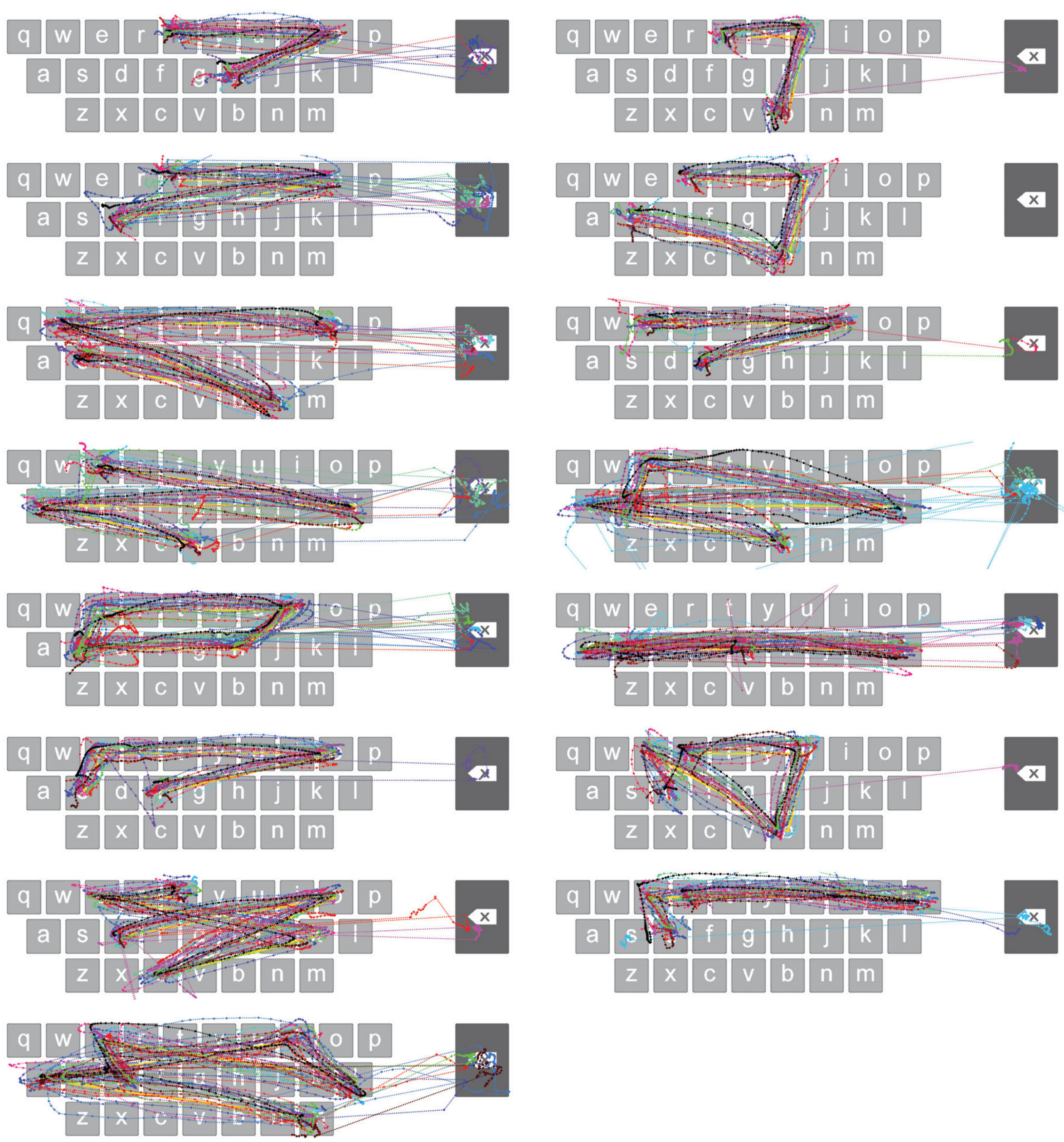


Figure C.2. The participant-generated word-gesture paths using the Leap Surface Keyboard for the dictionary words: ‘hot’, ‘tub’, ‘dot’, ‘rubs’, ‘owns’, ‘fire’, ‘vale’, ‘bales’, ‘shies’, ‘galls’, ‘fores’, ‘rubbed’, ‘docket’, ‘ripped’, ‘mailed’.

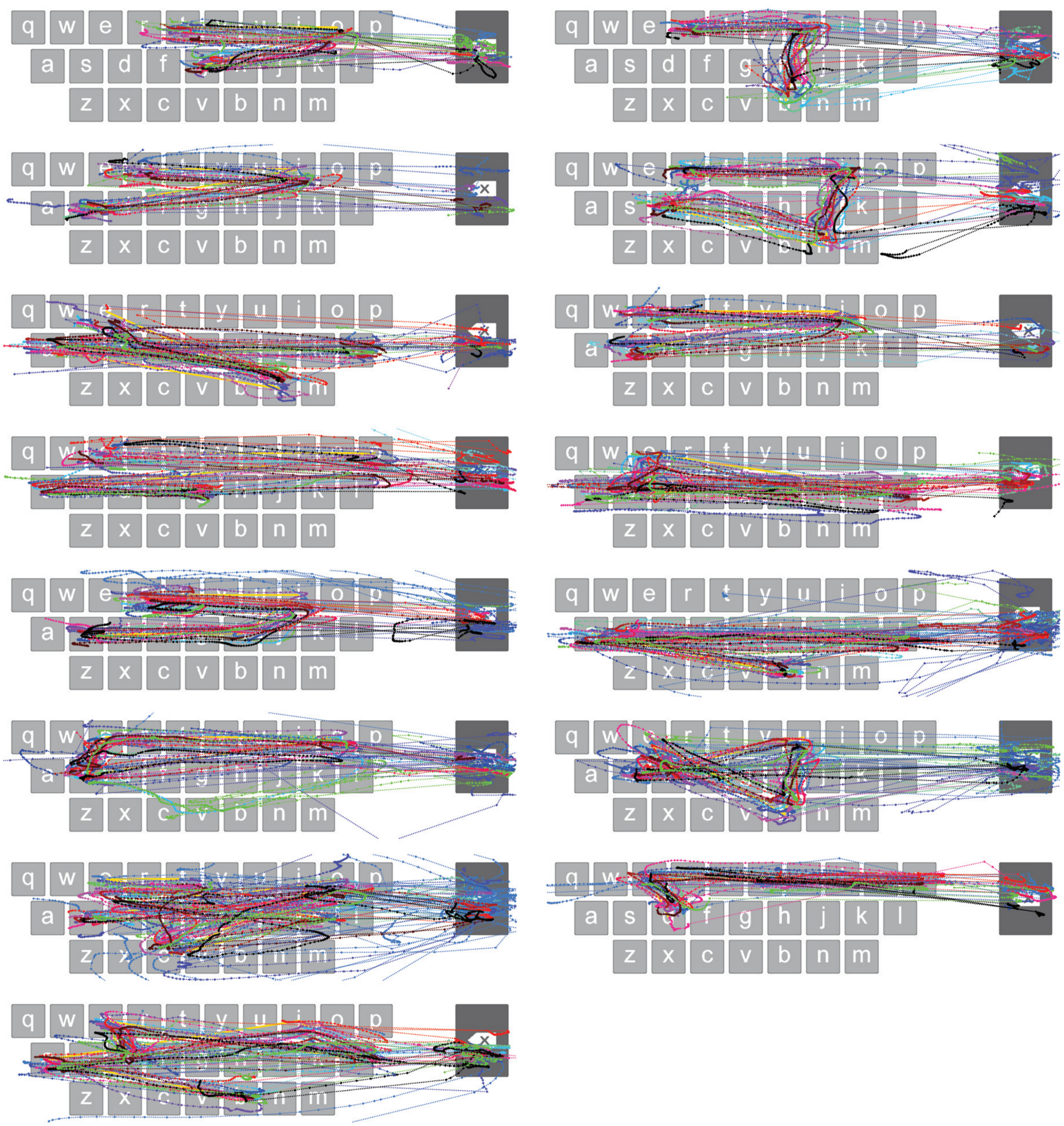


Figure C.3. The participant-generated word-gesture paths using the Leap Static-air Keyboard for the dictionary words: ‘got’, ‘rub’, ‘sir’, ‘rind’, ‘lane’, ‘sire’, ‘gape’, ‘hales’, ‘shirt’, ‘balls’, ‘sores’, ‘subbed’, ‘socket’, ‘yipped’, ‘bailed’.

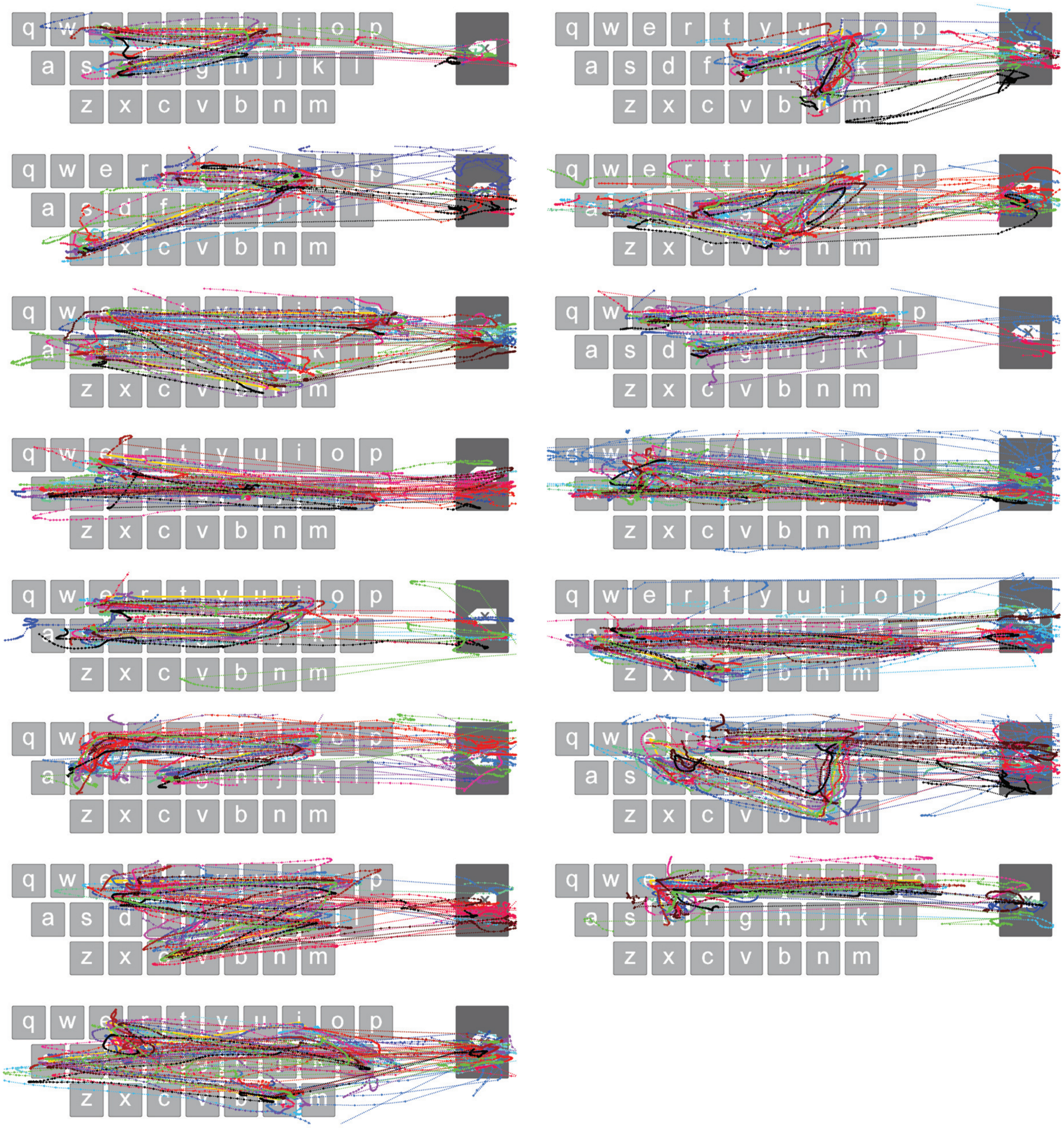


Figure C.4. The participant-generated word-gesture paths using the Leap Predictive-air Keyboard for the dictionary words: ‘due’, ‘gin’, ‘zit’, ‘fibs’, ‘pens’, ‘fore’, ‘hale’, ‘gales’, ‘shire’, ‘calls’, ‘fires’, ‘tinned’, ‘rocker’, ‘topped’, ‘nailed’.

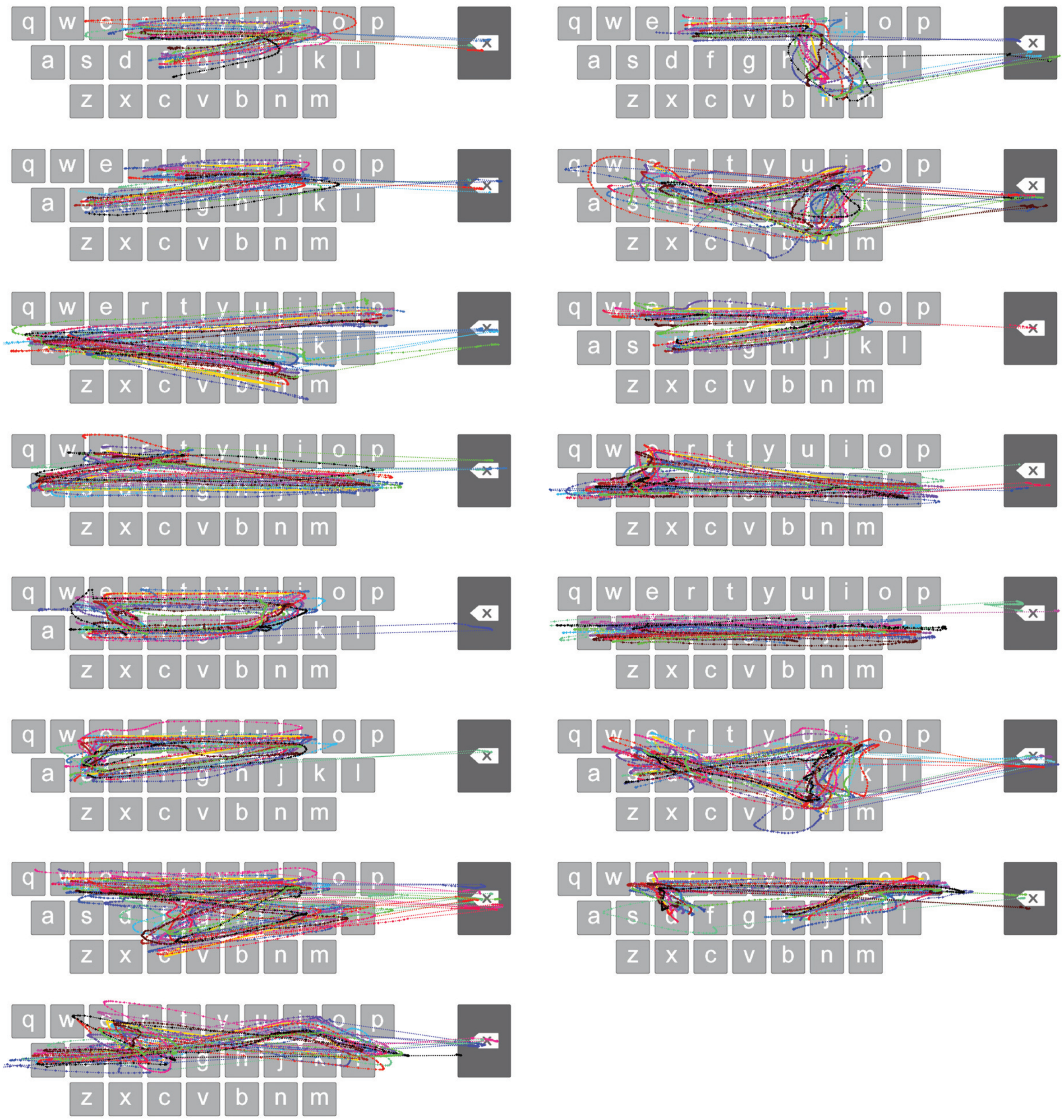


Figure C.5. The participant-generated word-gesture paths using the Leap Bimodal-air Keyboard for the dictionary words: ‘fir’, ‘run’, ‘sit’, ‘fine’, ‘pans’, ‘dire’, ‘tale’, ‘dales’, ‘shied’, ‘halls’, ‘sites’, ‘sinned’, ‘wicker’, ‘hopped’, ‘failed’.

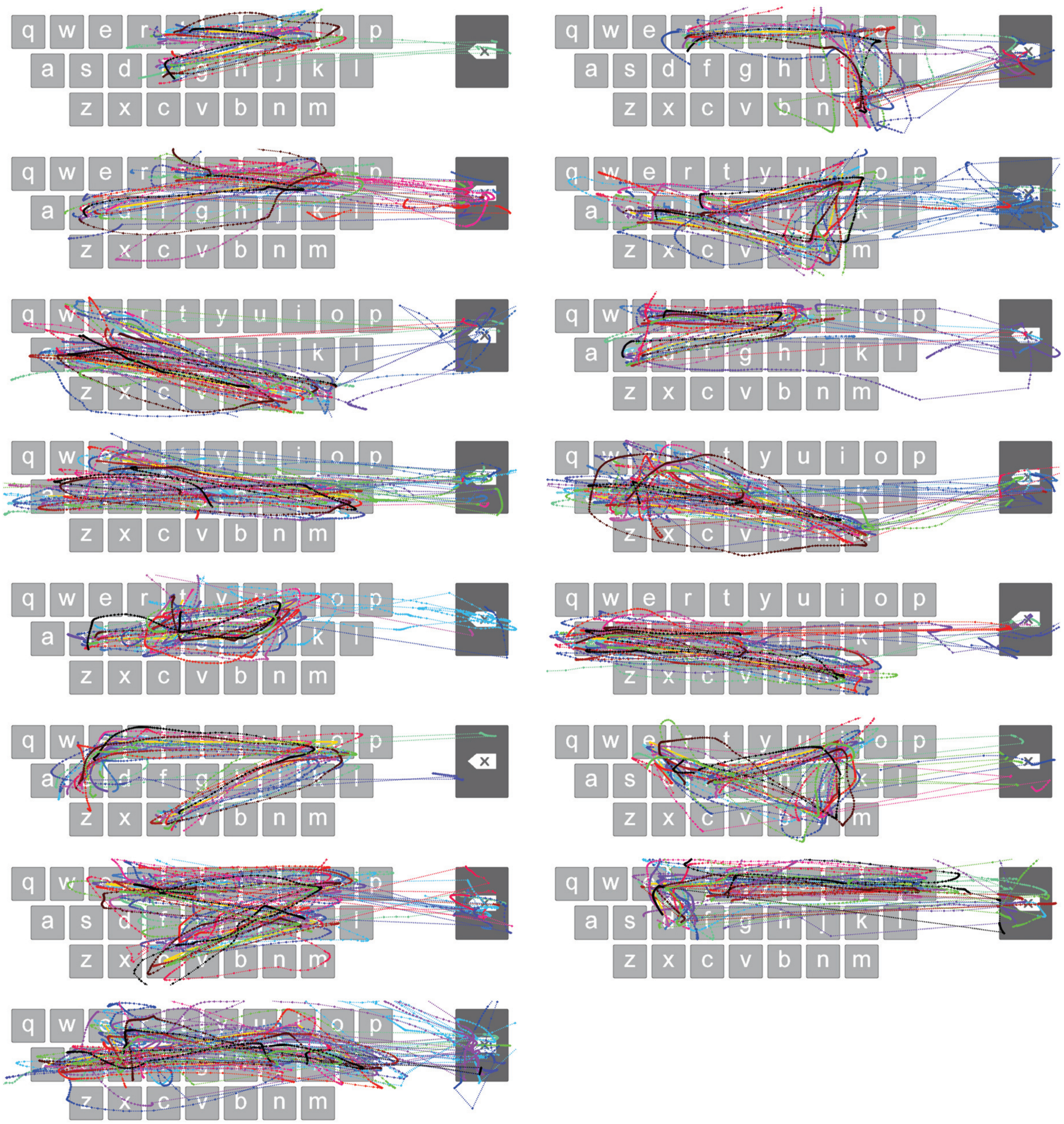


Figure C.6. The participant-generated word-gesture paths using the Leap Pinch-air Keyboard for the dictionary words: ‘fit’, ‘rim’, ‘soy’, ‘fins’, ‘mane’, ‘sure’, ‘gale’, ‘games’, ‘shift’, ‘gamma’, ‘cores’, ‘finned’, ‘rocket’, ‘tipped’, ‘jailed’.

APPENDIX D

Additional Results

This appendix contains box plots showing the medians of the depdent measures and 25th and 75th percentiles. In addition, tables are shown for dependent measures that required additional analysis. The statistical methods used for this project were one-way ANOVAs for each set of dependent measures (Wu and Hamada 2009, Neter, Wasserman, and Kutner 1996), followed by a post-hoc analysis using Tukey's Honest Significant Difference (HSD) for multiple-compare (Hochberg and Tamhane 2009). The ranking system from the exit survey used the Friedman's test in conjunction with Tukey's HSD for a post-hoc analysis (Hollander, Wolfe, and Chicken 2013).

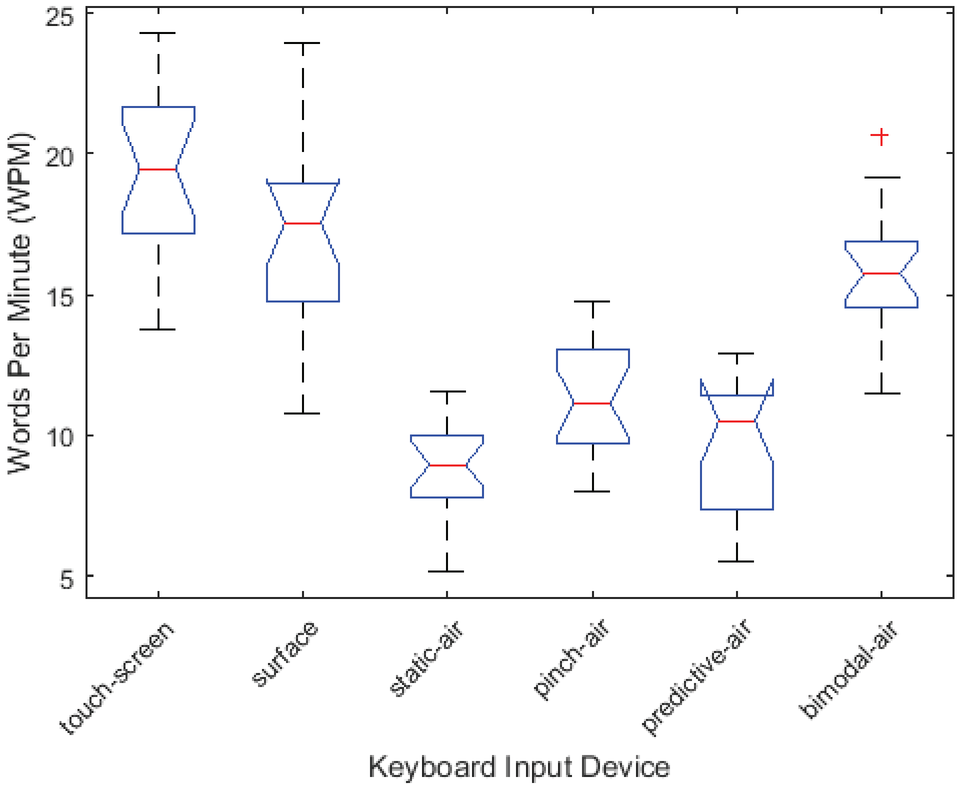


Figure D.1. Median Text-entry rates for each keyboard showing the 25th and 75th percentiles.

Table D.1. Tukey's Honest Significant Difference with multiple comparison for text-entry rate.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-0.0837	2.3522	4.7882	0.0648
touchscreen	static-air	8.4479	10.8839	13.3198	0.0000
touchscreen	pinch-air	5.6920	8.1280	10.5639	0.0000
touchscreen	predictive-air	7.4227	9.8587	12.2946	0.0000
touchscreen	bimodal-air	1.2832	3.7192	6.1551	0.0003
surface	static-air	6.0957	8.5316	10.9676	0.0000
surface	pinch-air	3.3398	5.7757	8.2117	0.0000
surface	predictive-air	5.0705	7.5064	9.9424	0.0000
surface	bimodal-air	-1.0690	1.3669	3.8029	0.5809
static-air	pinch-air	-5.1919	-2.7559	-0.3199	0.0170
static-air	predictive-air	-3.4612	-1.0252	1.4108	0.8249
static-air	bimodal-air	-9.6007	-7.1647	-4.7287	0.0000
pinch-air	predictive-air	-0.7053	1.7307	4.1667	0.3146
pinch-air	bimodal-air	-6.8448	-4.4088	-1.9728	0.0000
predictive-air	bimodal-air	-8.5755	-6.1395	-3.7035	0.0000

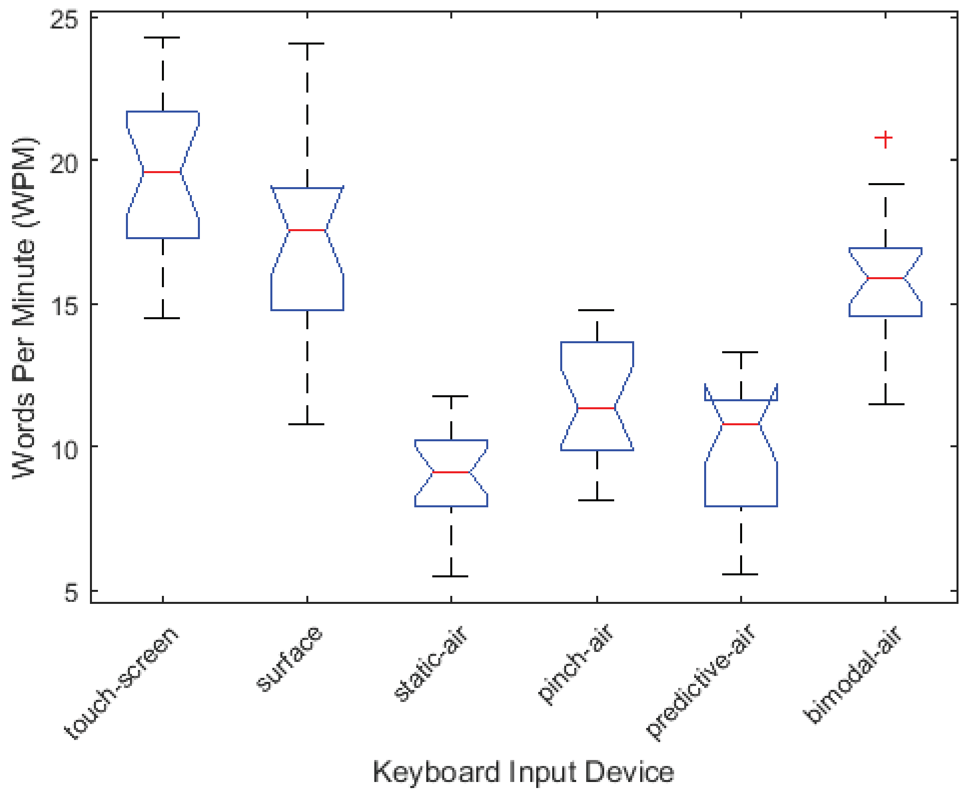


Figure D.2. Median text-entry rates using the shortest-transcribed modification for each keyboard showing the 25th and 75th percentiles.

Table D.2. Tukey's Honest Significant Difference with multiple comparison for Text-entry rate modified with shortest-transcribed.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-0.0162	2.4063	4.8288	0.0526
touchscreen	static-air	8.4069	10.8295	13.2520	0.0000
touchscreen	pinch-air	5.6271	8.0496	10.4722	0.0000
touchscreen	predictive-air	7.3096	9.7321	12.1546	0.0000
touchscreen	bimodal-air	1.3652	3.7877	6.2103	0.0002
surface	static-air	6.0006	8.4232	10.8457	0.0000
surface	pinch-air	3.2208	5.6433	8.0659	0.0000
surface	predictive-air	4.9033	7.3258	9.7483	0.0000
surface	bimodal-air	-1.0411	1.3814	3.8040	0.5636
static-air	pinch-air	-5.2024	-2.7798	-0.3573	0.0148
static-air	predictive-air	-3.5199	-1.0974	1.3251	0.7757
static-air	bimodal-air	-9.4643	-7.0417	-4.6192	0.0000
pinch-air	predictive-air	-0.7401	1.6825	4.1050	0.3399
pinch-air	bimodal-air	-6.6844	-4.2619	-1.8394	0.0000
predictive-air	bimodal-air	-8.3669	-5.9443	-3.5218	0.0000

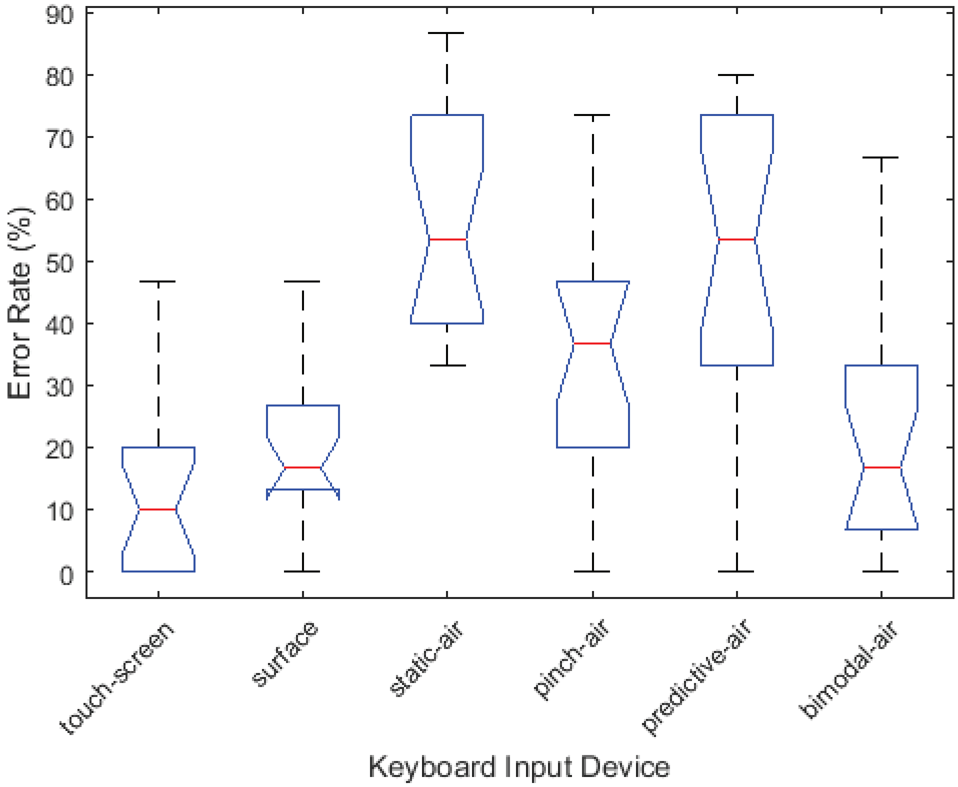


Figure D.3. Median Minimum Word Distance for each keyboard showing the 25th and 75th percentiles.

Table D.3. Tukey's Honest Significant Difference with multiple comparison for Minimum Word Distance.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-23.8140	-5.9259	11.9622	0.9287
touchscreen	static-air	-61.5918	-43.7037	-25.8156	0.0000
touchscreen	pinch-air	-39.3696	-21.4815	-3.5934	0.0091
touchscreen	predictive-air	-53.8140	-35.9259	-18.0378	0.0000
touchscreen	bimodal-air	-25.2955	-7.4074	10.4807	0.8346
surface	static-air	-55.6659	-37.7778	-19.8897	0.0000
surface	pinch-air	-33.4437	-15.5556	2.3326	0.1263
surface	predictive-air	-47.8881	-30.0000	-12.1119	0.0001
surface	bimodal-air	-19.3696	-1.4815	16.4066	0.9999
static-air	pinch-air	4.3341	22.2222	40.1103	0.0062
static-air	predictive-air	-10.1103	7.7778	25.6659	0.8043
static-air	bimodal-air	18.4082	36.2963	54.1844	0.0000
pinch-air	predictive-air	-32.3326	-14.4444	3.4437	0.1859
pinch-air	bimodal-air	-3.8140	14.0741	31.9622	0.2096
predictive-air	bimodal-air	10.6304	28.5185	46.4066	0.0002

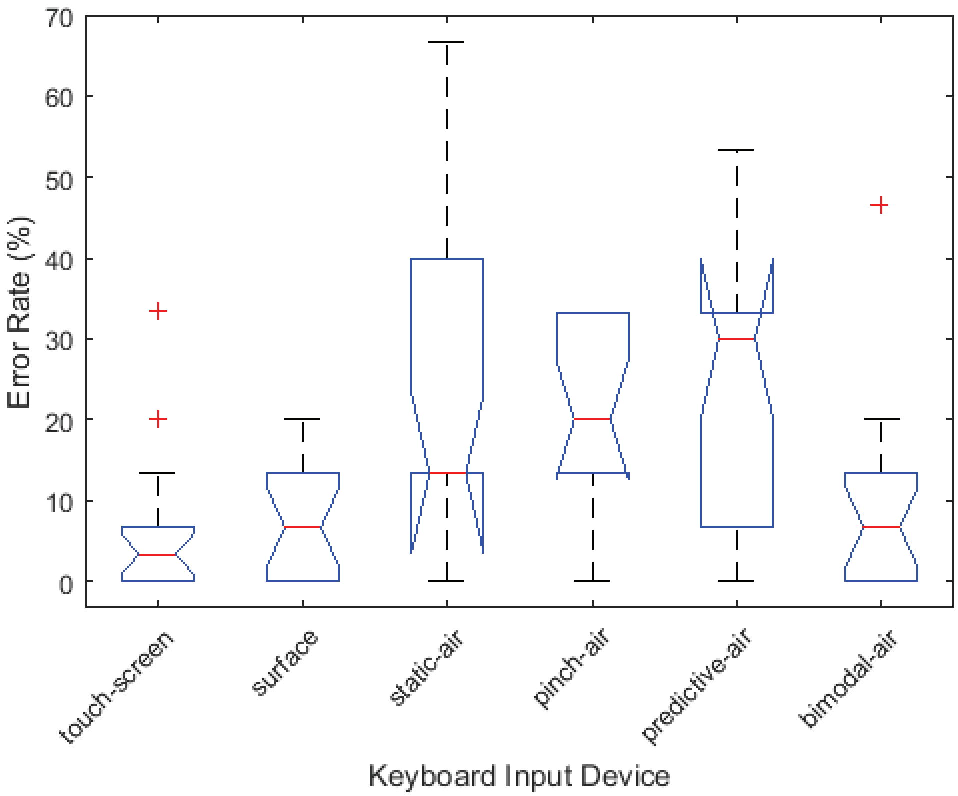


Figure D.4. Median Minimum Word Distance using the shortest-transcribed modification for each keyboard showing the 25th and 75th percentiles.

Table D.4. Tukey's Honest Significant Difference with multiple comparison for Minimum Word Distance using shortest-transcribed modification.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-13.0458	0.0000	13.0458	1.0000
touchscreen	static-air	-32.3051	-19.2593	-6.2135	0.0006
touchscreen	pinch-air	-26.7495	-13.7037	-0.6579	0.0336
touchscreen	predictive-air	-32.6754	-19.6296	-6.5838	0.0004
touchscreen	bimodal-air	-16.7495	-3.7037	9.3421	0.9623
surface	static-air	-32.3051	-19.2593	-6.2135	0.0006
surface	pinch-air	-26.7495	-13.7037	-0.6579	0.0336
surface	predictive-air	-32.6754	-19.6296	-6.5838	0.0004
surface	bimodal-air	-16.7495	-3.7037	9.3421	0.9623
static-air	pinch-air	-7.4902	5.5556	18.6013	0.8177
static-air	predictive-air	-13.4162	-0.3704	12.6754	1.0000
static-air	bimodal-air	2.5098	15.5556	28.6013	0.0099
pinch-air	predictive-air	-18.9717	-5.9259	7.1199	0.7737
pinch-air	bimodal-air	-3.0458	10.0000	23.0458	0.2349
predictive-air	bimodal-air	2.8801	15.9259	28.9717	0.0076

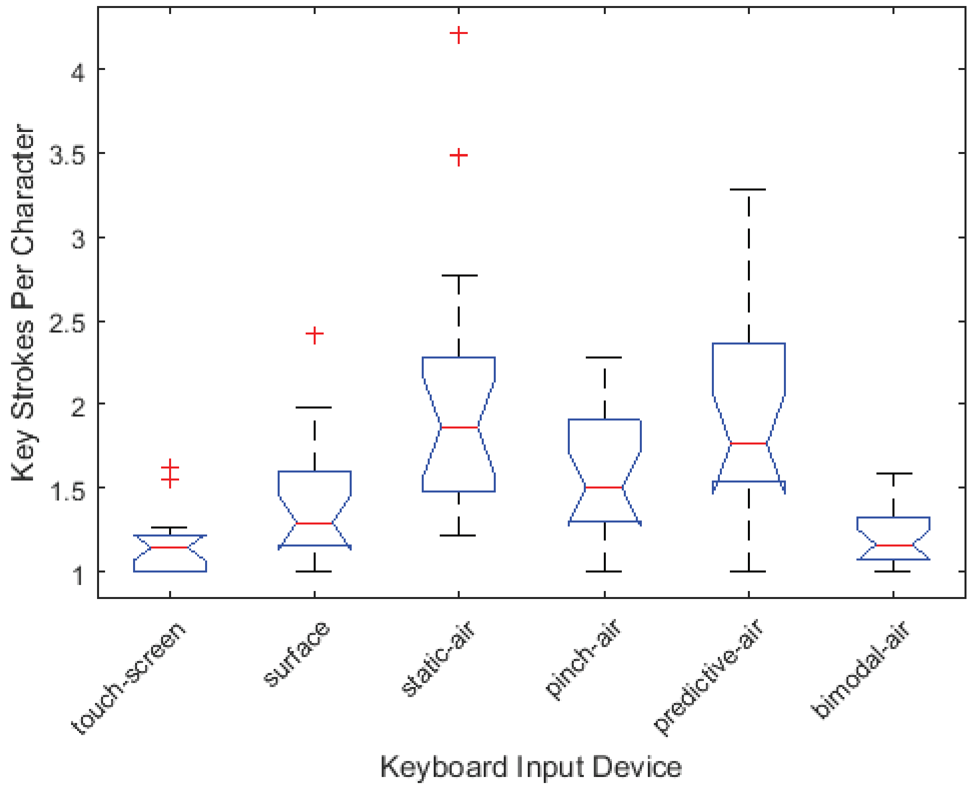


Figure D.5. Median Keystrokes Per Character for each keyboard showing the 25th and 75th percentiles.

Table D.5. Tukey's Honest Significant Difference with multiple comparison for Keystrokes Per Character.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-0.7298	-0.2597	0.2103	0.5972
touchscreen	static-air	-1.3559	-0.8859	-0.4158	0.0000
touchscreen	pinch-air	-0.8998	-0.4298	0.0403	0.0935
touchscreen	predictive-air	-1.2113	-0.7412	-0.2712	0.0002
touchscreen	bimodal-air	-0.5324	-0.0623	0.4077	0.9989
surface	static-air	-1.0962	-0.6262	-0.1561	0.0026
surface	pinch-air	-0.6401	-0.1701	0.3000	0.8993
surface	predictive-air	-0.9515	-0.4815	-0.0114	0.0414
surface	bimodal-air	-0.2726	0.1974	0.6675	0.8262
static-air	pinch-air	-0.0140	0.4561	0.9262	0.0626
static-air	predictive-air	-0.3254	0.1447	0.6147	0.9471
static-air	bimodal-air	0.3535	0.8236	1.2936	0.0000
pinch-air	predictive-air	-0.7815	-0.3114	0.1586	0.3935
pinch-air	bimodal-air	-0.1026	0.3675	0.8375	0.2157
predictive-air	bimodal-air	0.2089	0.6789	1.1490	0.0008

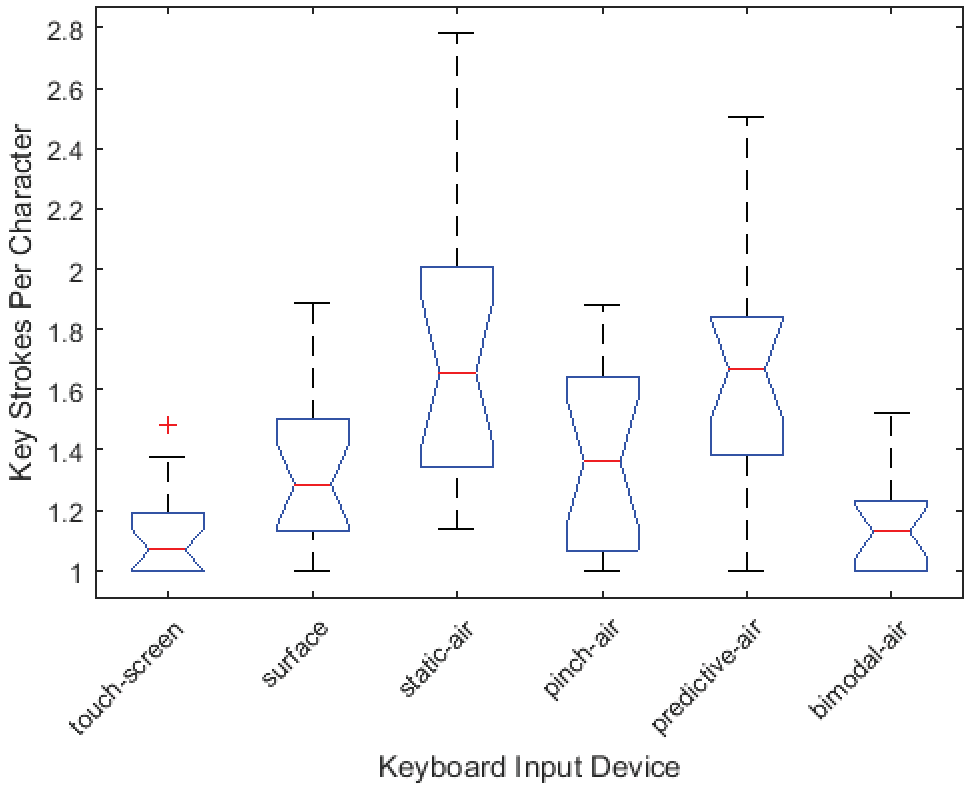


Figure D.6. Median Keystrokes Per Character using the shortest-transcribed modification for each keyboard showing the 25th and 75th percentiles.

Table D.6. Tukey's Honest Significant Difference with multiple comparison for Keystrokes Per Character using shortest-transcribed modification.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-0.5181	-0.2055	0.1070	0.4020
touchscreen	static-air	-0.9366	-0.6240	-0.3115	0.0000
touchscreen	pinch-air	-0.5532	-0.2407	0.0719	0.2304
touchscreen	predictive-air	-0.8488	-0.5363	-0.2237	0.0000
touchscreen	bimodal-air	-0.3346	-0.0220	0.2906	0.9999
surface	static-air	-0.7310	-0.4185	-0.1059	0.0024
surface	pinch-air	-0.3477	-0.0351	0.2774	0.9995
surface	predictive-air	-0.6433	-0.3307	-0.0182	0.0316
surface	bimodal-air	-0.1290	0.1835	0.4961	0.5313
static-air	pinch-air	0.0708	0.3833	0.6959	0.0072
static-air	predictive-air	-0.2248	0.0877	0.4003	0.9641
static-air	bimodal-air	0.2895	0.6020	0.9146	0.0000
pinch-air	predictive-air	-0.6082	-0.2956	0.0169	0.0748
pinch-air	bimodal-air	-0.0939	0.2187	0.5312	0.3316
predictive-air	bimodal-air	0.2017	0.5143	0.8269	0.0001

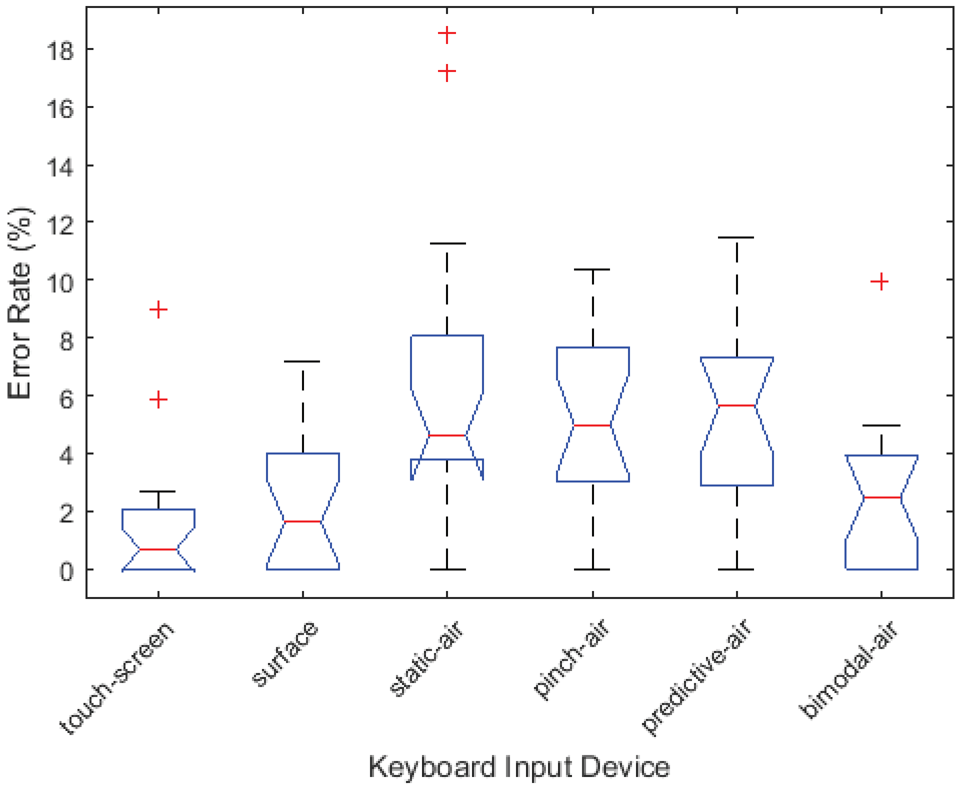


Figure D.7. Median Minimum String Distance using the shortest-transcribed modification for each keyboard showing the 25th and 75th percentiles.

Table D.7. Tukey's Honest Significant Difference with multiple comparison for Minimum String Distance using shortest-transcribed modification.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-3.7747	-0.6145	2.5457	0.9931
touchscreen	static-air	-7.9973	-4.8372	-1.6770	0.0003
touchscreen	pinch-air	-6.7038	-3.5437	-0.3835	0.0186
touchscreen	predictive-air	-6.9953	-3.8351	-0.6749	0.0081
touchscreen	bimodal-air	-4.0980	-0.9378	2.2223	0.9546
surface	static-air	-7.3828	-4.2227	-1.0625	0.0025
surface	pinch-air	-6.0893	-2.9292	0.2310	0.0856
surface	predictive-air	-6.3808	-3.2206	-0.0604	0.0431
surface	bimodal-air	-3.4835	-0.3233	2.8368	0.9997
static-air	pinch-air	-1.8667	1.2935	4.4537	0.8412
static-air	predictive-air	-2.1581	1.0021	4.1622	0.9403
static-air	bimodal-air	0.7392	3.8993	7.0595	0.0067
pinch-air	predictive-air	-3.4516	-0.2914	2.8687	0.9998
pinch-air	bimodal-air	-0.5543	2.6058	5.7660	0.1677
predictive-air	bimodal-air	-0.2629	2.8973	6.0574	0.0919

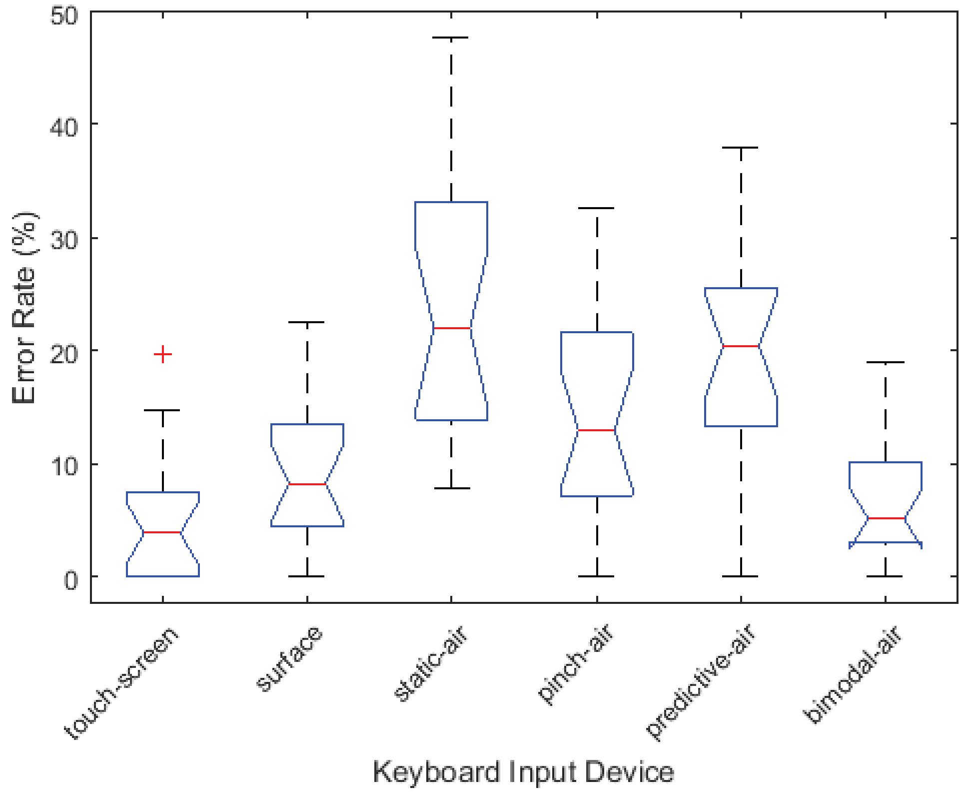


Figure D.8. Median Total Error Rate for each keyboard showing the 25th and 75th percentiles.

Table D.8. Tukey's Honest Significant Difference with multiple comparison for Total Error Rate.

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience Interval Upper End	p-value
touchscreen	surface	-12.9197	-4.3716	4.1766	0.6743
touchscreen	static-air	-27.2196	-18.6714	-10.1233	0.0000
touchscreen	pinch-air	-17.6640	-9.1158	-0.5677	0.0295
touchscreen	predictive-air	-23.6696	-15.1214	-6.5733	0.0000
touchscreen	bimodal-air	-10.4357	-1.8875	6.6606	0.9875
surface	static-air	-22.8480	-14.2999	-5.7517	0.0001
surface	pinch-air	-13.2924	-4.7443	3.8039	0.5926
surface	predictive-air	-19.2980	-10.7498	-2.2017	0.0054
surface	bimodal-air	-6.0641	2.4840	11.0322	0.9584
static-air	pinch-air	1.0075	9.5556	18.1038	0.0191
static-air	predictive-air	-4.9981	3.5500	12.0982	0.8329
static-air	bimodal-air	8.2358	16.7839	25.3321	0.0000
pinch-air	predictive-air	-14.5537	-6.0056	2.5425	0.3270
pinch-air	bimodal-air	-1.3198	7.2283	15.7764	0.1473
predictive-air	bimodal-air	4.6858	13.2339	21.7820	0.0003

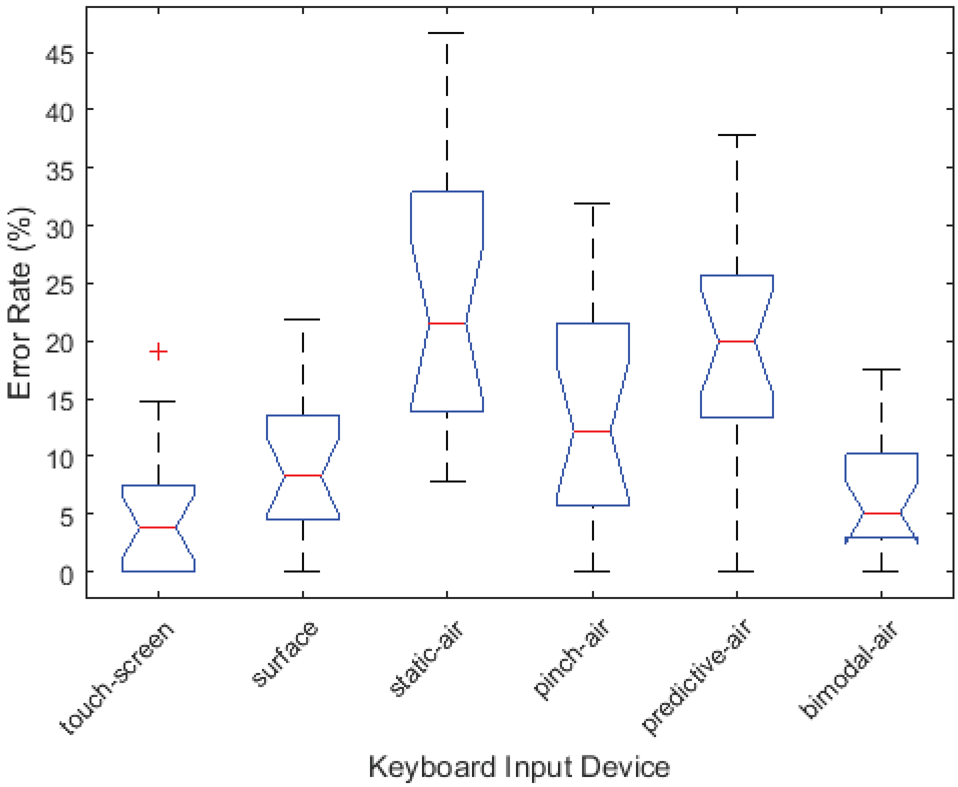


Figure D.9. Median Total Error Rate using the shortest-transcribed modification for each keyboard showing the 25th and 75th percentiles.

Table D.9. Tukey's Honest Significant Difference with multiple comparison for Total Error Rate using shortest-transcribed modification.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-12.6316	-4.3367	3.9581	0.6532
touchscreen	static-air	-26.7322	-18.4374	-10.1426	0.0000
touchscreen	pinch-air	-16.8635	-8.5686	-0.2738	0.0386
touchscreen	predictive-air	-22.7732	-14.4783	-6.1835	0.0000
touchscreen	bimodal-air	-10.1405	-1.8456	6.4492	0.9871
surface	static-air	-22.3955	-14.1007	-5.8058	0.0000
surface	pinch-air	-12.5268	-4.2319	4.0629	0.6765
surface	predictive-air	-18.4364	-10.1416	-1.8468	0.0075
surface	bimodal-air	-5.8037	2.4911	10.7859	0.9523
static-air	pinch-air	1.5739	9.8688	18.1636	0.0101
static-air	predictive-air	-4.3358	3.9591	12.2539	0.7351
static-air	bimodal-air	8.2969	16.5918	24.8866	0.0000
pinch-air	predictive-air	-14.2045	-5.9097	2.3851	0.3116
pinch-air	bimodal-air	-1.5718	6.7230	15.0178	0.1826
predictive-air	bimodal-air	4.3379	12.6327	20.9275	0.0003

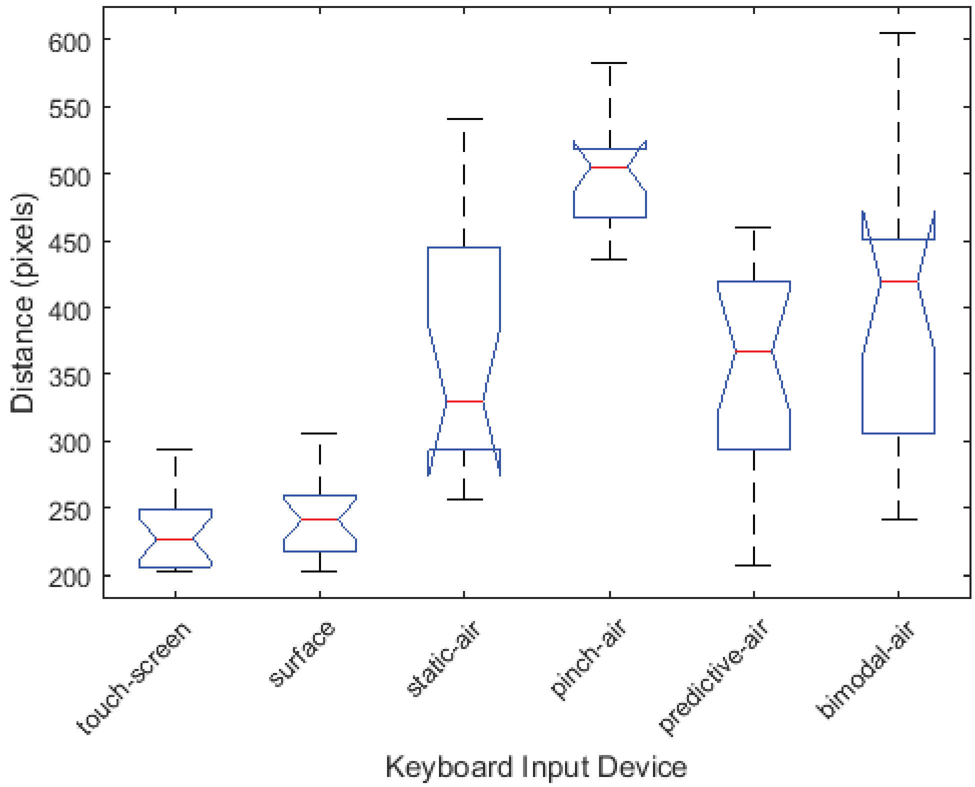


Figure D.10. Median Fréchet Distance for each keyboard showing the 25th and 75th percentiles.

Table D.10. Tukey's Honest Significant Difference with multiple comparison for Fréchet Distance.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-78.8540	-11.6328	55.5884	0.9960
touchscreen	static-air	-200.3012	-133.0800	-65.8588	0.0000
touchscreen	pinch-air	-337.2356	-270.0144	-202.7931	0.0000
touchscreen	predictive-air	-186.6273	-119.4061	-52.1849	0.0000
touchscreen	bimodal-air	-232.5464	-165.3252	-98.1040	0.0000
surface	static-air	-188.6684	-121.4472	-54.2260	0.0000
surface	pinch-air	-325.6027	-258.3815	-191.1603	0.0000
surface	predictive-air	-174.9945	-107.7733	-40.5521	0.0001
surface	bimodal-air	-220.9136	-153.6924	-86.4712	0.0000
static-air	pinch-air	-204.1555	-136.9343	-69.7131	0.0000
static-air	predictive-air	-53.5473	13.6739	80.8951	0.9914
static-air	bimodal-air	-99.4664	-32.2452	34.9760	0.7309
pinch-air	predictive-air	83.3870	150.6083	217.8295	0.0000
pinch-air	bimodal-air	37.4679	104.6891	171.9103	0.0002
predictive-air	bimodal-air	-113.1403	-45.9191	21.3021	0.3586

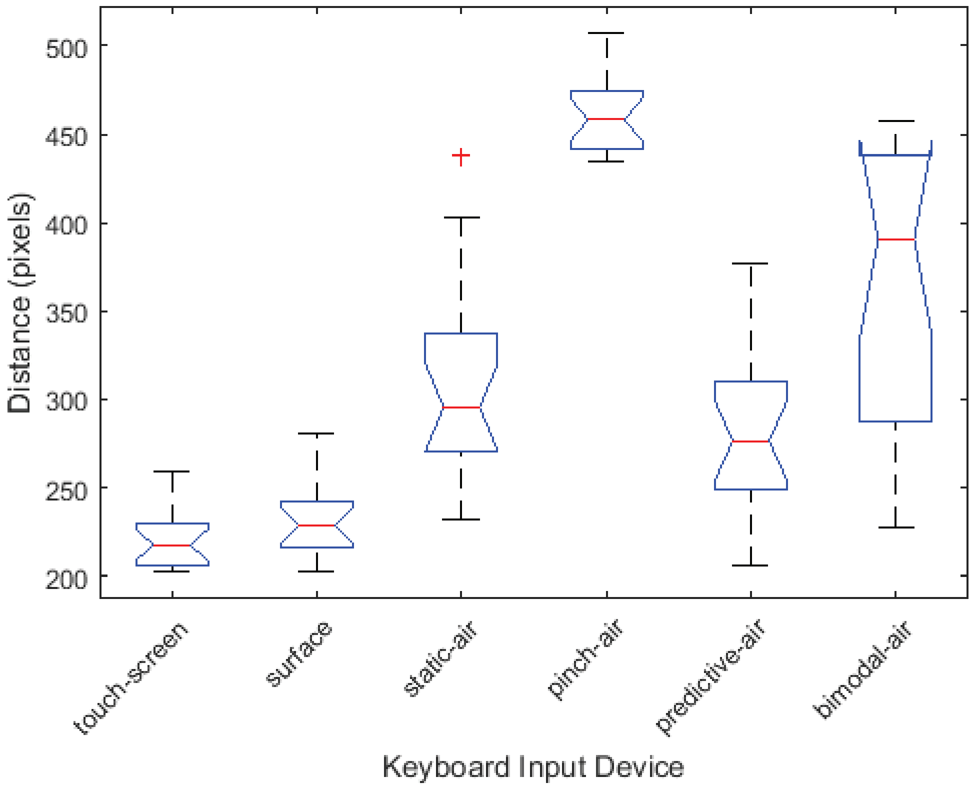


Figure D.11. Median Fréchet Distance using the shortest-transcribed modification for each keyboard showing the 25th and 75th percentiles.

Table D.11. Tukey's Honest Significant Difference with multiple comparison for Fréchet Distance using shortest-transcribed modification.

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience Interval Upper End	p-value
touchscreen	surface	-55.6656	-9.9934	35.6788	0.9881
touchscreen	static-air	-131.1279	-85.4556	-39.7834	0.0000
touchscreen	pinch-air	-283.8592	-238.1870	-192.5147	0.0000
touchscreen	predictive-air	-106.2951	-60.6229	-14.9506	0.0027
touchscreen	bimodal-air	-193.0612	-147.3890	-101.7167	0.0000
surface	static-air	-121.1345	-75.4622	-29.7900	0.0001
surface	pinch-air	-273.8658	-228.1936	-182.5213	0.0000
surface	predictive-air	-96.3017	-50.6295	-4.9572	0.0207
surface	bimodal-air	-183.0678	-137.3956	-91.7233	0.0000
static-air	pinch-air	-198.4036	-152.7313	-107.0591	0.0000
static-air	predictive-air	-20.8395	24.8328	70.5050	0.6140
static-air	bimodal-air	-107.6056	-61.9333	-16.2611	0.0020
pinch-air	predictive-air	131.8919	177.5641	223.2363	0.0000
pinch-air	bimodal-air	45.1258	90.7980	136.4702	0.0000
predictive-air	bimodal-air	-132.4383	-86.7661	-41.0939	0.0000

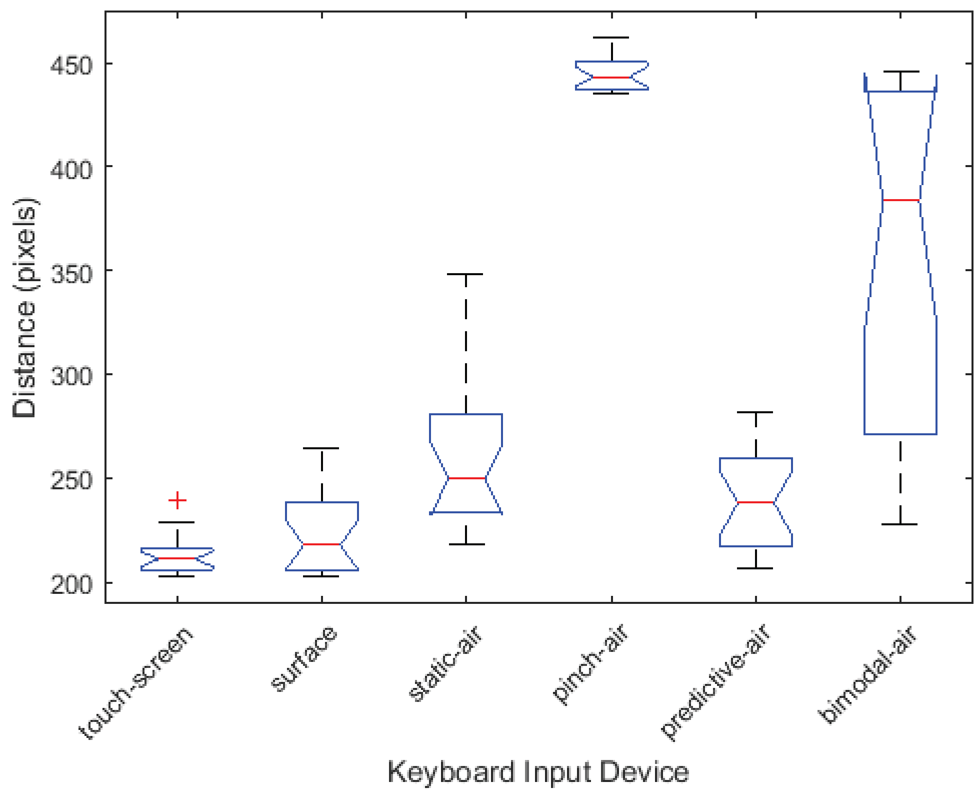


Figure D.12. Median Fréchet Distance using the backspace-transcribed modification for each keyboard showing the 25th and 75th percentiles.

Table D.12. Tukey's Honest Significant Difference with multiple comparison for Fréchet Distance using backspace-transcribed modification.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-49.6728	-11.2179	27.2370	0.9577
touchscreen	static-air	-85.0013	-46.5465	-8.0916	0.0084
touchscreen	pinch-air	-270.8703	-232.4155	-193.9606	0.0000
touchscreen	predictive-air	-63.8306	-25.3757	13.0791	0.3981
touchscreen	bimodal-air	-182.9460	-144.4911	-106.0363	0.0000
surface	static-air	-73.7834	-35.3285	3.1263	0.0907
surface	pinch-air	-259.6524	-221.1975	-182.7427	0.0000
surface	predictive-air	-52.6127	-14.1578	24.2970	0.8924
surface	bimodal-air	-171.7281	-133.2732	-94.8183	0.0000
static-air	pinch-air	-224.3239	-185.8690	-147.4141	0.0000
static-air	predictive-air	-17.2842	21.1707	59.6256	0.6011
static-air	bimodal-air	-136.3995	-97.9447	-59.4898	0.0000
pinch-air	predictive-air	168.5848	207.0397	245.4946	0.0000
pinch-air	bimodal-air	49.4694	87.9243	126.3792	0.0000
predictive-air	bimodal-air	-157.5703	-119.1154	-80.6605	0.0000

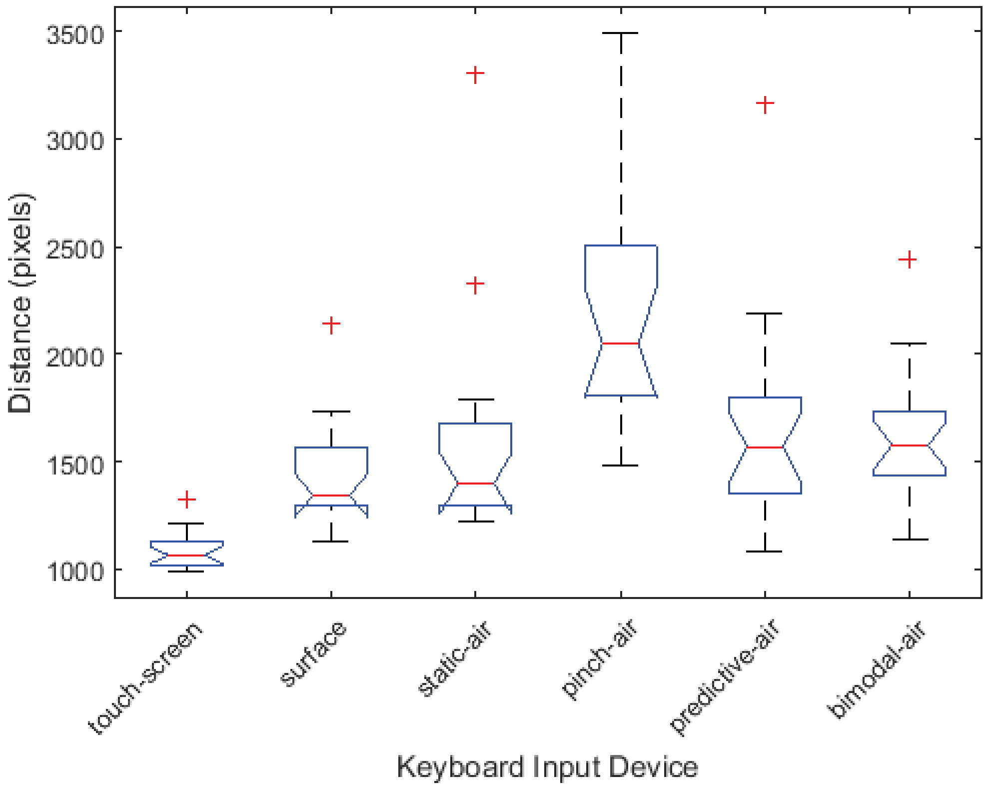


Figure D.13. Median Word-gesture Distance for each keyboard showing the 25th and 75th percentiles.

Table D.13. Tukey's Honest Significant Difference with multiple comparison for Word-gesture Distance.

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience Interval Upper End	p-value
touchscreen	surface	-722.3100	-336.4100	49.4910	0.1245
touchscreen	static-air	-889.2300	-503.3300	-117.4300	0.0034
touchscreen	pinch-air	-1463.6000	-1077.7000	-691.8200	0.0000
touchscreen	predictive-air	-933.1100	-547.2100	-161.3100	0.0011
touchscreen	bimodal-air	-926.0700	-540.1700	-154.2700	0.0013
surface	static-air	-552.8200	-166.9200	218.9800	0.8077
surface	pinch-air	-1127.2000	-741.3100	-355.4100	0.0000
surface	predictive-air	-596.7000	-210.8000	175.1000	0.6092
surface	bimodal-air	-589.6600	-203.7600	182.1500	0.6435
static-air	pinch-air	-960.3000	-574.3900	-188.4900	0.0005
static-air	predictive-air	-429.7800	-43.8820	342.0200	0.9995
static-air	bimodal-air	-422.7400	-36.8390	349.0600	0.9998
pinch-air	predictive-air	144.6100	530.5100	916.4100	0.0017
pinch-air	bimodal-air	151.6500	537.5500	923.4600	0.0014
predictive-air	bimodal-air	-378.8600	7.0433	392.9400	1.0000

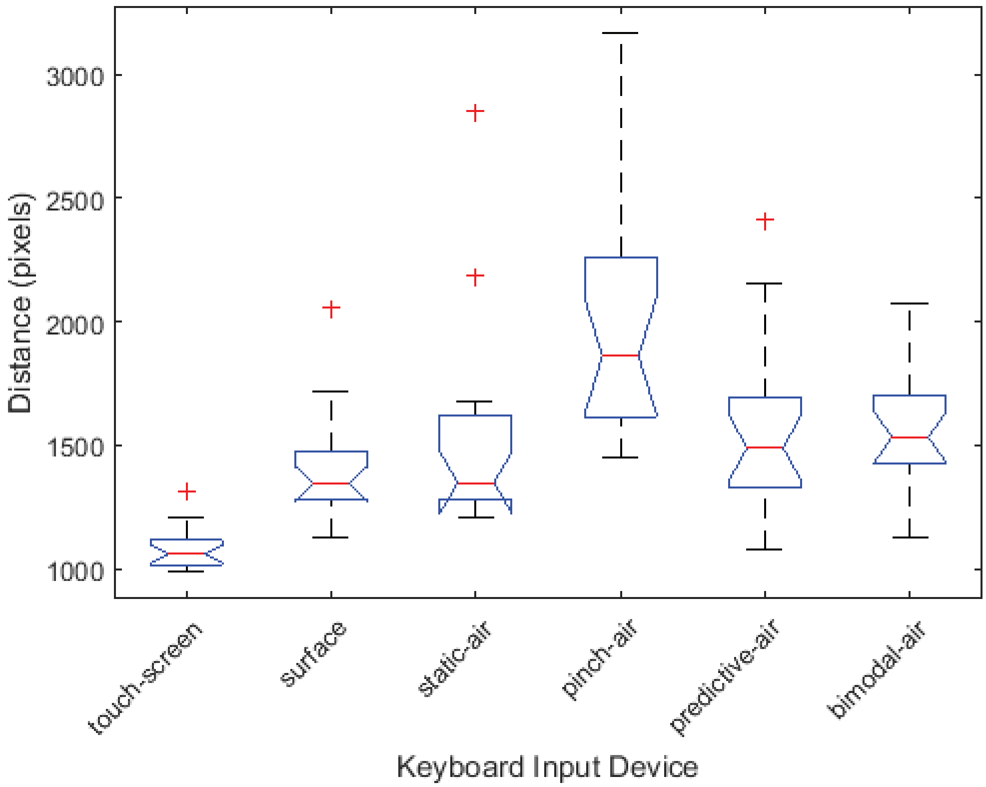


Figure D.14. Median Word-gesture Distance using the shortest-transcribed modification for each keyboard showing the 25th and 75th percentiles.

Table D.14. Tukey's Honest Significant Difference with multiple comparison for Word-gesture Distance using shortest-transcribed modification.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-630.0100	-315.2300	-0.4611	0.0494
touchscreen	static-air	-758.2800	-443.5100	-128.7300	0.0012
touchscreen	pinch-air	-1214.9000	-900.1000	-585.3300	0.0000
touchscreen	predictive-air	-770.2100	-455.4400	-140.6600	0.0008
touchscreen	bimodal-air	-799.9600	-485.1900	-170.4200	0.0003
surface	static-air	-443.0400	-128.2700	186.5000	0.8437
surface	pinch-air	-899.6400	-584.8700	-270.0900	0.0000
surface	predictive-air	-454.9800	-140.2000	174.5700	0.7878
surface	bimodal-air	-484.7300	-169.9600	144.8200	0.6211
static-air	pinch-air	-771.3700	-456.6000	-141.8200	0.0008
static-air	predictive-air	-326.7100	-11.9330	302.8400	1.0000
static-air	bimodal-air	-356.4600	-41.6840	273.0900	0.9989
pinch-air	predictive-air	129.8900	444.6600	759.4400	0.0011
pinch-air	bimodal-air	100.1400	414.9100	729.6900	0.0030
predictive-air	bimodal-air	-344.5200	-29.7520	285.0200	0.9998

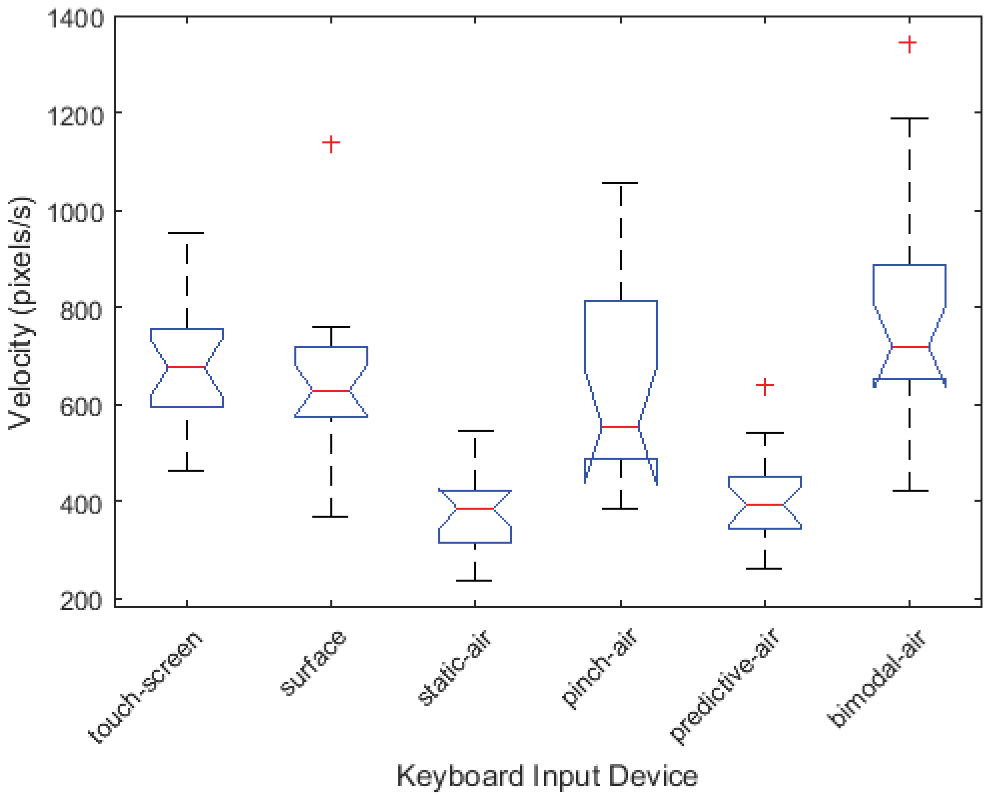


Figure D.15. Median Word-gesture Velocity for each keyboard showing the 25th and 75th percentiles.

Table D.15. Tukey's Honest Significant Difference with multiple comparison for Word-gesture Velocity.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-124.4400	32.2030	188.8500	0.9910
touchscreen	static-air	135.7300	292.3800	449.0200	0.0000
touchscreen	pinch-air	-127.1800	29.4680	186.1200	0.9940
touchscreen	predictive-air	111.0400	267.6900	424.3400	0.0000
touchscreen	bimodal-air	-260.9500	-104.3100	52.3400	0.3877
surface	static-air	103.5300	260.1700	416.8200	0.0001
surface	pinch-air	-159.3800	-2.7346	153.9100	1.0000
surface	predictive-air	78.8390	235.4900	392.1300	0.0004
surface	bimodal-air	-293.1600	-136.5100	20.1370	0.1248
static-air	pinch-air	-419.5500	-262.9100	-106.2600	0.0001
static-air	predictive-air	-181.3300	-24.6870	131.9600	0.9974
static-air	bimodal-air	-553.3300	-396.6800	-240.0400	0.0000
pinch-air	predictive-air	81.5740	238.2200	394.8700	0.0004
pinch-air	bimodal-air	-290.4200	-133.7800	22.8720	0.1397
predictive-air	bimodal-air	-528.6400	-372.0000	-215.3500	0.0000

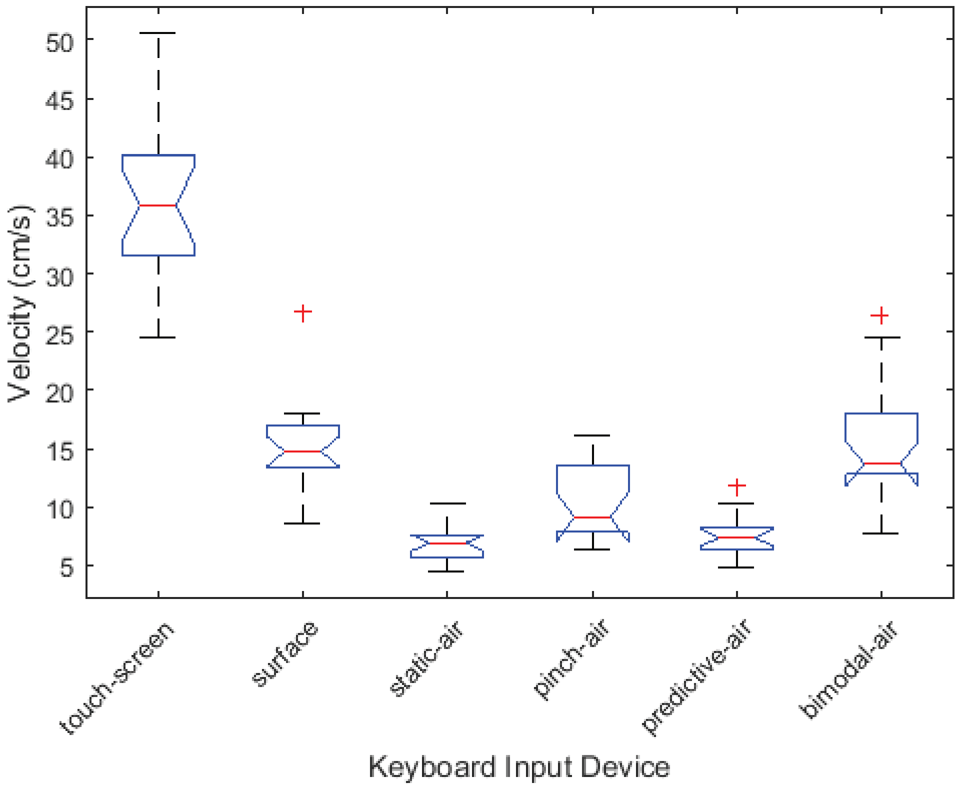


Figure D.16. Median Hand Velocity for each keyboard showing the 25th and 75th percentiles.

Table D.16. Tukey's Honest Significant Difference with multiple comparison for Hand Velocity.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	16.7046	20.6775	24.6504	0.0000
touchscreen	static-air	24.8793	28.8522	32.8251	0.0000
touchscreen	pinch-air	21.2448	25.2177	29.1906	0.0000
touchscreen	predictive-air	24.1600	28.1329	32.1058	0.0000
touchscreen	bimodal-air	16.5779	20.5508	24.5237	0.0000
surface	static-air	4.2018	8.1747	12.1476	0.0000
surface	pinch-air	0.5673	4.5403	8.5132	0.0154
surface	predictive-air	3.4825	7.4554	11.4284	0.0000
surface	bimodal-air	-4.0996	-0.1267	3.8462	1.0000
static-air	pinch-air	-7.6074	-3.6345	0.3385	0.0932
static-air	predictive-air	-4.6922	-0.7193	3.2536	0.9950
static-air	bimodal-air	-12.2743	-8.3014	-4.3285	0.0000
pinch-air	predictive-air	-1.0577	2.9152	6.8881	0.2798
pinch-air	bimodal-air	-8.6399	-4.6670	-0.6940	0.0116
predictive-air	bimodal-air	-11.5551	-7.5821	-3.6092	0.0000

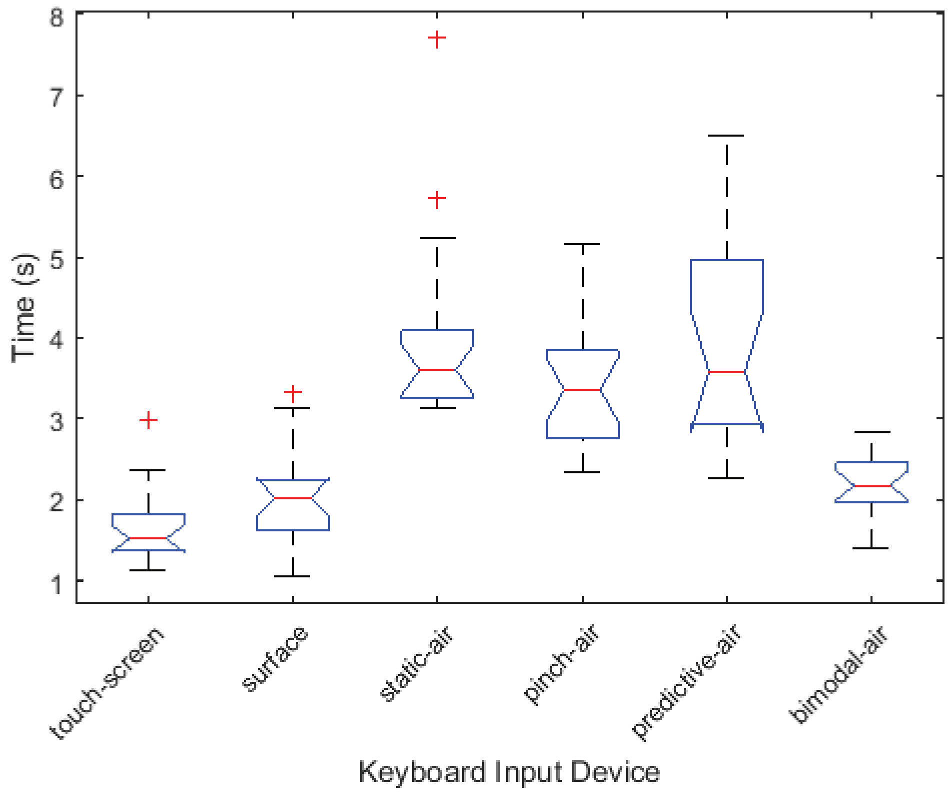


Figure D.17. Median Word-gesture Duration for each keyboard showing the 25th and 75th percentiles.

Table D.17. Tukey's Honest Significant Difference with multiple comparison for Word-gesture Duration.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-1.1732	-0.3842	0.4047	0.7181
touchscreen	static-air	-3.1457	-2.3568	-1.5679	0.0000
touchscreen	pinch-air	-2.5019	-1.7130	-0.9241	0.0000
touchscreen	predictive-air	-3.0051	-2.2162	-1.4272	0.0000
touchscreen	bimodal-air	-1.3204	-0.5315	0.2574	0.3743
surface	static-air	-2.7615	-1.9726	-1.1836	0.0000
surface	pinch-air	-2.1177	-1.3288	-0.5398	0.0001
surface	predictive-air	-2.6208	-1.8319	-1.0430	0.0000
surface	bimodal-air	-0.9362	-0.1473	0.6417	0.9943
static-air	pinch-air	-0.1451	0.6438	1.4327	0.1766
static-air	predictive-air	-0.6483	0.1406	0.9296	0.9954
static-air	bimodal-air	1.0364	1.8253	2.6142	0.0000
pinch-air	predictive-air	-1.2921	-0.5031	0.2858	0.4373
pinch-air	bimodal-air	0.3926	1.1815	1.9704	0.0005
predictive-air	bimodal-air	0.8957	1.6847	2.4736	0.0000

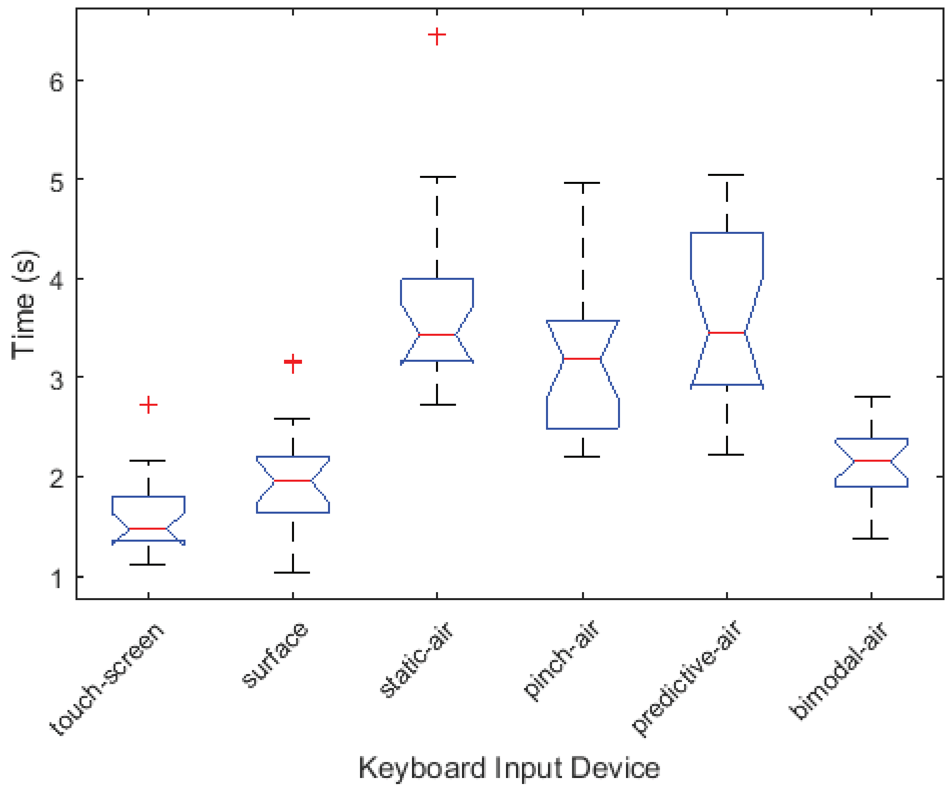


Figure D.18. Median Word-gesture Duration using the shortest-transcribed modification for each keyboard showing the 25th and 75th percentiles.

Table D.18. Tukey's Honest Significant Difference with multiple comparison for Word-gesture Duration using shortest-transcribed modification.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-1.0464	-0.3939	0.2587	0.5004
touchscreen	static-air	-2.7937	-2.1411	-1.4885	0.0000
touchscreen	pinch-air	-2.2167	-1.5641	-0.9115	0.0000
touchscreen	predictive-air	-2.6087	-1.9561	-1.3036	0.0000
touchscreen	bimodal-air	-1.2034	-0.5508	0.1018	0.1488
surface	static-air	-2.3998	-1.7472	-1.0947	0.0000
surface	pinch-air	-1.8228	-1.1702	-0.5177	0.0000
surface	predictive-air	-2.2148	-1.5623	-0.9097	0.0000
surface	bimodal-air	-0.8095	-0.1569	0.4957	0.9817
static-air	pinch-air	-0.0756	0.5770	1.2296	0.1147
static-air	predictive-air	-0.4676	0.1850	0.8375	0.9626
static-air	bimodal-air	0.9378	1.5903	2.2429	0.0000
pinch-air	predictive-air	-1.0446	-0.3920	0.2605	0.5058
pinch-air	bimodal-air	0.3608	1.0133	1.6659	0.0002
predictive-air	bimodal-air	0.7528	1.4054	2.0579	0.0000

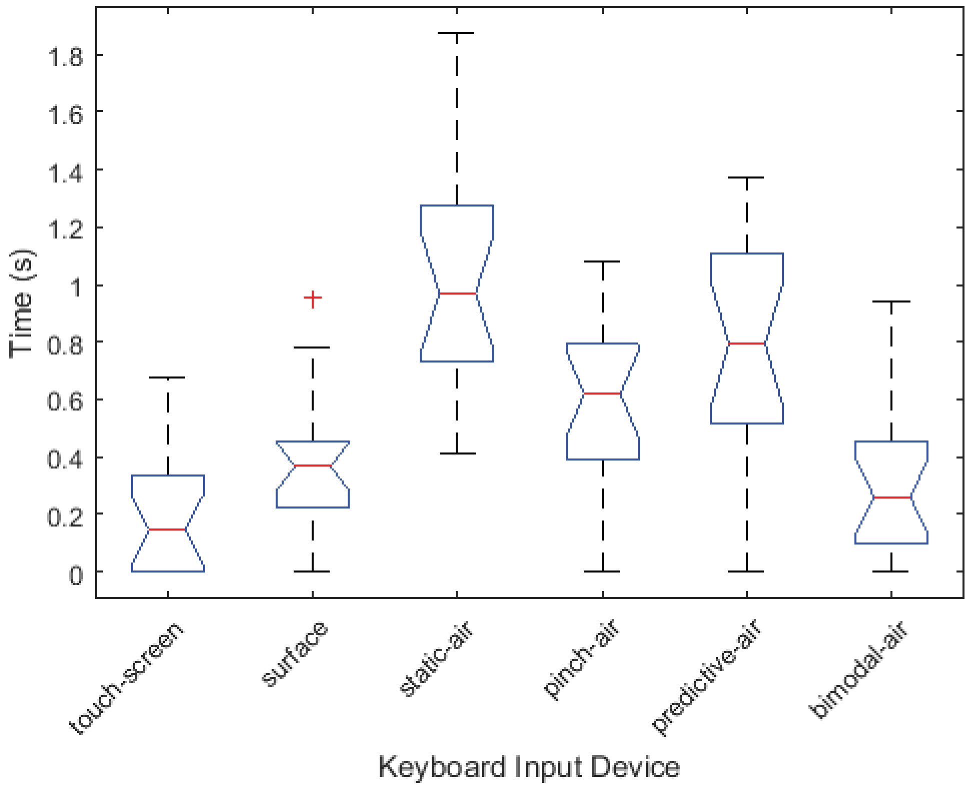


Figure D.19. Median Reaction Time to Errors for each keyboard showing the 25th and 75th percentiles.

Table D.19. Tukey's Honest Significant Difference with multiple comparison for Reaction Time to Errors.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-0.4967	-0.1824	0.1318	0.5441
touchscreen	static-air	-1.1210	-0.8068	-0.4925	0.0000
touchscreen	pinch-air	-0.6899	-0.3756	-0.0614	0.0096
touchscreen	predictive-air	-0.8725	-0.5582	-0.2440	0.0000
touchscreen	bimodal-air	-0.3874	-0.0731	0.2411	0.9842
surface	static-air	-0.9386	-0.6244	-0.3101	0.0000
surface	pinch-air	-0.5075	-0.1932	0.1210	0.4792
surface	predictive-air	-0.6901	-0.3758	-0.0616	0.0096
surface	bimodal-air	-0.2050	0.1093	0.4235	0.9136
static-air	pinch-air	0.1169	0.4311	0.7454	0.0017
static-air	predictive-air	-0.0657	0.2486	0.5628	0.2047
static-air	bimodal-air	0.4194	0.7336	1.0479	0.0000
pinch-air	predictive-air	-0.4968	-0.1826	0.1317	0.5431
pinch-air	bimodal-air	-0.0117	0.3025	0.6168	0.0662
predictive-air	bimodal-air	0.1708	0.4851	0.7993	0.0003

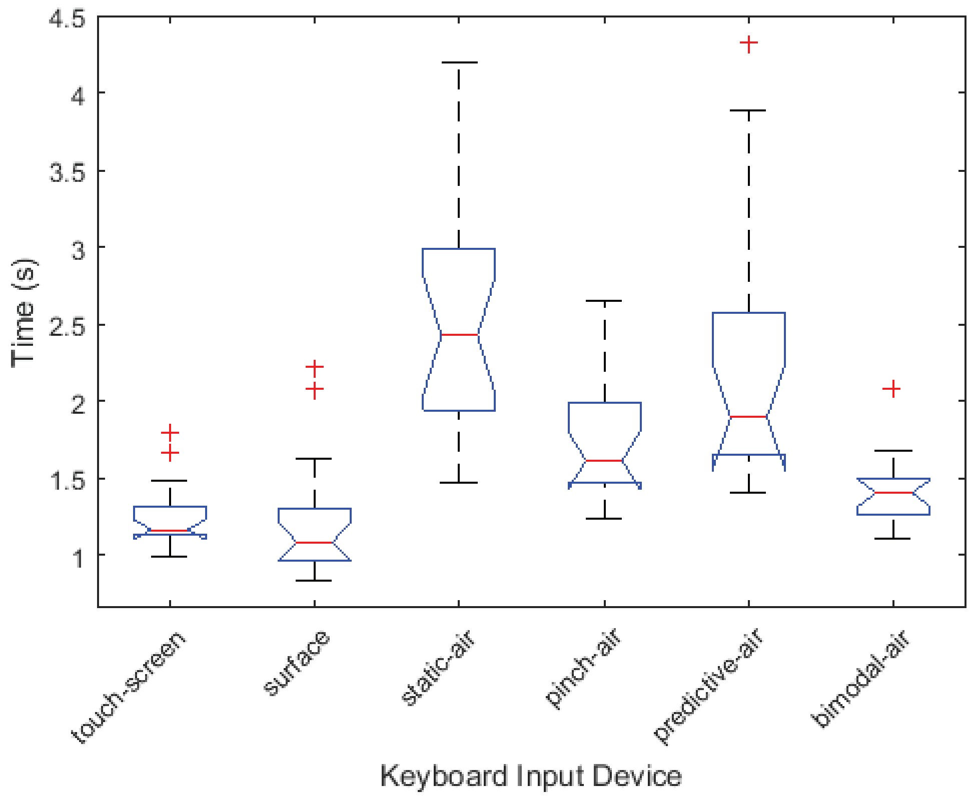


Figure D.20. Median Reaction Time to Simulate Touch for each keyboard showing the 25th and 75th percentiles.

Table D.20. Tukey's Honest Significant Difference with multiple comparison for Reaction Time to Simulate Touch.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-0.5034	0.0211	0.5456	1.0000
touchscreen	static-air	-1.8777	-1.3532	-0.8287	0.0000
touchscreen	pinch-air	-0.9956	-0.4711	0.0533	0.1044
touchscreen	predictive-air	-1.5133	-0.9888	-0.4643	0.0000
touchscreen	bimodal-air	-0.6888	-0.1644	0.3601	0.9431
surface	static-air	-1.8988	-1.3743	-0.8498	0.0000
surface	pinch-air	-1.0167	-0.4922	0.0323	0.0789
surface	predictive-air	-1.5344	-1.0099	-0.4854	0.0000
surface	bimodal-air	-0.7099	-0.1854	0.3391	0.9079
static-air	pinch-air	0.3576	0.8821	1.4066	0.0001
static-air	predictive-air	-0.1601	0.3644	0.8889	0.3395
static-air	bimodal-air	0.6644	1.1889	1.7134	0.0000
pinch-air	predictive-air	-1.0422	-0.5177	0.0068	0.0552
pinch-air	bimodal-air	-0.2177	0.3068	0.8313	0.5356
predictive-air	bimodal-air	0.3000	0.8245	1.3490	0.0002

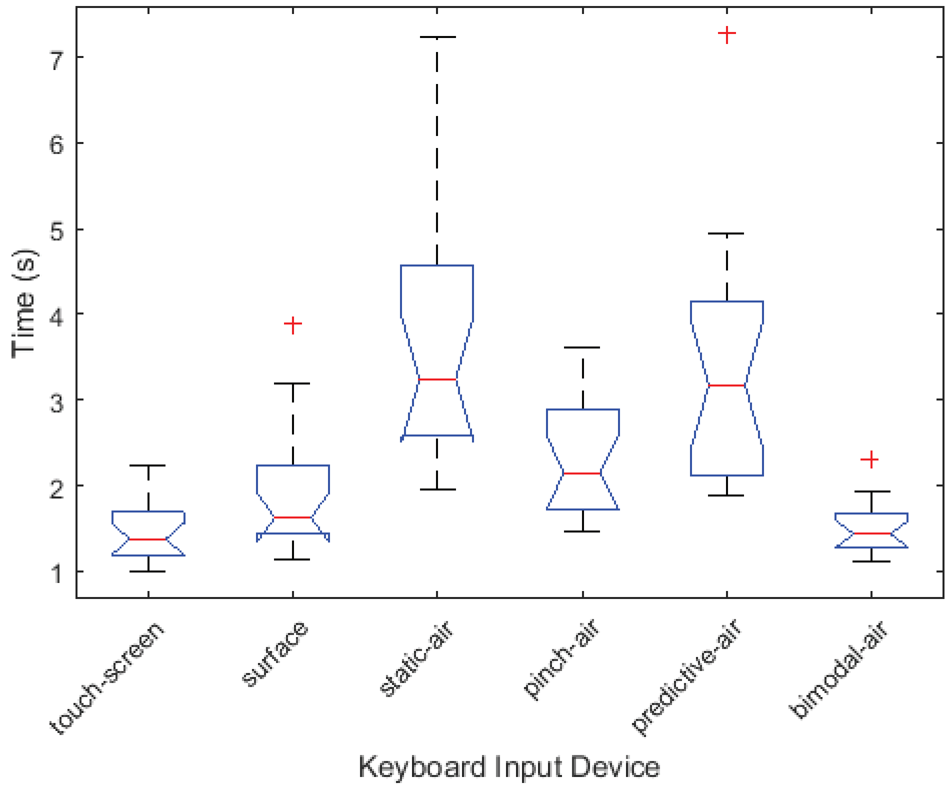


Figure D.21. Median Reaction Time for First Correct Letter for each keyboard showing the 25th and 75th percentiles.

Table D.21. Tukey's Honest Significant Difference with multiple comparison for Reaction Time for First Correct Letter.

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience Interval Upper End	p-value
touchscreen	surface	-1.4101	-0.4882	0.4337	0.6406
touchscreen	static-air	-3.1678	-2.2460	-1.3241	0.0000
touchscreen	pinch-air	-1.8268	-0.9050	0.0169	0.0575
touchscreen	predictive-air	-2.8838	-1.9620	-1.0401	0.0000
touchscreen	bimodal-air	-0.9788	-0.0569	0.8649	1.0000
surface	static-air	-2.6796	-1.7578	-0.8359	0.0000
surface	pinch-air	-1.3386	-0.4168	0.5051	0.7772
surface	predictive-air	-2.3956	-1.4738	-0.5519	0.0001
surface	bimodal-air	-0.4906	0.4313	1.3531	0.7512
static-air	pinch-air	0.4192	1.3410	2.2629	0.0007
static-air	predictive-air	-0.6379	0.2840	1.2059	0.9469
static-air	bimodal-air	1.2672	2.1891	3.1109	0.0000
pinch-air	predictive-air	-1.9789	-1.0570	-0.1352	0.0149
pinch-air	bimodal-air	-0.0738	0.8480	1.7699	0.0899
predictive-air	bimodal-air	0.9832	1.9051	2.8269	0.0000

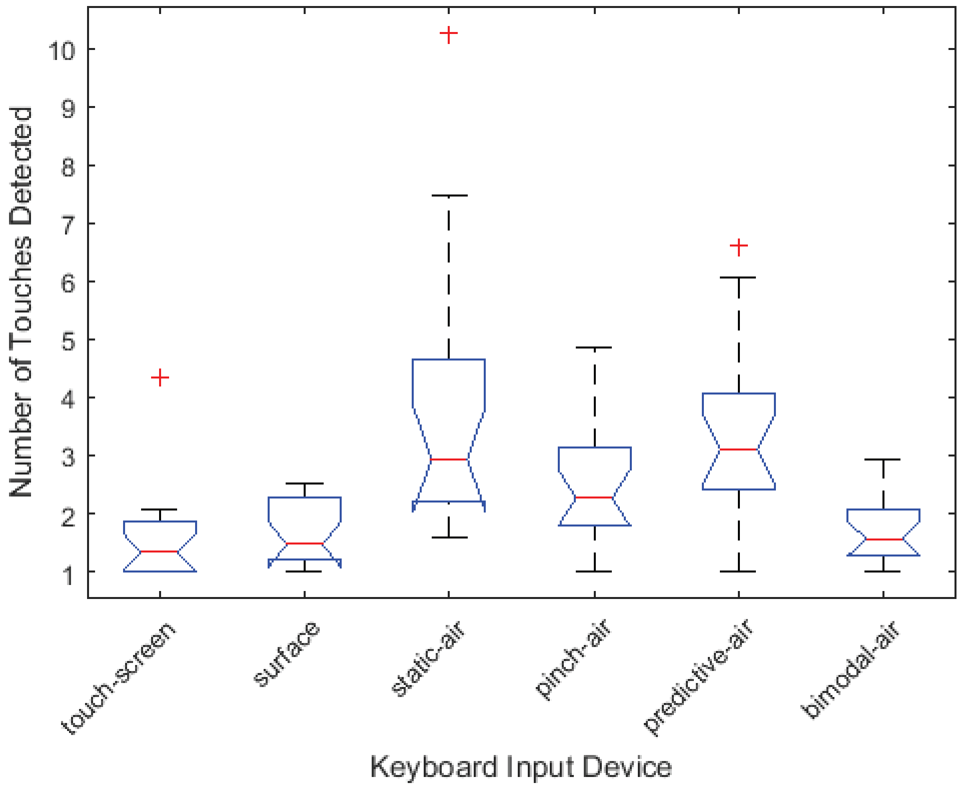


Figure D.22. Median Number of Touches Simulated for each keyboard showing the 25th and 75th percentiles.

Table D.22. Tukey's Honest Significant Difference with multiple comparison for Number of Touches Simulated.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-1.2824	-0.0444	1.1935	1.0000
touchscreen	static-air	-3.4306	-2.1926	-0.9546	0.0000
touchscreen	pinch-air	-2.0824	-0.8444	0.3935	0.3602
touchscreen	predictive-air	-2.9861	-1.7481	-0.5102	0.0011
touchscreen	bimodal-air	-1.3269	-0.0889	1.1491	0.9999
surface	static-air	-3.3861	-2.1481	-0.9102	0.0000
surface	pinch-air	-2.0380	-0.8000	0.4380	0.4222
surface	predictive-air	-2.9417	-1.7037	-0.4657	0.0017
surface	bimodal-air	-1.2824	-0.0444	1.1935	1.0000
static-air	pinch-air	0.1102	1.3481	2.5861	0.0245
static-air	predictive-air	-0.7935	0.4444	1.6824	0.9022
static-air	bimodal-air	0.8657	2.1037	3.3417	0.0000
pinch-air	predictive-air	-2.1417	-0.9037	0.3343	0.2852
pinch-air	bimodal-air	-0.4824	0.7556	1.9935	0.4878
predictive-air	bimodal-air	0.4213	1.6593	2.8972	0.0024

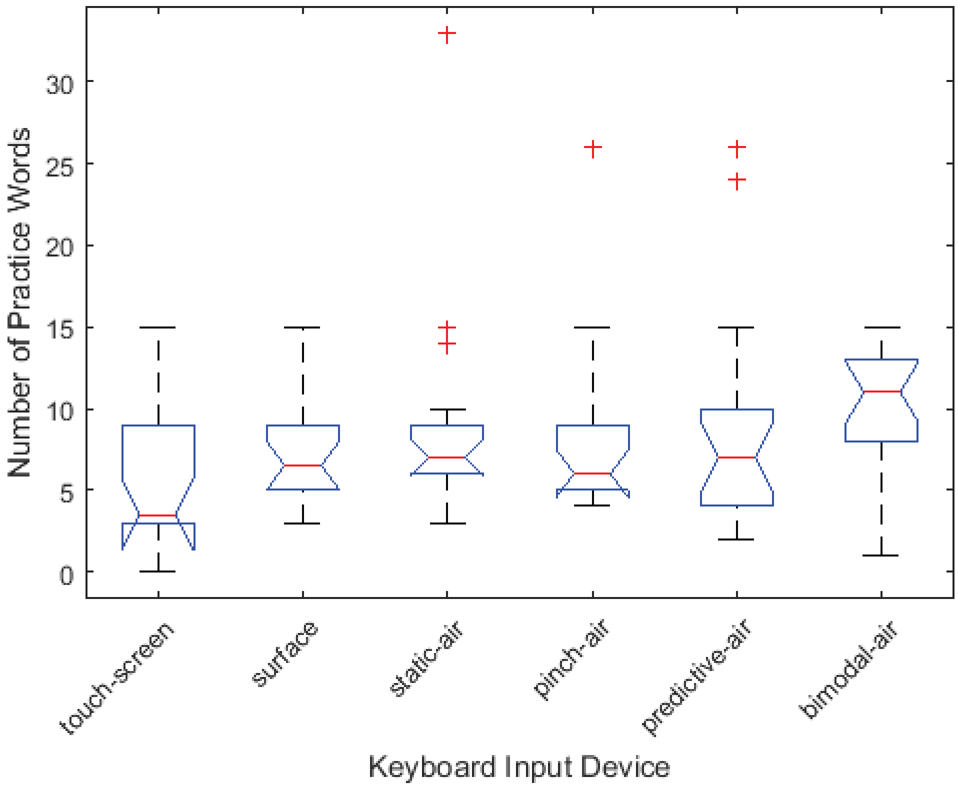


Figure D.23. Median Number of Practice Words for each keyboard showing the 25th and 75th percentiles.

Table D.23. Tukey's Honest Significant Difference with multiple comparison for Level of Discomfort.

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence Interval Upper End	p-value
touchscreen	surface	-1.0795	-0.1667	0.7462	0.9948
touchscreen	static-air	-2.3573	-1.4444	-0.5316	0.0002
touchscreen	pinch-air	-1.8573	-0.9444	-0.0316	0.0381
touchscreen	predictive-air	-2.2462	-1.3333	-0.4205	0.0007
touchscreen	bimodal-air	-1.3573	-0.4444	0.4684	0.7184
surface	static-air	-2.1906	-1.2778	-0.3649	0.0013
surface	pinch-air	-1.6906	-0.7778	0.1351	0.1414
surface	predictive-air	-2.0795	-1.1667	-0.2538	0.0044
surface	bimodal-air	-1.1906	-0.2778	0.6351	0.9496
static-air	pinch-air	-0.4128	0.5000	1.4128	0.6063
static-air	predictive-air	-0.8017	0.1111	1.0239	0.9993
static-air	bimodal-air	0.0872	1.0000	1.9128	0.0232
pinch-air	predictive-air	-1.3017	-0.3889	0.5239	0.8174
pinch-air	bimodal-air	-0.4128	0.5000	1.4128	0.6063
predictive-air	bimodal-air	-0.0239	0.8889	1.8017	0.0610

Table D.24. Tukey's Honest Significant Difference with multiple comparison for Level of Fatigue.

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience Interval Upper End	p-value
touchscreen	surface	-1.4179	-0.3333	0.7513	0.9475
touchscreen	static-air	-3.4735	-2.3889	-1.3043	0.0000
touchscreen	pinch-air	-2.6402	-1.5556	-0.4709	0.0009
touchscreen	predictive-air	-2.5846	-1.5000	-0.4154	0.0015
touchscreen	bimodal-air	-2.0846	-1.0000	0.0846	0.0886
surface	static-air	-3.1402	-2.0556	-0.9709	0.0000
surface	pinch-air	-2.3068	-1.2222	-0.1376	0.0177
surface	predictive-air	-2.2513	-1.1667	-0.0821	0.0273
surface	bimodal-air	-1.7513	-0.6667	0.4179	0.4797
static-air	pinch-air	-0.2513	0.8333	1.9179	0.2326
static-air	predictive-air	-0.1957	0.8889	1.9735	0.1729
static-air	bimodal-air	0.3043	1.3889	2.4735	0.0043
pinch-air	predictive-air	-1.0291	0.0556	1.1402	1.0000
pinch-air	bimodal-air	-0.5291	0.5556	1.6402	0.6728
predictive-air	bimodal-air	-0.5846	0.5000	1.5846	0.7626

Table D.25. Tukey's Honest Significant Difference with multiple comparison for Level of Difficulty.

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience Interval Upper End	p-value
touchscreen	surface	-1.2306	-0.2778	0.6750	0.9578
touchscreen	static-air	-2.4528	-1.5000	-0.5472	0.0002
touchscreen	pinch-air	-2.0083	-1.0556	-0.1028	0.0209
touchscreen	predictive-air	-2.0083	-1.0556	-0.1028	0.0209
touchscreen	bimodal-air	-1.3972	-0.4444	0.5083	0.7535
surface	static-air	-2.1750	-1.2222	-0.2694	0.0042
surface	pinch-air	-1.7306	-0.7778	0.1750	0.1763
surface	predictive-air	-1.7306	-0.7778	0.1750	0.1763
surface	bimodal-air	-1.1195	-0.1667	0.7861	0.9958
static-air	pinch-air	-0.5083	0.4444	1.3972	0.7535
static-air	predictive-air	-0.5083	0.4444	1.3972	0.7535
static-air	bimodal-air	0.1028	1.0556	2.0083	0.0209
pinch-air	predictive-air	-0.9528	0.0000	0.9528	1.0000
pinch-air	bimodal-air	-0.3417	0.6111	1.5639	0.4308
predictive-air	bimodal-air	-0.3417	0.6111	1.5639	0.4308

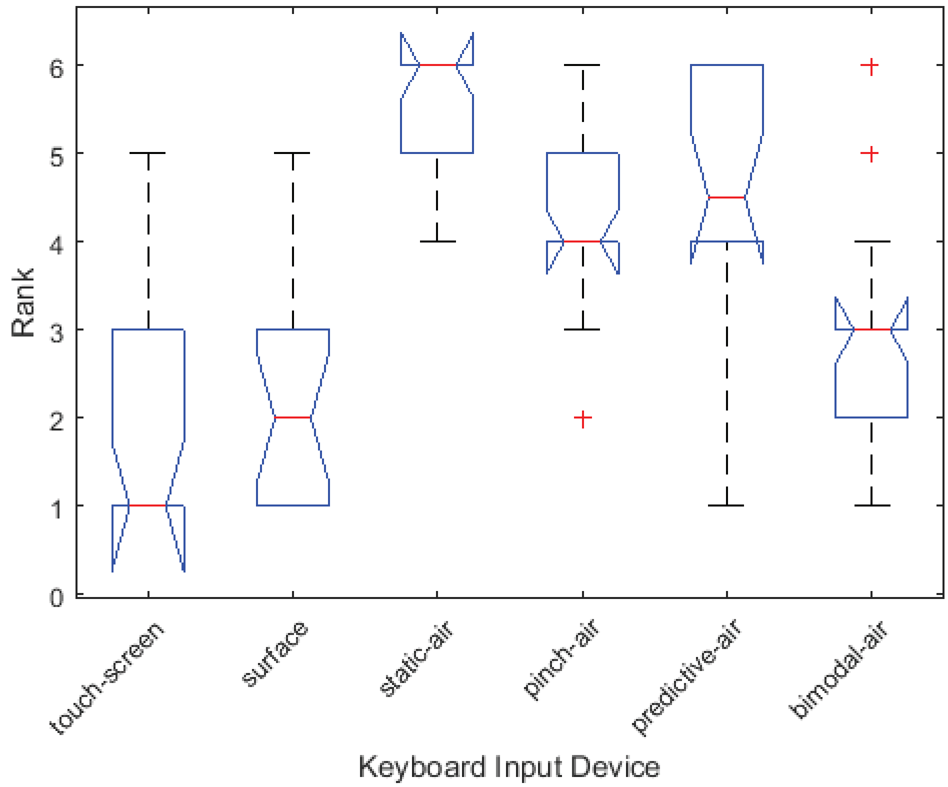


Figure D.24. Median Preference Ranking for each keyboard showing the 25th and 75th percentiles.

Table D.26. Tukey's Honest Significant Difference with multiple comparison for Preference Ranking.

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience Interval Upper End	p-value
touchscreen	surface	-2.1104	-0.3333	1.4438	0.9948
touchscreen	static-air	-5.2215	-3.4444	-1.6673	0.0000
touchscreen	pinch-air	-4.1104	-2.3333	-0.5562	0.0025
touchscreen	predictive-air	-4.3882	-2.6111	-0.8340	0.0004
touchscreen	bimodal-air	-2.7215	-0.9444	0.8327	0.6549
surface	static-air	-4.8882	-3.1111	-1.3340	0.0000
surface	pinch-air	-3.7771	-2.0000	-0.2229	0.0169
surface	predictive-air	-4.0549	-2.2778	-0.5007	0.0035
surface	bimodal-air	-2.3882	-0.6111	1.1660	0.9244
static-air	pinch-air	-0.6660	1.1111	2.8882	0.4776
static-air	predictive-air	-0.9438	0.8333	2.6104	0.7648
static-air	bimodal-air	0.7229	2.5000	4.2771	0.0009
pinch-air	predictive-air	-2.0549	-0.2778	1.4993	0.9978
pinch-air	bimodal-air	-0.3882	1.3889	3.1660	0.2251
predictive-air	bimodal-air	-0.1104	1.6667	3.4438	0.0808

Amma, C., M. Georgi, and T. Schultz (2012). Airwriting: Hands-free mobile text input by spotting and continuous recognition of 3d-space handwriting with inertial sensors. In *Wearable Computers (ISWC), 2012 16th International Symposium*, pp. 52–59. IEEE.

Bachmann, D., F. Weichert, and G. Rinkenauer (2015). Evaluation of the leap motion controller as a new contact-free pointing device. *Sensors (Basel, Switzerland) 15*(1), 214–233.

Banerjee, A., J. Burstyn, A. Girouard, and R. Vertegaal (2012). Multipoint: Comparing laser and manual pointing as remote input in large display interactions. In *International Journal of Human - Computer Studies*, Volume 70(10) of *CHI '11*, pp. 690–702. Elsevier Ltd.

Bi, X., C. Chelba, T. Ouyang, K. Partridge, and S. Zhai (2012). Bimanual gesture keyboard. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pp. 137–146. ACM.

Castellucci, S. J. and I. S. MacKenzie (2008). Graffiti vs. unistrokes: An empirical comparison. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, New York, NY, USA, pp. 305–308. ACM.

Cockburn, A., P. Quinn, C. Gutwin, G. Ramos, and J. Looser (2011). Air pointing: Design and evaluation of spatial target acquisition with and without visual feedback. In *International Journal of Human-Computer Studies*, Volume 69(6), pp. 401–414. Elsevier Ltd.

Dattalo, P. (2008). *Determining Sample Size: Balancing Power, Precision, and Practicality*. Oxford University Press.

Devoe, D. (1967). Alternatives to handprinting in the manual entry of data. In *IEEE Transactions on Human Factors in Electronics*, Volume HFE-8(1), pp. 21–32. IEEE.

Finn, J. (1997). Aging and information technology: The promise and the challenge. *Generations–Journal of The American Society on Aging 21*(3), 5–6.

Freeman, D., R. Vennelakanti, and S. Madhvanath (2012). Freehand pose-based gestural interaction: Studies and implications for interface design. In *Intelligent Human Computer Interaction (IHCI), 2012 4th International Conference*, pp. 1–6. IEEE.

Fuccella, V., M. De Rosa, and G. Costagliola (2014). Novice and expert performance of keyscretch: A gesture-based text entry method for touch-screens. *IEEE Transactions on Human-Machine Systems 44*(4), 511–523.

Guinness, D. (2015). *Models for Rested Touchless Gestural Interaction*. Masters thesis, Department of Computer Science, Baylor University, Waco, TX.

Guna, J., G. Jakus, M. Pogačnik, S. Tomažič, and J. Sodnik (2014). An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. *Sensors (Basel, Switzerland) 14*(2), 3702–3720.

Har-Peled, S. and B. Raichel (2014, January). The frÉchet distance revisited and extended. *ACM Trans. Algorithms 10*(1), 3:1–3:22.

Hari Haran, A. J. (2014). *Giving the Users a Hand: Towards Touchless Hand Gestures for the Desktop*. Masters thesis, Department of Computer Science, Baylor University, Waco, TX.

Hincapié-Ramos, J. D., X. Guo, and P. Irani (2014). The consumed endurance workbench: A tool to assess arm fatigue during mid-air interactions. In *Proceedings of the 2014 Companion Publication on Designing Interactive Systems*, DIS Companion '14, New York, NY, USA, pp. 109–112. ACM.

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

Hochberg, Y. and A. C. Tamhane (2009). *Multiple Comparison Procedures*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Hollander, M., D. A. Wolfe, and E. Chicken (2013). *Nonparametric Statistical Methods* (3 ed.). Hoboken, New Jersey: John Wiley & Sons, Inc.

Jones, B., R. Sodhi, D. Forsyth, B. Bailey, and G. Maciocci (2012). Around device interaction for multiscale navigation. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '12, pp. 83–92. ACM.

Jones, E., J. Alexander, A. Andreou, P. Irani, and S. Subramanian (2010). Gestext: Accelerometer-based gestural text-entry systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, New York, NY, USA, pp. 2173–2182. ACM.

Jota, R., M. A. Nacenta, J. A. Jorge, S. Carpendale, and S. Greenberg (2010). A comparison of ray pointing techniques for very large displays. In *Proceedings - Graphics Interface*, pp. 269–276. Canadian Information Processing Society.

Kim, J. (2010). Within-subjects design. See Salkind (2010).

Kristensson, P.-O. (2007). *Discrete and Continuous Shape Writing for Text Entry and Control*. Doctoral dissertation, Linkping University, Sweden.

Kristensson, P.-O., T. Nicholson, and A. Quigley (2012). Continuous recognition of one-handed and two-handed gestures using 3d full-body motion tracking sensors. In *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces*, IUI '12, New York, NY, USA, pp. 89–92. ACM.

MacKenzie, I. S. and R. W. Soukoreff (2002). A character-level error analysis technique for evaluating text entry methods. In *Proceedings of the Second Nordic Conference on Human-computer Interaction*, NordiCHI '02, New York, NY, USA, pp. 243–246. ACM.

MacKenzie, I. S. and R. W. Soukoreff (2003). Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, New York, NY, USA, pp. 754–755. ACM.

MacKenzie, I. S. and K. Tanaka-Ishii (2007). *Text Entry Systems: Mobility, Accessibility, Universality*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Malik, S. and J. Laszlo (2004). Visual touchpad: A two-handed gestural input device. In *Proceedings of the 6th International Conference on Multimodal Interfaces*, ICMI '04, pp. 289–296. ACM.

Mansson, R. and P. Prescott (2001, August). Missing values in replicated latin squares. *Journal of Applied Statistics 28*(6), 743–757.

Markussen, A., M. R. Jakobsen, and K. Hornbæk (2013). Selection-based mid-air text entry on large displays. In *Human-Computer Interaction INTERACT 2013*, Volume 8117(1) of *Lecture Notes in Computer Science*, pp. 401–418. Springer Berlin Heidelberg.

Markussen, A., M. R. Jakobsen, and K. Hornbæk (2014). Vulture: A mid-air word-gesture keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pp. 1073–1082. ACM.

Motion, Leap, I. (2012). Leap motion controller. `https://www.leapmotion.com/` `[lastaccessed:2015-10-10]`.

Nancel, M., J. Wagner, E. Pietriga, O. Chapuis, and W. Mackay (2011). Mid-air pan-and-zoom on wall-sized displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 177–186. ACM.

Neter, J., W. Wasserman, and M. H. Kutner (1996). *Applied Linear Statistical Models: Regression, Analysis of Variance, and Experimental Designs* (4 ed.). Irwin Press.

Ni, T., D. Bowman, and C. North (2011). Airstroke: Bringing unistroke text entry to freehand gesture interfaces. In *Proceedings of the SIGCHI Conference on human factors in computing systems*, CHI '11, pp. 2473–2476. ACM.

Piumsomboon, T., A. Clark, M. Billinghurst, and A. Cockburn (2013). User-defined gestures for augmented reality. In *Human-Computer Interaction INTERACT 2013*, Volume 8118(2) of *Lecture Notes in Computer Science*, pp. 282–299. Springer Berlin Heidelberg.

Proctor, W. R. and L. K.-P. Vu (2006). *Stimulus-Response Compatibility Principles: Data, Theory, and Application*. Boca Raton, FL, USA: Taylor & Francis Group.

Salkind, N. J. (2010). *Encyclopedia of Research Design*. Thousand Oaks, California: SAGE Publications, Inc.

Schick, A., D. Morlock, C. Amma, T. Schultz, and R. Stiefelhagen (2012). Vision-based handwriting recognition for unrestricted text input in mid-air. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, ICMI '12, New York, NY, USA, pp. 217–220. ACM.

Shengli, Z., D. Zhuxin, J. W. Li, and K. P. Chung (2008). Hand-written character recognition using mems motion sensing technology. In *Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference*, pp. 1418–1423. IEEE.

Shoemaker, G., L. Findlater, J. Q. Dawson, and K. S. Booth (2009). Mid-air text input techniques for very large wall displays. In *Proceedings - Graphics Interface*, pp. 231–238. Canadian Information Processing Society.

Shoemaker, G., A. Tang, and K. S. Booth (2007). Shadow reaching: A new perspective on interaction for large displays. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pp. 53–56. ACM.

Soukoreff, R. W. and I. S. MacKenzie (2003). Metrics for text entry research: An evaluation of msd and kspc, and a new unified error metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, New York, NY, USA, pp. 113–120. ACM.

Sutton, J. (2013). Air painting with corel painter freestyle and the leap motion controller: A revolutionary new way to paint! In *ACM SIGGRAPH 2013 Studio Talks*, SIGGRAPH '13, pp. 21:1–21:1. ACM.

Taveira, A. D. and S. D. Choi (2009). Review study of computer input devices and older users. *International Journal of Human-Computer Interaction 25*(5), 455–474.

Teixeira, V. (2011). *Improving elderly access to audiovisual and social media, using a multimodal human-computer interface*. Doctoral dissertation, Faculdade de Engenharia, Universidade do Porto.

Toimil, M., N. Shin, and A. Eisner (2013). Gesture technology: An innovative leap for workflow. In *Proceedings of the International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government*, EEE '13, pp. 291. WorldComp.

Triesch, J. and C. Von Der Malsburg (1998). Robotic gesture recognition by cue combination. In *Informatik '98*, pp. 223–232. Springer Berlin Heidelberg.

Vikram, S., L. Li, and S. Russell (2013). Writing and sketching in the air, recognizing and controlling on the fly. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pp. 1179–1184. ACM.

Vogel, D. and R. Balakrishnan (2005). Distant freehand pointing and clicking on very large, high resolution displays. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, pp. 33–42. ACM.

Von Hardenberg, C. and F. Bérard (2001). Bare-hand human-computer interaction. In *Proceedings of the 2001 Workshop on Perceptive User Interfaces*, PUI '01, New York, NY, USA, pp. 1–8. ACM.

Wachs, J. P., M. Kölsch, H. Stern, and Y. Edan (2011, February). Vision-based hand-gesture applications. *Commun. ACM 54*(2), 60–71.

Weichert, F., D. Bachmann, B. Rudak, and D. Fisseler (2013). Analysis of the accuracy and robustness of the leap motion controller. *Sensors (Basel, Switzerland) 13*(5), 6380–6393.

Witt, H., M. Lawo, and M. Drugge (2008). Visual feedback and different frames of reference: The impact on gesture interaction techniques for wearable computing. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '08, New York, NY, USA, pp. 293–300. ACM.

Wu, C. F. J. and M. S. Hamada (2009). *Experiments: Planning, Analysis, and Optimization*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Xie, Y., R. Xu, X. Guo, E. Zhu, and J. Wu (2013, October). Natural bare-hand interaction for remote operating large touch screen. *Journal of Computers 8*(10), 2512–2519.

Yoo, J., S. Lee, and C. Ahn (2012). Air hook: Data preloading user interface. In *ICT Convergence (ICTC), 2012 International Conference*, pp. 163–167. IEEE.

Yu, W., R. Kato, F. Fabio, H. Yokoi, and Y. Kakazu (2003). An emg keyboard for forearm amputees. *Applied Bionics and Biomechanics 1*(1), 33–43.

Zhai, S. and P.-O. Kristensson (2003). Shorthand writing on stylus keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, New York, NY, USA, pp. 97–104. ACM.

Zhai, S. and P.-O. Kristensson (2004). Shark2: A large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, pp. 43–52. ACM.

Zhai, S. and P. O. Kristensson (2012). The word-gesture keyboard: Reimagining keyboard interaction. *Communications of the ACM 55*(9), 91–101.

Zhai, S., P.-O. Kristensson, P. Gong, M. Greiner, S. A. Peng, L. M. Liu, and A. Dunnigan (2009). Shapewriter on the iphone: From the laboratory to the real world. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pp. 2667–2670. ACM.

Zhai, S., A. Sue, and J. Accot (2002). Movement model, hits distribution and learning in virtual keyboarding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, New York, NY, USA, pp. 17–24. ACM.

Zhang, H. (2010). Latin square design. See Salkind (2010).