ABSTRACT

Alternatives using the Leap Motion to extend Mid-Air Word-Gesture Keyboards

Garrett Benoit, M.S.

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

Lately, the use of touchless, mid-air, gesture-based interactions has increased significantly due to the popularity of augmented and virtual reality and advances in other industries (e.g., medicine, gaming), and with this wide-spread application comes the need for effecient, mid-air text-entry. Word-gesture keyboards have garnered attention in recent years, now coming standard on most Android devices, offering efficient means of gesture-based text-entry. For the first time, Markussen et al. combined the two with the inception of Vulture (Markussen, Jakobsen, and Hornbæk 2014), the first mid-air, word-gesture keyboard, providing the fastest means of mid-air text-entry yet. This thesis builds on the findings of Markussen et al. and presents alternatives means for word separation in mid-air text-entry for word-gesture keyboards, exploring and identifying the problems of new techniques and presenting possible solutions. Of the new techniques, a bimodal approach shows great promise, reaching a mean text-entry rate of 15.8 Words Per Minutes for a single session with no training.

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

Michael G. Poor, Ph.D., Chairperson

Bill Poucher, Ph.D.

Brian Garner, Ph.D.

Accepted by the Graduate School
October 2015

J. Larry Lyon, Ph.D., Dean

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

TABLE OF CONTENTS

LIST OF FIGURES x

LIST OF TABLES xiii

ACKNOWLEDGMENTS xv

DEDICATION xvi

1 Introduction . . . 1

1.1 Motivation . . . 1

1.2 Background . . . 2

1.3 Challenges . . . 4

1.3.1 Word Separation . . . 4

1.3.2 Motor Space vs Display Space . . . 5

1.3.3 Fatigue . . . 5

1.4 Solutions . . . 6

1.4.1 Word Separation . . . 6

1.4.2 Motor Space vs Display Space . . . 6

1.4.3 Fatigue . . . 7

2 Literature Review . . . 8

2.1 Mid-Air Pointing . . . 8

2.2 Barehanded Interaction . . . 9

2.3 Mid-Air Text-Entry . . . 9

2.3.1 Selection-Based . . . 9

2.3.2 Gesture-Based . . . 10

2.3.3 Word-Gesture Keyboards 11

2.4 Word-Gesture Keyboards 11

2.5 Text-Entry Evaluation 12

2.5.1 Deviating from the Standard Phrases 12

2.5.2 Text-Entry Rates 13

2.5.3 Error Rates 13

2.5.4 Correctness 13

3 Keyboard Design 15

3.1 Word-Gesture Keyboard Implementation 15

3.1.1 Lacking Word-Recognition 15

3.1.2 Design 15

3.1.3 Display 16

3.1.4 Calibration 19

3.1.5 Dictionary Creation 20

3.2 Word-Gesture Keyboards 23

3.2.1 Touch Screen Keyboard 23

3.2.2 Leap Surface Keyboard 25

3.2.3 Leap Static-Air Keyboard 26

3.2.4 Leap Predictive-Air Keyboard 27

3.2.5 Leap Bimodal-Air Keyboard 28

3.2.6 Leap Pinch-Air Keyboard 30

4 Methodology 32

4.1 Pre-Pilot Study 32

4.1.1 Participants 32

4.1.2 Input Devices and Interaction 32
4.1.3 Task Design 34
4.1.4 Experimental Design 35
4.1.5 Procedure 35
4.1.6 Dependent Measures 37
4.2 Pilot Study 37
4.2.1 Participants 37
4.2.2 Input Devices and Interaction 38
4.2.3 Task Design 39
4.2.4 Experimental Design 41
4.2.5 Procedure 42
4.2.6 Dependent Measures 43
4.3 Final Study 48
4.3.1 Participants 49
4.3.2 Input Devices and Interaction 50
4.3.3 Task Design 52
4.3.4 Experimental Design 53
4.3.5 Procedure 54
4.3.6 Dependent Measures 56

5 Results and Analysis 60
5.1 Text-Entry Rate 60
5.1.1 Text-Entry Rate Modified-Shortest 62
5.2 Error Rate 64
5.2.1 Minimum Word Distance 64
5.2.2 Keystrokes Per Character 68

5.2.3 Minimum String Distance . . . 72
5.2.4 Total Error Rate . . . 73
5.3 Correctness . . . 76
5.3.1 Fréchet Distance . . . 76
5.4 Distance Measures . . . 82
5.4.1 Word-Gesture Distance . . . 82
5.4.2 Word-Gesture Velocity . . . 85
5.4.3 Hand Velocity . . . 86
5.5 Timing Measures . . . 87
5.5.1 Word-Gesture Duration . . . 87
5.5.2 Reaction Time to Errors . . . 90
5.5.3 Reaction Time to Simulate Touch . . . 92
5.5.4 Reaction Time for First Correct Letter . . . 93
5.6 Quantitative Measures . . . 95
5.6.1 Number of Touches Simulated . . . 95
5.6.2 Number of Practice Words . . . 97
5.7 Qualitative Measures . . . 98
5.7.1 Level of Discomfort . . . 98
5.7.2 Level of Fatigue . . . 100
5.7.3 Level of Difficulty . . . 100
5.7.4 Preference Ranking . . . 102
6 Discussion, Future Work and Conclusion . . . 105
6.1 Discussion . . . 105
6.1.1 Pinching Interaction . . . 106
6.1.2 3-Dimensional Interaction . . . 106

6.1.3 Bimodal Interaction 107
6.1.4 Lacking Word-Recognition 108
6.2 Future Work 109
6.2.1 Implement Word-Recognition 109
6.2.2 Redesign Task 109
6.2.3 Augmented Reality 109
6.2.4 The "Skimming" Issue 110
6.2.5 Better Tracking 110
6.2.6 Alternative Interaction Plane 111
6.2.7 Multi-functional Space 112
6.2.8 Image Processing 113
6.2.9 Gaming Console Keyboards 113
6.2.10 Accessibility 113
6.3 Conclusion 114

APPENDICES 115

APPENDIX A Surveys 116

APPENDIX B Results Tables and Boxplots 118
B.1 Text-Entry Rate 119
B.1.1 Text-Entry Rate Modified-Shortest 121
B.2 Error Rates 123
B.2.1 Minimum Word Distance 123
B.2.2 Keystrokes Per Character 127
B.2.3 Minimum String Distance 131
B.2.4 Total Error Rate 133

B.3 Correctness 137
B.3.1 Fréchet Distance 137
B.4 Distance Measures 143
B.4.1 Word-Gesture Distance 143
B.4.2 Word-Gesture Velocity 147
B.4.3 Hand Velocity 149
B.5 Timing Measures 151
B.5.1 Word-Gesture Duration 151
B.5.2 Reaction Time to Errors 155
B.5.3 Reaction Time to Simulate Touch 157
B.5.4 Reaction Time for First Correct Letter 159
B.6 Quantitative Measures 161
B.6.1 Number of Touches Simulated 161
B.6.2 Number of Practice Words 163
B.7 Qualitative Measures 164
B.7.1 Level of Discomfort 164
B.7.2 Level of Fatigue 165
B.7.3 Level of Difficulty 166
B.7.4 Preference Ranking 167

BIBLIOGRAPHY 169

LIST OF FIGURES

3.1 Larger Key Example 16
3.2 Protected Path Example 17
3.3 Display: Keyboard Layout 18
3.4 Display: Text Area 18
3.5 Display: Text Area 19
3.6 Calibration 20
3.7 Angled Plane 21
3.8 Motor Space Comparison 22
3.9 Touch Screen Word Separation 23
3.10 Ideum Multi-touch Table Platform 24
3.11 Leap Surface Holder 25
3.12 Leap Surface Word Separation 26
3.13 Leap Static-Air Word Separation 27
3.14 Leap Bimodal-Air Word-Separation 29
3.15 Leap Pinch-Air Word Separation 30

4.1 Fréchet Distance code snippet 47

5.1 Mean Text-Entry Rates 61
5.2 Mean Text-Entry Rates for Modified-Shortest 63
5.3 Mean Minimum Word Distance 65
5.4 Mean Minimum Word Distance for Modified-Shortest 67
5.5 Arcing Motion 68
5.6 Mean Keystrokes Per Character 69
5.7 Mean Keystrokes Per Character for Modified-Shortest 71

5.8 Mean Minimum String Distance for Modified-Shortest 73
5.9 Mean Total Error Rates 74
5.10 Mean Total Error Rate for Modified-Shortest 76
5.11 Mean Fréchet Distance 77
5.12 Mean Fréchet Distance for Modified-Shortest 79
5.13 Mean Fréchet Distance for Modified-Backspace 81
5.14 Mean Word-Gesture Distance 83
5.15 Mean Word-Gesture Distance for Modified-Shortest 84
5.16 Mean Word-Gesture Velocity 85
5.17 Mean Hand Velocity 87
5.18 Mean Word-Gesture Duration 88
5.19 Mean Word-Gesture Dur for Modified-Shortest 90
5.20 Mean Reaction Time to Errors 91
5.21 Mean Reaction Time to Simulate Touch 93
5.22 Mean Reaction Time for First Correct Letter 94
5.23 Mean Number of Touches Simulated 96
5.24 Mean Number of Practice Words 98
5.25 Level of Discomfort Boxplot 99
5.26 Level of Fatigue Boxplot 101
5.27 Level of Difficulty Boxplot 102
5.28 Mean Preference Rankings 103

6.1 Skimming Problem 107
6.2 Skimming Solution 110
6.3 Blocking Problem 111
6.4 Bad Calibration Example 112

A.1 Intermittent Survey 116
A.2 Exit Survey 117

B.1 Text-Entry Rate Boxplot 119
B.2 Text-Entry Rates Boxplot for Modified-Shortest 121
B.3 Minimum Word Distance Boxplot 123
B.4 Minimum Word Distance Boxplot for Modified-Shortest 125
B.5 Keystrokes Per Character Boxplot 127
B.6 Keystrokes Per Character Boxplot for Modified-Shortest 129
B.7 Minimum String Distance Boxplot for Modified-Shortest 131
B.8 Total Error Rate Boxplot 133
B.9 Total Error Rate Boxplot for Modified-Shortest 135
B.10 Fréchet Distance Boxplot 137
B.11 Fréchet Distance Boxplot for Modified-Shortest 139
B.12 Fréchet Distance Boxplot for Modified-Backspace 141
B.13 Word-Gesture Distance Boxplot 143
B.14 Word-Gesture Distance Boxplot for Modified-Shortest 145
B.15 Word-Gesture Velocity Boxplot 147
B.16 Hand Velocity Boxplot 149
B.17 Word-Gesture Duration Boxplot 151
B.18 Word-Gesture Duration Boxplot for Modified-Shortest 153
B.19 Reaction Time to Errors Boxplot 155
B.20 Reaction Time to Simulate Touch Boxplot 157
B.21 Reaction Time for First Correct Letter Boxplot 159
B.22 Number of Touches Simulated Boxplot 161
B.23 Number of Practice Words Boxplot 163
B.24 Preference Ranking Boxplot 167

LIST OF TABLES

4.1 Pre-pilot Study Details of Participants 32

4.2 Pre-pilot Study Schedule of Assessments 37

4.3 Pilot Study Details of Participants 38

4.4 Latin Squares Example 41

4.5 Pilot Study Schedule of Assessments 43

4.6 Final Study Details of Participants 50

4.7 Replicated Latin Squares Example 54

4.8 Final Study Schedule of Assessments 56

4.9 Shortest-transcribed Examples 56

4.10 Backspace-transcribed Examples 57

B.1 Text-Entry Rate Multiple Comparison 120

B.2 Text-Entry Rate Multiple Comparison Modified-Shortest 122

B.3 MWD Multiple Comparison 124

B.4 MWD Multiple Comparison Modified-Shortest 126

B.5 KSPC Multiple Comparison 128

B.6 KSPC Multiple Comparison Modified-Shortest 130

B.7 MSD Multiple Comparison Modified-Shortest 132

B.8 Total Error Rate Multiple Comparison 134

B.9 Total Error Rate Multiple Comparison Modified-Shortest 136

B.10 Fréchet Distance Multiple Comparison 138

B.11 Fréchet Distance Multiple Comparison Modified-Shortest 140

B.12 Fréchet Distance Multiple Comparison Modified-Backspace 142

B.13 Word-Gesture Distance Multiple Comparison 144

B.14 Word-Gesture Distance Multiple Comparison Modified-Shortest . . . 146

B.15 Word-Gesture Velocity Multiple Comparison 148
B.16 Hand Velocity Multiple Comparison 150
B.17 Word-Gesture Duration Multiple Comparison 152
B.18 Word-Gesture Duration Multiple Comparison Modified-Shortest . . . 154
B.19 Reaction Time to Errors Multiple Comparison 156
B.20 Reaction Time to Simulate Touch Multiple Comparison 158
B.21 Reaction Time for First Correct Letter Multiple Comparison 160
B.22 Number of Touches Simulated Multiple Comparison 162
B.23 Level of Discomfort Multiple Comparison 164
B.24 Level of Fatigue Multiple Comparison 165
B.25 Level of Difficulty Multiple Comparison 166
B.26 Preference Ranking Multiple Comparison 168

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 helps shaped the direction of this project. Also, to all of those in the HCI Lab who helped with my pilot studies! And to all the volunteers who made this research possible by generously donating their time. Finally, to my sister Heather Benoit who inspired me to go to Baylor and seek a greater future.

To those who pioneer the future.

CHAPTER ONE

Introduction

1.1 Motivation

This research was inspired by my Grandpa, an amputee who has trouble typing efficiently with standard computer keyboards. It is a long and arduous task to type in long emails or messages when using only one hand and typing each character individually. This led me to consider all of the other recent amputee's from the long period of war in the Middle East and how many people have come home with their lives changed. Thinking about these individuals, I thought, "There must be a better way!". A way to make typing and other interactions with computers even simpler and more efficient, not as tedious as they were before. Working in the same space as Alvin Jude Hari Haran and Darren Guiness and seeing their work being done on accessibility with using mid-air gestures to control the mouse using the Leap Motion (Motion 2012, Hari Haran 2014, Guinness 2015), it inspired me to try to put the same technology to use for working with mid-air keyboards. It was evident to me, with all the phones and tablets out today coming standard with word-gesture keyboards and efficient, single-handed text-entry, I was inspired to look for a way to apply these 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. I 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, the first on using word-gesture keyboards in mid-air, and so 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*** we interact with them.

1.2 Background

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. Mid-air, gesture-controlled content has seen it's 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, means of reasonable mid-air text entry are fairly new. Past technologies allowed for mid-air text-entry, but those approaches have fallen short of any meaningful text entry rates (Malik and Laszlo 2004). More modern approaches of mid-air text-entry have seen improved results but still low, around 13 (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 also lacked the multi-tap feature 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. (Markussen, Jakobsen, and Hornbæk 2014) transitioned word-gesture keyboards for mid-air use 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 and repeated measures, indicating 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 (Markussen, Jakobsen, and Hornbæk 2014).

The purpose of this study is to use the Leap Motion (Motion 2012), a new and emerging technology (Sutton 2013, Weichert, Bachmann, Rudak, and Fisseler 2013, Guna, Jakus, Pogačnik, Tomažič, and Sodnik 2014), for interpreting mid-air gestural inputs for text entry (Toimil, Shin, and Eisner 2013). The only previous attempt for text entry with the Leap Motion was in mid-air handwriting (Vikram, Li, and Russell 2013); however, even regular hand-writing is slow, and confined 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 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 intends to use the Leap Motion to find alternatives to mid-air, word-gesture keyboards and find a better approach than 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 will explore the option of using the extra degrees of freedom available in mid-air (e.g., depth) which was decided against by Markussen et al. (Markussen, Jakobsen, and Hornbæk 2014) to point out the problems of solutions of these techniques. This study will also use other techniques of simulating touch for the mid-air equivalent of tapping and releasing for delimiters of words. The idea is to achieve something that is as simple to using a touch-based devices as possible while still allowing for a multi-functional workspace.

The rationale behind this research is to improve mid-air text entry using techniques that will still allow for other gesture-controls when working with gesture-interfaces (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). Touch-less gesture-controllers with mid-air text entry will benefit augmented reality (e.g., Google Glass, Microsoft HoloLens) and virtual reality (e.g., Oculus Rift) as well as benefit the medical world (e.g., operating rooms) when it comes to sanitation and sterile environments to reduce the spread of pathogens and to aid in efficient text-entry for those with disabilities.

1.3 Challenges

1.3.1 Word Separation

When it comes to mid-air text-entry, one of the greatest challenges is finding a suitable means of distinguishing between separate inputs 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 limitations with these are that selection-based techniques and using hand-gestures to interpret letters for text-entry are slow with limited text-entry speeds (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. (Markussen, Jakobsen, and Hornbæk 2014).

Markussen et al. used pinching as a means of separating between words using a glove with reflective markers and a large projected display (Markussen, Jakobsen, and Hornbæk 2014). 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 by Vulture lacks adaptability between users, adding complexity and limiting word separation. In addition, using a glove with reflective markers, the glove's required tracking equipment, and using a large projector display is very inconvenient for the typical user and not something that can be easily utilized. The experience needs to be confined down to something that can be used easily with a desktop or laptop computer, and be variable enough to be used by those who do not have the means to pinch.

1.3.2 Motor Space vs Display Space

Another factor contributing to the complexity of mid-air text-entry is the fact that we have to deal with the motor space and display space being decoupled when working in mid-air (Markussen, Jakobsen, and Hornbæk 2014). According to Markussen et al., 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 however still only reached 59% the text-entry rate of touch-based inputs.

1.3.3 Fatigue

The Last thing to consider when working with mid-air is how fatiguing these gestures can be to produce over long periods of time. Even in Vulture, the participant is required to stand in front of the display, holding their arms out to interact

(Markussen, Jakobsen, and Hornbæk 2014). This sort of posture while performing gestures is known to cause fatigue, known as the "Gorilla Arm Syndrom" (Yoo, Lee, and Ahn 2012, Teixeira 2011), and should be minimized for text-entry.

1.4 Solutions

1.4.1 Word Separation

To attempt to solve the issues faced by complex word separation, we will use the Leap Motion Controller to explore different means of word separation. The Leap Motion Controller is beneficial in that it allows 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 can be easily obtained by ordinary users and runs on typical desktop computers.

The Leap Motion especially allows us to explore means of word separation in the 3rd-Dimension, however, this was discouraged in favor of pinching in Vulture (Markussen, Jakobsen, and Hornbæk 2014). Though discouraged, no empirical evidence was provided, prompting this study to explore the 3rd-Dimension to discover it's full range of problems and to seek possible solutions. The 3rd-Dimension will be utilized by implementing various approaches that also consider the z-direction of hand-movement.

A final alternative, is to use a bimodal approach to mid-air interaction using a secondary input that replaces the pinching gesture. Removing the requirement for producing a gesture or using the 3rd-Dimension, the aim is to vastly reduce the complexity of word separation.

1.4.2 Motor Space vs Display Space

To address the issue of decoupling, the 3rd-Dimension will be utilized by implementing a default approach with a static plane in mid-air which is expected to most show the effects of decoupling. To study the real limitations and issues that this decoupling has, the static interaction plane will also be projected onto a surface with a printed keyboard beneath it. Also in an attempt to minimize decoupling between the motor space and display space while exploring 3D, an implementation will be used that tries to predict when users are gesturing to touch the mid-air interaction plane.

Additionally, as Vulture did with pinching, the bimodal approach aims to minimize the effects of decoupling by reducing the overall complexity of the interaction between the motor space and the displayed screen and removing the need to actively make specific hand-gestures. The bimodal approach aims to increase the simplicity of simulating touching.

1.4.3 Fatigue

This study introduced calibration of a custom interaction space, similar to that used in Personal Space (Hari Haran 2014) to allow users to rest their arms during text-entry and combat the effects of "Gorilla Arm Syndrom" (Guinness 2015, Yoo, Lee, and Ahn 2012). Though this area of research is highly beneficial to reducing fatigue for mid-air gestures, this feature was dropped as a requirement in favor of a study that was more focused on text-entry and word separation and can be further improved by implementing better methods of personally defined interaction spaces for resting arms (Hari Haran 2014, Guinness 2015).

CHAPTER TWO

Literature Review

2.1 Mid-Air Pointing

Finding alternatives to word separation 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 the 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 began 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 has heavily affected these ray-casting techniques is their dependence on the distance from the display screen (Shoemaker, Findlater, Dawson, and Booth 2009). Markussen et al. chose to minimize the distance dependence by projecting movement orthogonally onto the display to maintain a constant control-display ration at the cost of the user's reach (Markussen, Jakobsen, and Hornbæk 2013). A second option, the option used in this study, is to instead project an interaction plane in front of the user within their reach that is independent from the display space (Hari Haran 2014, Guinness 2015, Bachmann, Weichert, and Rinkenauer 2015). The major issue in decoupling the motor space and display space is that participants have to recouple the two spaces mentally, addressing the principle of stimulus-response compatibility (Markussen, Jakobsen, and Hornbæk 2014, Proctor and Vu 2006). Although it is difficult to recouple these spaces, promising results have shown that the effects can be minimized using different techniques to calibrate the motor space and provide for efficient pointing throughput (Hari Haran 2014, Guinness 2015).

2.2 Barehanded Interaction

Unlike Vulture, which utilized a glove with reflective markers, this study decided to make interactions as natural as possible and use a barehanded approach for gestural interactions. Users should not be forced to wear a glove or reflective markers for interactions (Wachs, Kölsch, Stern, and Edan 2011), this is known as the design principle, "Come As You Are" (Triesch and Von Der Malsburg 1998). The barehanded approach for gestural interactions has been showed to be better than traditional input devices (Von Hardenberg and Bérard 2001), and has been used for natural and relaxed gestures for computer interaction (Freeman, Vennelakanti, and Madhvanath 2012). More recently, the Leap Motion Controller 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).

2.3 Mid-Air Text-Entry

Since it's inception, many techniques have been used for mid-air text-entry that all vary in effectiveness, namely error rate and text-entry rate. 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 mainly on selection-based techniques. For selection-based text-entry, efforts were centered around making targeted movements, hovering, or other techniques representing selection in order to indicate that a character was being pressed. The major limitation to these techniques is that they use single-input text-entry and are slow. One of the earliest implementations, even similar to what was constructed in this experiment is Visual Touchpad (Malik and Laszlo 2004), however it was extremely 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 recently performed better for selection-based techniques reaching rates as high as 18.9 WPM (Shoemaker, Findlater, Dawson, and Booth 2009) and 13.2 WPM (Markussen, Jakobsen, and Hornbæk 2013) by utilizing new selection-based methods for text-entry utilizing a QWERTY-based layout. These studies were limited however, the former providing no character production on erroneous input (Shoemaker, Findlater, Dawson, and Booth 2009), and the latter using orthogonal projection of the hand's position onto the display (Markussen, Jakobsen, and Hornbæk 2013).

2.3.2 Gesture-Based

More recent efforts as technology has improved and gestures have become more easily analyzed and detected, have focused on looking at gesture-based techniques for mid-air text-entry. One of the prominent examples is AirStroke (Ni, Bowman, and North 2011) which utilized a Graffiti-based alphabet for text-entry and achieved text-entry rates as high as 11.0 WPM. It was also shown, implemented by using an Xbox Kinect, 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 alternatives have attempted to use hand-writing as a gesture-based technique 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 normal text-entry using hand-writing is known to be slow, limited to around 15 WPM (Devoe 1967).

2.3.3 Word-Gesture Keyboards

At the beginning of this research there was no other work that had been published on word-gesture keyboards in mid-air, however, shortly after research began, Markussen et al. released Vulture, the first mid-air word-gesture keyboard (Markussen, Jakobsen, and Hornbæk 2014). The Vulture keyboard was able to achieve superior text-entry rates for mid-air in two studies, reaching 20.6 WPM and 28.1 WPM using repeated sessions and training on short phrases (Markussen, Jakobsen, and Hornbæk 2014). Vulture used one of the few publicly available, and better-known implementations of work-gesture keyboards that was developed for phones, the SHARK[2] (Kristensson 2007, Zhai and Kristensson 2004), and uses a glove with reflective markers for tracking pinching-gestures on an orthogonally projected display (Markussen, Jakobsen, and Hornbæk 2013).

2.4 Word-Gesture Keyboards

Word-gesture keyboards have seen their emergence along-side the growth of touch-based, and hand-held devices, coming default nowadays on many smart phones. 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). Word-gesture keyboards have garnered text-entry rates as high as 25 WPM after 35 minutes of practice for beginners and up to 46.5 WPM for well-practice phrases (Kristensson 2007).

A traditional part of implementing word-gesture keyboards is to include shape-recognition algorithms in order to best determine what words the user is 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). One of the most thorough, available examples of this is SHARK^2 (Kristensson 2007, Zhai and Kristensson 2004) as was used in Vulture (Markussen, Jakobsen, and Hornbæk 2014).

Word-recognition has been proven to work for word-gesture keyboards, therefore it sits outside the goals of this study. Since it is known in advanced what words and characters that a user is going to be attempting to produce, we believe a word-gesture keyboard can be pseudo-implemented without word-recognition. To ensure that the results are accurate and to analyze the effectiveness of a pseudo-implementation of a word-gesture keyboard, both a pinching-based and touch-based implementation will also be examined.

2.5 Text-Entry Evaluation

2.5.1 Deviating from the Standard Phrases

In the past, to evaluate text-entry, predefined phrases were generated and used (MacKenzie and Soukoreff 2003), however due to the pseudo word-gesture keyboard implementation, and the large number of conditions used, it was chosen to deviate from these predefined sets. The goal was to create new word dictionaries, avoiding phrases to avoid confusing language, using a custom algorithm to measure the similarity of different gesture-shapes. This deviation was motivated by trying to create as similar of an experience as possible for each keyboard to ensure that there were no skewed results because of the limited number of trials in combination with an abundance of different conditions. Section 3.1.5 goes into full details about how this was implemented.

2.5.2 *Text-Entry Rates*

The most important measure for testing the various methods of word separation is by monitoring the text-entry rate. Achieving text-entry rates on par with, or greater than those produced in Vulture, will shed light on better techniques for mid-air interactions for text-entry. Text-entry rate is measured in Words Per Minute (Bi, Chelba, Ouyang, Partridge, and Zhai 2012), however timing for the first character must be included when timing word gestures (MacKenzie and Tanaka-Ishii 2007). More details on this measure are given in Chapter 4.

2.5.3 *Error Rates*

The Vulture keyboard uses Minimum Word Distance to primarily detect error rates in the phrases that are transcribed (Markussen, Jakobsen, and Hornbæk 2014). However, due to the pseudo-implementation of the word-gesture keyboard and the fact that characters and errors can be produced mid-gesture, some character level error rates were also observed. These error rates are the Keystrokes Per Character, Minimum String Distance, and Total Error Rate (Soukoreff and MacKenzie 2003). Of the three, Keystrokes Per Character isn't without it's flaws as noted by Soukoreff and MacKenzie, however, because characters are produced during the gesturing process, this might give some insight on the character production rate of the pseudo word-gesture keyboard implementation. Chapter 4 provides more details on how these were specifically used.

2.5.4 *Correctness*

Word-gesture keyboards heavily rely on shape-recognition algorithms in order to determine which words are recognized and produced, however, many of these studies seem to analyze error rates of the 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, is 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 is between the two paths, the closer the two paths produced are to each other, giving a way to rate the correctness of the generated paths. Section 4.2.6.3 gives the implementation of the algorithm in more detail.

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 outside the scope of this study. Word-recognition had already been proven to work and benefit word-gesture keyboards immensely (Kristensson 2007, Zhai and Kristensson 2012, Zhai, Kristensson, Gong, Greiner, Peng, Liu, and Dunnigan 2009, Zhai and Kristensson 2004, Zhai and Kristensson 2003). In addition, the words that each participant was entering was known in advance, therefore a pseudo word-gesture keyboard implementation was used.

3.1.2 Design

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

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 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, there was no feedback shown to the participant of the change in key size.

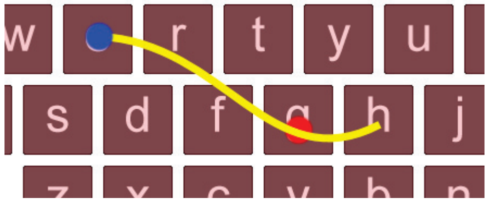
Figure 3.1. Behind the scenes, invisible to the participant, the next key to be pushed is increased in size to make hitting it easier.

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 detected. 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 the how the expected path protects against natural deviations when moving from one key to the next.

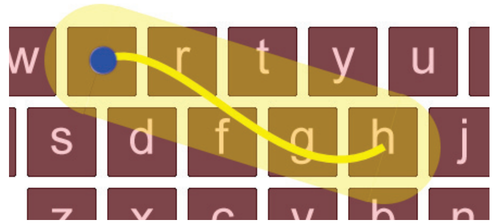
The specific values for detecting key presses were found using trial and error and were used to create an experience as close to a word-gesture keyboard as possible. The main deviation between between this implementation and one with word-recognition is that participants are shown updates in real-time of what the keyboard path is doing. This was determined to be an acceptable limitation.

3.1.3 Display

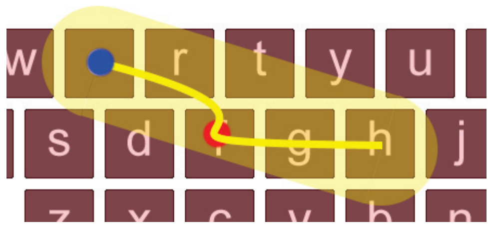
3.1.3.1 Keyboard Layout. The keyboard layout, seen in Figure 3.3, was laid out in a typical QWERTY keyboard fashion with key sizes of 64x64 pixels and gaps of 10 pixels. All special keys and number keys were removed to simplify the keyboard,



(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 stays on the path, errors are significantly reduced. **(c)** shows that errors can still be detected on a protected path if a deviation with an angle less than 90 degrees is detected.

with a backspace key added to the right of the keyboard to allow for deletion of erroneous characters.

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 showed the current word to transcribe, shown in Figure 3.4a, and the bottom area showed the currently transcribed text, shown in Figure 3.4b. The characters of both the displayed word and transcribed text were highlighted in green when the characters matched and were correct, whereas the letters in only the transcribed text would highlight in red if errors were made during transcription. The participant can then use the back space in order to correct words, and they are finally highlighted in green, as seen in Figure 3.4c, when a word was completed and correct.

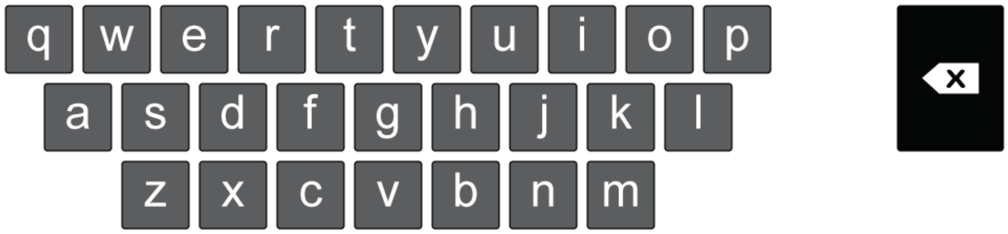
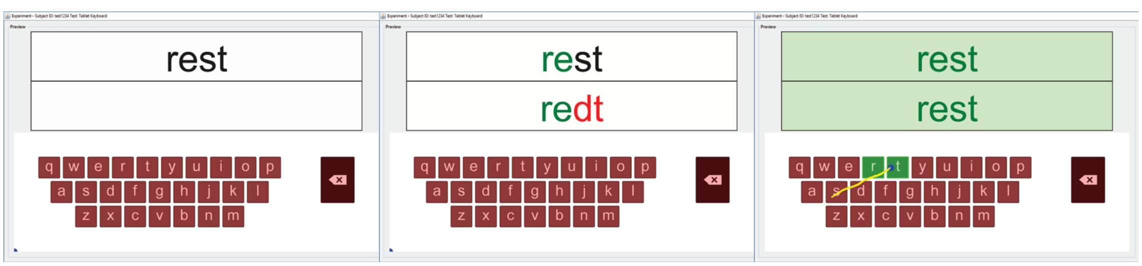


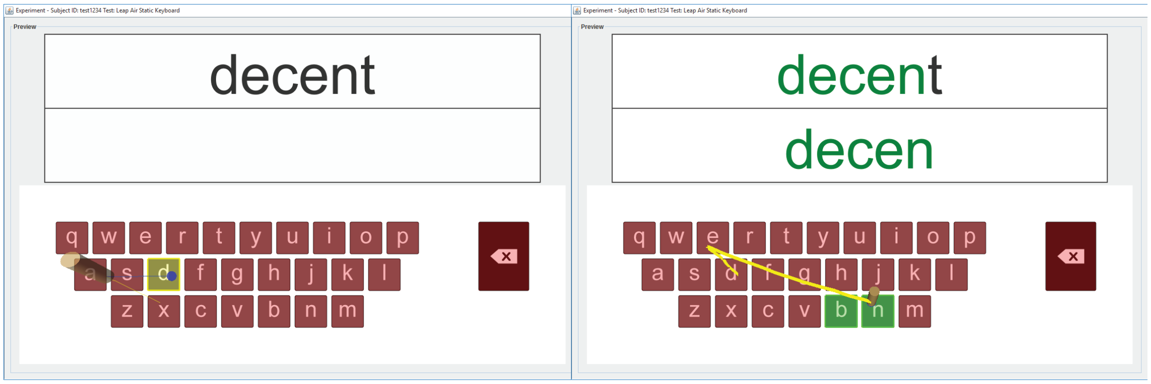
Figure 3.3. The keyboard layout used during the final study.

3.1.3.3 Real-Time Updates. As a participant is drawing the gesture-shape of a word, their progress is tracked in real time, as shown in Figure 3.5. For the keyboards that track a participant's finger or the stylus, Figure 3.5a shows how a cylinder is displayed to indicate the direction that the finger is pointing, as well as which letter is being hovered over, indicated by a blue dot. As in Figure 3.5b, the participant is shown the path that they are traveling as well as the letter that have been hit. The gesture-trail has a maximum length before it starts to decay as to not clutter the display.



(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 transcribe as it first appears. **(b)** shows how correct letters are highlighted green in both text areas and transcription errors are highlighted in red. **(c)** shows a completed word, the background lights up to indicate correctness.



(a) Prior to pressing ’d’

(b) During the word-gesture

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

3.1.4 Calibration

Each mid-air keyboard, as well as the Leap Surface Keyboard, had the ability to be calibrated in a similar manner as Personal Space (Hari Haran 2014), seen in Figure 3.6. Many of the calibration spaces however had to be interacted with directly. Calibration was optional and a default calibration was used that worked okay for most people. Calibration had less of a lasting effect because the Leap Motion Controller was allowed to be repositioned which was sometimes a greater factor than the calibration itself. Because this study is not an accessibility study, participants were allowed to calibrate the keyboard with their arms rested or raised, not addressing the ”Gorilla Arm Syndrome” mentioned in Section **??**.

3.1.4.1 Motor Space vs Display Space. The calibrated interaction plane, or the motor space, was always attempted to be placed in parallel to the screen, however for working with the 3rd-Dimension, a keyboard angled away from the participant was sometimes more effective. Figure 3.7 shows the difference between a straight plane and angled plane.

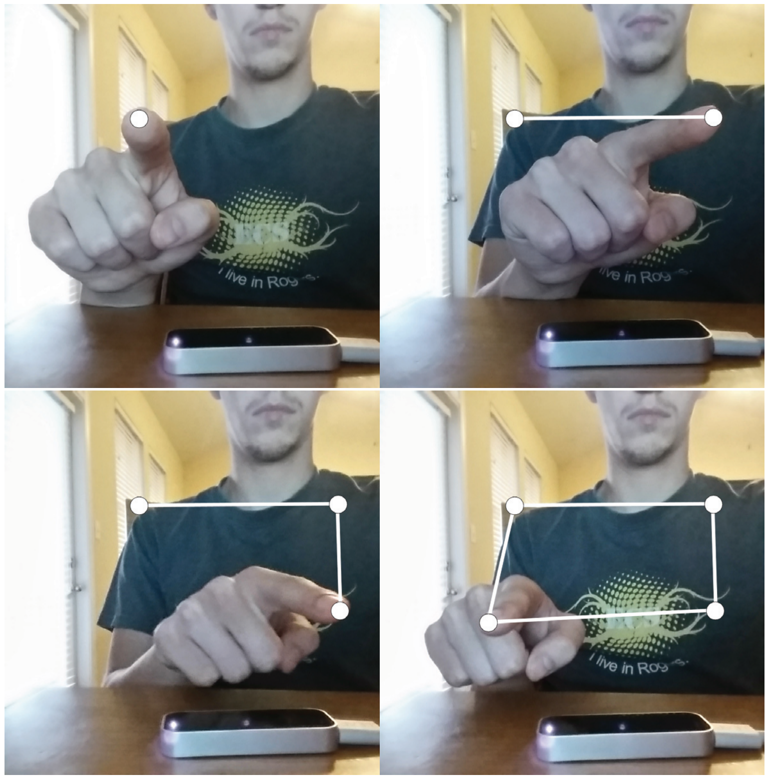
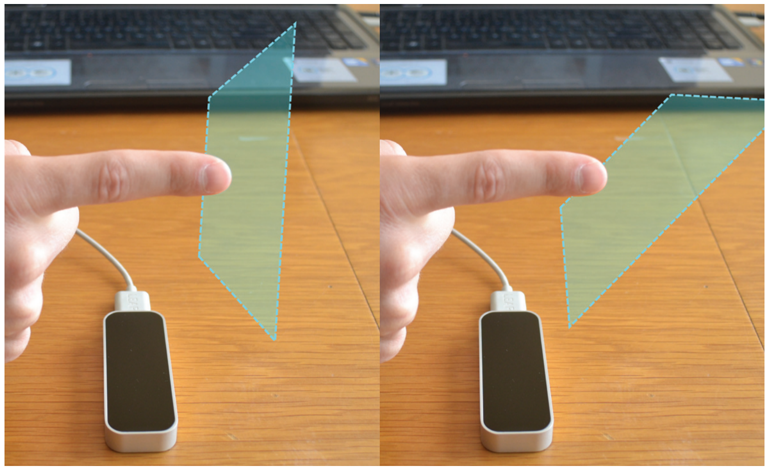


Figure 3.6. Participants were able to follow on-screen instructions to calibrate the space interaction space using their finger.

3.1.4.2 Size of the Motor Space. The size of the motor space was dependent on the device being used or the calibration of the keyboards. For all of the keyboards the motor space was mapped to a display space with the size of 952x212 pixels and keys that were 64x64 pixels with gaps of 10 pixels. The real-world size of the display was dependent on the screen being used. Figure 3.8 shows the average sizes of the motor spaces.

3.1.5 Dictionary Creation

To make each keyboard experience as similar as possible, custom dictionaries were created for each keyboard input device containing words with similar gesture-shapes between keyboards. Dictionaries were created using a custom gesture-shape dissimilarity algorithm. The dictionaries contained words between the character



(a) Straight Plane (b) Angled Plane

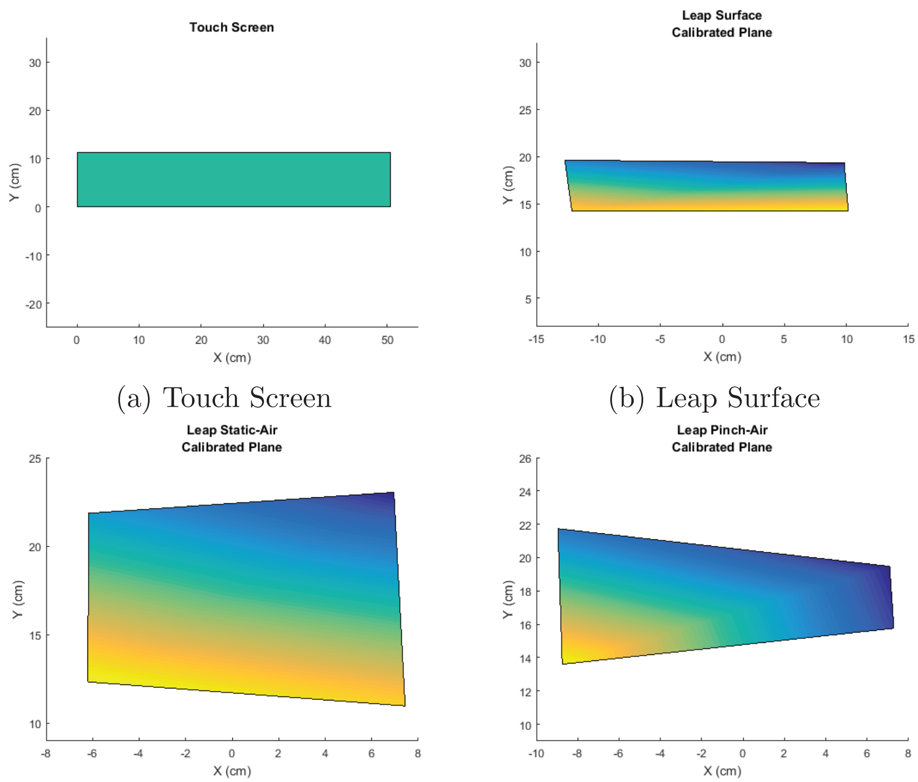
Figure 3.7. Examples of a straight plane versus an angled plane.

lengths of 3 and 6 so that words would be rather simple, and words that most participants had seen before.

3.1.5.1 Using Similar Gesture-Shapes. The reason that custom dictionaries were chosen to be created was because it was thought that dependent measures would be more accurately represented for each keyboard if the experiences between those keyboards was nearly the same, only changing due to implementation. There was no previous research on words with similar gesture-shapes, or if this was truly necessary or beneficial.

3.1.5.2 Gesture-Shape Dissimilarity. Originally the Fréchet Distance was used to try to find the most similar word gesture-shapes using the sets of words with the least distance between each gesture-shape. Fr'echet Distance gave acceptable results, however there were noticeable differences in some of the paths created.

In order to achieve gesture-shapes with even closer similarity, a custom dissimilarity algorithm to rate words' gesture-shapes of the same length was created. The top results for words with the least dissimilarity between each other were then split among the dictionaries. The dissimilarity between two words is defined by the



(a) Touch Screen

(b) Leap Surface

(c) Static/Predictive/Bimodal

(d) Leap Pinch-Air

Figure 3.8. Average motor spaces for each keyboard. **(a)** shows 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. **d** shows the average calibration for the Leap Pinch-Air motor space.

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 length, P_i and Q_i were the vector locations of the characters of the words on the virtual keyboard, *max distance* was the maximum distance between any two letters on the virtual keyboard, $dist(...)$ was the distance between two vector locations, and $angle(...)$ was the angle between two vectors. The dissimilarity formula forced 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 simulation 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 touch-based. The user interacts directly with a touch screen surface.

3.2.1.2 Word Separation. Figure 3.9 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 3.9. A touch is simulated when the tabletop screen is touched.

3.2.1.3 Size of the Motor Space. The motor space for the Touch Screen Keyboard was larger than the other keyboards because it utilized the Ideum Multi-touch Table Platform 46" tabletop, shown in Figure 3.10, which had a maximum resolution of 1920x1080 pixels. 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 Ideum Multi-touch Table Platform was still preferred over using very small touch-keyboards such as those on phones or tablets. Having similar sized motor spaces was to help standardize results between touch and mid-air.

Figure 3.10. The Ideum Multi-touch Table Platform tabletop.

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 into a custom holder, shown in Figure 3.11, 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 except for that it is calibrated to a surface rather than mid-air. A stylus was chosen to be used as an interaction tool for this setup because the positioning of the Leap Motion Controller, to allow for accurate surface emulation, was in a position that made it difficult to successfully track a participants hand or finger. Unfortunately the Leap Controller had to be positioned as it was because the hardware, at the time of implementation, was not configured to recognize a hands from the other direction, or otherwise ”upsidedown”.

Figure 3.11. The custom built holder in order to project an interaction plane onto a surface.

3.2.2.2 Word Separation. Word separation for the Leap Surface Keyboard worked in a similar way to how touch is simulated using a stylus for a phone or tablet. Figure 3.12 shows how touch was simulated by pressing the tip of the stylus against the surface of the printed keyboard. The word-gestures was drawn and then the stylus removed from the surface to complete the action.

Figure 3.12. A touch is simulated when the stylus hits the paper surface.

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 so that it would simulate a touch screen in mid-air by projecting a quadrilateral plane in the air. 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 finger and penetrating the mid-air interaction plane as seen in Figure 3.13. Then while maintaining the intersection, the pointer finger was used to draw the word-gesture, and then finally by pulling the finger away from the mid-air interaction plane, touch was released.

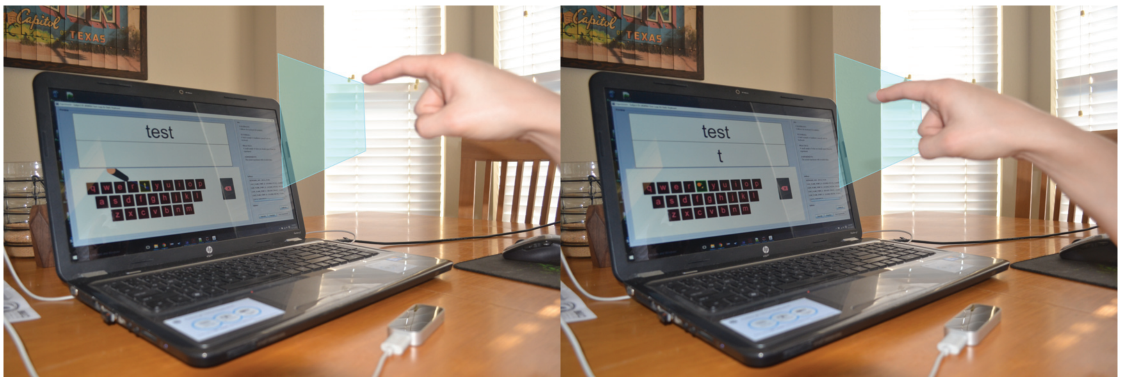


Figure 3.13. A touch is simulated by penetrating the interaction plane.

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 so that it would simulate a touch screen in mid-air by projecting a quadrilateral plane in the air, however instead of having to interact with a static, unchanging plane, the Predictive-Air Keyboard ties the interaction plane with the pointer finger. As the pointer finger moves forward or backward the plane follows. The

Predictive-Air Keyboard then tries to move the interaction plane to the pointer finger by predicting when a touch is being simulated by analyzing forward and backward hand gestures. Slow gestures serve mostly to move the plane, whereas quick gestures generally snap the pointer finger to the plane. The implementation of the forward and backward hand gestures are received from the base Leap Motion API.

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 is in mid-air. The mid-air interaction plane was kept at a static distance away from the detected pointer finger until a forward hand gesture was detected to simulate a touch. Once a touch was simulated, the pointer finger can then be used to draw the word-gesture until it is completed. Finally by making a backward hand gesture away from the interaction plane, the simulated touch is released. The plane interaction is visually similar to Figure 3.13, 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 used the Leap Motion Controller to track the pointer finger of either hand for interaction. It was designed by projecting a quadrilateral plane in the air and snapping the movements of the pointer finger to that plane. A touch was simulated by using a secondary input, in this case a standard keyboard's space bar key.

3.2.5.2 Word Separation. Word separation for the Leap Bimodal-Air Keyboard worked by using a secondary input, the space bar. The interaction plane for simulated touch, as seen in Figure 3.14, was still projected in mid-air. Touch was simulated by using either pointer finger to determine the position over the interaction plane in the x-direction and y-direction and then by pressing and holding the space bar key to simulate touch. Then while holding down the space bar, the pointer finger was used to draw the word-gesture, and then finally releasing space bar to release the touch.

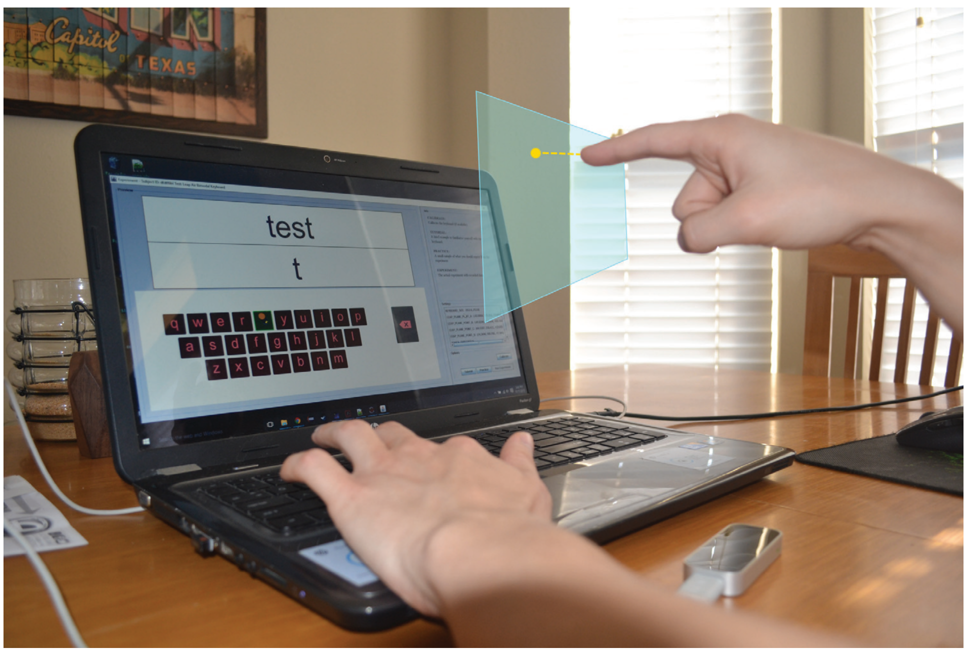


Figure 3.14. A touch is simulated simply by pressing the space bar on the keyboard.

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 Bimodal-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 in mid-air by projecting a quadrilateral plane in the air and snapping the palm position to the plane in the z-direction. The hand then could be used to form a pinch-gesture to simulate touch. The implementation of the pinch-gestures are received from the base Leap Motion API. It is important to note that unlike in Vulture (Markussen, Jakobsen, and Hornbæk 2014), no glove is required, and many different pinch-gestures are recognized.

3.2.6.2 Word Separation. Word separation for the Leap Pinch-Air Keyboard worked by using a pinching-gesture, however the interaction plane was still placed in mid-air. Touch was simulated by using either hand and forming and holding a pinch-gesture as shown in Figure 3.15. While pinching, the word-gesture was drawn and then the pinching-gesture was released to release the touch.

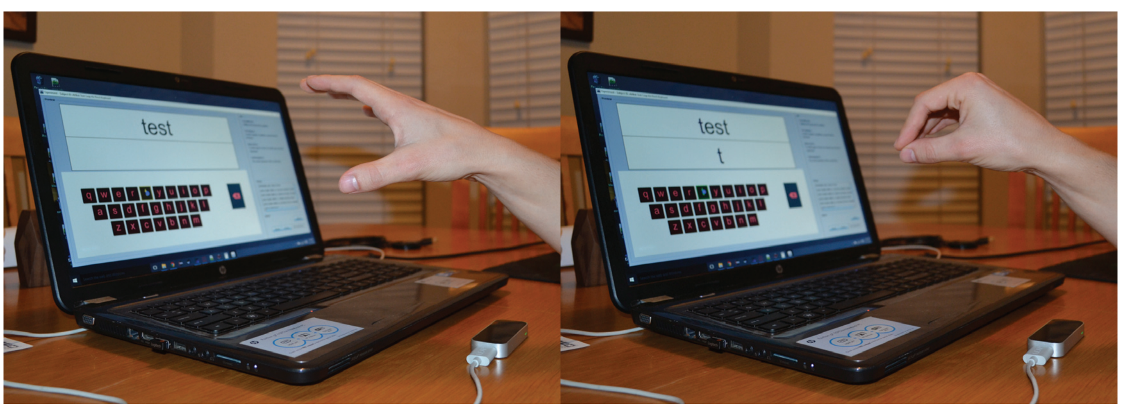


Figure 3.15. A touch is simulated by making a pinching gesture.

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*.

CHAPTER FOUR

Methodology

4.1 Pre-Pilot Study

The pre-pilot study was not a full-blown pilot. This study was used to test the conceptual feasibility of using 3-Dimensions as a means of word separation for simulated touch input and to test the pseudo-implementation for a word-gesture keyboard. See Chapter 3 for implementation details.

4.1.1 Participants

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

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

4.1.2 Input Devices and Interaction

The interaction methods used within the study were dependent on the currently active keyboard input. 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 for a more comfortable experience suited to their arm. Calibration was only possible if applicable to the active keyboard input. Participants were also encouraged to reposition the gesture-controller, if applicable, and were given 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.

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 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.

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. 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 placement was not guaranteed to be identical between uses, in order to accurately simulate an interaction plane projected onto the printed keyboard, the Leap Motion Surface Keyboard required a single calibration after first being inserted into the holder. The participant then used a stylus, as before, that was tracked by the Leap Motion 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.

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 any of the directional sticks or the D-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 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 input devices 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 performed by the participants consisted of 30 trials for each of the 4 keyboard input devices creating a total of 120 trials per participant. For each trial, a word was chosen at random of length 3 through 6 from the Oxford English Dictionary and displayed on the screen. A blank text-area was positioned directly below the displayed word for the transcribed, or user-generated, 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 input device 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 highlighted in green and text entry errors being highlighted in red whereas only correctly matching letters, again highlighted in green, were applied to the displayed word. The participants were required to use the active keyboard input device'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 onto 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. Both participants used every keyboard input device.

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 how many calibrations were performed. 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 input device's tasks.

First, the participants were given a brief explanation and demonstration of the active keyboard input. The explanation dialog contained the name of the active keyboard, how it was interacted with, and whether or not it was a word-gesture keyboard. The researcher then demonstrated how to enter the word "test" using the active keyboard. The participants were then given control and handed the stylus or used their bare hand to interact with the keyboard to get a sense of how it worked.

Participants were then instructed to use the keyboard to perform practice words which were randomly chosen from the Oxford English Dictionary with lengths 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 it was 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 input device. None of the experiment words were duplicated between themselves or the words used during practice and again were pulled at random from the Oxford English Dictionary with lengths 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 input devices, 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 devices as well as rank each device on a numerical scale from best to worst.

Table 4.2. Schedule of Assessments for a single study visit (in minutes).

	Controller	Leap Motion Surface	Leap Motion Static-Air	Leap Motion Pinch-Air	Exit Survey	**total**
explain	.5	.5	.5	.5	0	**2**
calibrate	0	2	2	2	0	**6**
practice	5	5	5	5	0	**20**
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**

4.1.6 Dependent Measures

The pre-pilot study only collected the playback data of participants. The playback data included detected key presses, the calibrated interaction plane, and the tracking location data.

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 more 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, whereas all but one participant had previous experience with gesture-controllers and only 57% having 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

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, interaction with the different keyboard devices were dependent on whatever device was active at the 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 Ideum Multi-touch Table Platform tabletop, 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 which was placed on the desk in front of the participant. The participant then used a stylus which was tracked by the Leap Motion in order 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. 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 which was placed on the desk in front of the participant. The participant then used a stylus which was tracked by the Leap Motion 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 input devices. 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 input devices. 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 device 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 in an attempt to standardize the data collected rather than using random, changing 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 keyboard device for all participants. This was handled by generating static, unchanging dictionaries for each keyboard, guaranteeing a total of 70 unique words rather than a total of 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 of length 3 through 6 characters. This meant that only 10 different gesture-shapes were used by each participant across all input devices, ensuring that all participants' experiences with each keyboards were as similar as possible to each other and other participants. The creation of these dictionaries was 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 keyboard that was previously displayed 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 input device 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 device's backspace key to remove errors, however already correct 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 again used for this study (Kim 2010). The strength of a within-subjects design is that the overall power will increase and there will be a reduction in error variance associated with individual differences. The weakness of using the within-subjects design is that it suffers from carryover effects between each keyboard input device. The participation in one condition may affect performance in other conditions, this is usually referred to as "carryover effects". 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.

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

subject	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

4.2.5 Procedure

Again, 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 input devices' tasks.

First, the participants were given a brief explanation and demonstration of the active keyboard input. The explanation dialog contained the name of the active keyboard, how it was interacted with, and whether or not it was a word-gesture keyboard. The researcher then demonstrated how to enter the word ”test” using the active keyboard. The participants were then given control and handed the stylus or used their bare hand to interact with the keyboard to get a sense of how it worked.

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 type each word with minimal errors. During the practice phase, the opportunity recalibrate the interaction-space any number of times was given if it was applicable to the active keyboard.

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 input device. 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 input devices, 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 a Likert scale section relating to difficulty, discomfort and fatigue experienced when using the devices as well as rank each device on a numerical scale from best to worst.

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

	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	.5	.5	.5	.5	.5	.5	.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**

4.2.6 Dependent Measures

The design choice to not fully realize word-recognition, for the implemented word-gesture keyboard, heavily influenced the design of the task and therefore also 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's possible to analyze these values at the phrase level if every the combined trials are viewed as a single phrase.

4.2.6.1 Text-Entry Rate. Typically, text-entry rates are calculated using the standard Words Per Minute formula

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

where $|T - 1|$ is the length of the transcribed string and S is 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, however, the formula must be modified to:

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

where $|T-1|$ is 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 is 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 was 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 study to use Minimum Word Distance, if the participant made any errors at all regardless of being forced to correct them, then the word was counted as incorrect. This gives the formula

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

where IW are words where the participant made any mistakes at all regardless of corrections and CW are words where the participant got the word correct with one 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, namely backspace, and IF were the incorrect but fixed characters in the input stream, but not editing keys. The Keystrokes Per Character method was less than ideal and has many limitations as an error metric (Soukoreff and MacKenzie 2003), yet it was still beneficial since the task required each word to be typed correctly before moving on to the next and the word-gesture keyboards were designed with a lack of true word-recognition. It's 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 participants were required to correctly transcribe each word.

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, again, 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, namely backspace, and IF were the incorrect but fixed characters in the input stream, but not editing keys. Again, Formula 4.7 can 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, is 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 4.1 shows the recursive implementation of Fréchet Distance used in this study 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 are 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,

```java
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 4.1. A snippet of code showing the recursive implementation of Fréchet distance.

recorded in centimeters, and the average velocity of the participant's hand recorded in centimeters per second.

4.2.6.5 Timing Measures. Timing measures were used to calculate text-entry rates as well as attempt to evaluate the participant's level of focus. The primary time measure taken was the duration required to complete a word's gesture-shape in seconds. The time measure attempting to determine a participant's level of focus was by recording the reaction time to respond to errors. Finally, the duration it took for participants to first simulate a touch and to correctly enter the first letter were recorded.

4.2.6.6 Quantitative Measures. There were two 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.

The number of practice words the participant completed aims to rate the complexity of each keyboard, comparing it to the recorded qualitative measures. Tracking the number of times a touch was simulated should be linked to error rates. This helped to try 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 this 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 that they 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 input device was confusing to understand how to use, and fatigue asked the participant if they had experienced tiredness, soreness, or fatigue from the keyboard they had just used. Lastly, participants ranked the keyboards from 1, most preferred, to 7, the least preferred.

4.3 Final Study

In the final study, the Xbox Controller Keyboard was removed due to it's irrelevance, see Section 6.2.9 for more information on how gaming consoles' virtual keyboards can 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.

4.3.1 Participants

A sample size of 18 was used in the final 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):

$$N = \frac{2(z_{\frac{\alpha}{2}} + z_{1-\beta})^2}{(\frac{\mu_1 - \mu_2}{\sigma_{pooled}})^2} \tag{4.9}$$

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} is 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 4.3.4.

There were 13 male and 5 female participants, 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 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, whereas

83% of participants had previous experience with gesture-controllers and only 56% having previous experience with word-gesture keyboards. No participants had any impairment that affected their ability to enter text with computers. Refer to Table 4.6 for more specific details on each participant.

Table 4.6. 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

4.3.2 Input Devices and Interaction

In order to focus the study on only word-gesture keyboards, the final study saw the removal of the Xbox Controller Keyboard. Again, 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 Ideum Multi-touch Table Platform tabletop, running 64-bit Windows 8. All receivers or controllers were connected through USB 2.0. The participants were again allowed to recalibrate the active keyboard's interaction plane, however a default interaction space was provided that was unlikely to need to be recalibrated. Participants were encouraged to use the default-calibrated interaction space. Again, the participants were allowed to reposition the gesture-controller and were told to use the keyboards in whatever way they felt they could perform best.

A major change to all of the mid-air keyboards is that the stylus was removed. The absence of the stylus was aimed to remove the barrier between the participant and the keyboard input. The goal was to make it feel more natural and more like a touch screen.

4.3.2.1 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.

4.3.2.2 Leap Motion Surface Keyboard. Again, for the Leap Motion Surface Keyboard a Leap Motion Controller was used for tracking. 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. Placement, this time, was insured 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, as before, that was tracked by the Leap Motion 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.

4.3.2.3 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 the pointer finger of their dominant hand, which was tracked by the Leap Motion, 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.

4.3.2.4 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.

4.3.2.5 Leap Motion Bimodal-Air Keyboard. The Leap Motion Bimodal-Air Keyboard used a Leap Motion Controller positioned on the desk in front of the participant as before. The participant then used the pointer finger of their dominant hand, which was tracked by the Leap Motion, 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.

4.3.2.6 Leap Motion Predictive-Air Keyboard. As the others, the Leap Motion Predictive-Air Keyboard saw a Leap Motion Controller placed on the desk in front of the participant. The participant then used the pointer finger of their dominant hand, which was tracked by the Leap Motion, 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.

4.3.3 Task Design

As in the pilot, the conditions of the task were represented by the 6 different keyboard input devices. 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 keyboard input devices. Each task profile consisted of 15 separate trials for a total of 90 trials per participant. The increase in trials from 10 words to 15 words for each device was due to the removal of the Xbox Controller Keyboard. 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 again 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 of length 3 through 6 characters.

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 textN. Beneath both text areas, the keyboard input devices were virtually represented. The participants were then required to use the currently active keyboard input device to enter the displayed word using word-gesturing. During the word-gesturing process, participants were again shown real-time updates. The participants were required to use the active keyboard device's backspace key to remove errors, however already correct transcribed characters were still protected from being deleted. Once a word was correctly entered, the participants were to release the simulated touch to move to the next word.

4.3.4 Experimental Design

As justified in the pilot, a within-subjects design was used for the final study. To minimize carryover effects, the study was supplemented with a Replicated Latin Squares design for counterbalancing (Mansson and Prescott 2001). Table 4.7 shows how the Replicated Latin Squares design was utilized for 6 different keyboard inputs with a sample size of 18.

Table 4.7. 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

4.3.5 Procedure

Each subject participated in a single study visit which took between 30 and 45 minutes to complete. For this study, almost no calibrations were performed. The full set of tasks, and their expected durations, are detailed in Table 4.8, the final study schedule of assessments. The participants followed the same process for each of the 6 keyboard input devices' tasks.

First, the participants were given a brief explanation and demonstration of the active keyboard input. The explanation dialog contained the name of the active keyboard, how it was interacted with. The researcher then demonstrated how to enter the word "test" using the active keyboard. The participants were then given control and interacted with the keyboard to get a sense of how it worked.

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 again 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. The participants were told that the keyboards could be recalibrated if absolutely necessary but were encouraged to learn to use the default calibration. This change was brought about because in the pilot study, participants had a hard time finding calibrations that worked, sometimes taking upwards of 30 minutes with poor results.

Next, the participants performed the task itself. As detailed in Section 4.3.3, the participants were instructed to enter a total of 15 words for the current keyboard input device. The words selected were pulled at random 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 A.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 devices.

Finally, after all tasks were completed for each of the 6 input devices, the participants were asked to fill out an exit survey shown in Figure A.2. The exit survey, as in the pilot, 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 device on a numerical scale from best to worst.

Table 4.8. Schedule of Assessments for a single study visit (in minutes).

	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	.5	.5	.5	.5	.5	.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	.5	.5	.5	.5	.5	.5	3	**6**
total	**5.5**	**5.5**	**6**	**5.5**	**6**	**6**	**3**	**37.5**

4.3.6 Dependent Measures

Again because 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. It's possible to analyze these values at the phrase level if every the combined trials are viewed as a single phrase. In addition, 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 have had modified forms added. The modifications that were used directly effect the data processing for the transcribed word.

The first of the modifications was by using the shortest form of the transcribed word, Table 4.9 shows how this worked. The shortest-transcribed modification helps to account for device detection errors as well as the fact that participants were forced to correct words. The drawback, however, can be seen in Example 4 of Table 4.9, the participant's intention seemed to be the word ”fired” instead of ”fire” but this information was lost with the shortest-transcribed modification.

Table 4.9. Examples of the shortest-transcribed modification.

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

The second modification was by including backspaces as part of the presented word when a participant made mistakes, since participants were required to correct errors. With the modification to the presented text, the transcribed string was now represented by the entire input stream. Table 4.10 shows how this worked. The main motivation behind this modification was to mirror the participants' requirement to use backspaces until a word was correctly entered, especially for the correctness measure, Fréchet Distance.

Table 4.10. Examples of the backspace-transcribed modification.

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

4.3.6.1 Text-Entry Rate. Again, text-entry rates are calculated using modified Words Per Minute, Formula 4.2, where $|T-1|$ is 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 text-entry rate was calculated with and without the shortest-transcribed modification.

4.3.6.2 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, again, uses modified Minimum Word Distance, Formula 4.4, where IW are words where the participant made any mistakes at all regardless of corrections and CW are words where the participant got the word correct with one 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 because of the lack of word-recognition implementation of the word-gesture keyboards, the Minimum String Distance error rate was able to be used (Soukoreff and MacKenzie 2003). Using the simplified keystroke taxonomy that was presented before, Minimum String Distance can be defined as the formula

$$MSD\ error\ rate = \frac{INF}{C + INF} \times 100\% \tag{4.10}$$

wher eC represented the correct characters in the transcribed text and INF were the incorrect, not fixed characters in the transcribed text.

As before, the simplified Keystrokes Per Character formula, Formula 4.6, was used 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, namely backspace, and IF were the incorrect but fixed characters in the input stream, but not editing keys. Additionally, Formula 4.5 was used with the shortest-transcribed modification.

The final error rate, again, was the simplified Total Error Rate from the Formula 4.8 where, again, 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, namely backspace, and IF were the incorrect but fixed characters in the input stream, but not editing keys. With the edition of the shortest-transcribed modification, Formula 4.7 was also utilized.

4.3.6.3 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 4.1 shows the recursive implementation of Fréchet Distance used in this study 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 are the indices that were being examined for that particular recursive phase. The Fréchet Distance was also calculated for both the shortest-transcribed and backspace-transcribed modifications.

4.3.6.4 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.

4.3.6.5 Timing Measures. As before in the pilot, the primary time 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 were recorded.

4.3.6.6 Quantitative Measures. Again, there were two 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.

4.3.6.7 Qualitative Measures. The qualitative measures, in the final 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. Again, see Figure A.2 to see the exit survey.

CHAPTER FIVE

Results and Analysis

The statistical methods used consisted of One-Way ANOVAs for each set of the dependent measures (Wu and Hamada 2009, Neter, Wasserman, and Kutner 1996), followed by using Tukey's Honest Significant Difference (HSD) for multiple-compare for the post-hoc analysis (Hochberg and Tamhane 2009). The ranking system from the exit survey used the Friedman's test for analysis 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 were no significant effects with word-gesture experience as a factor, meaning, participants who had word-gesture experience performed about the same as participants who didn't.

5.1 Text-Entry Rate

Figure 5.1 shows the mean text-entry rates for each keyboard input device. 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. It should be noted that the participant with 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, is consistent with the results from Vulture ($M = 11.8$) for participants' first session (Markussen, Jakobsen, and Hornbæk 2014). Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 55.5017$, p-value $= 1.4206e$-27, ($SD_{pooled} = 2.5$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

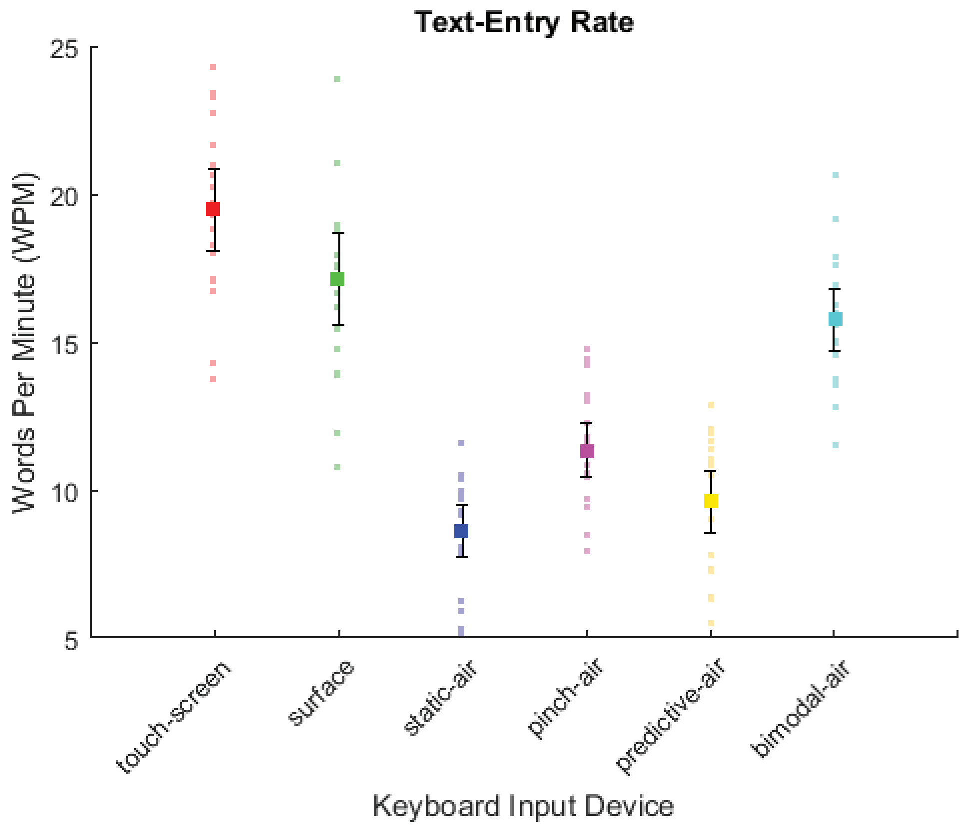


Figure 5.1. Mean Text-Entry Rates for each keyboard with error bars showing 95% confidence intervals.

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

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

The Static-Air keyboard underperformed with the Predictive-Air keyboard achieving marginally better results, however these results were expected. Adding the 3rd-Dimension as a means of interaction further increased keyboard complexity, and increased the mental coupling required to move from the gestures in the motor-space to the feedback on the display, which is difficult (Markussen, Jakobsen, and Hornbæk 2014, Proctor and Vu 2006). It is, however, interesting to see how well the Leap Surface Keyboard performed since it had the same implementation as the Leap Static-Air Keyboard, just instead being projected onto a piece of paper representing the keyboard. This implies that visually representing the Leap Static-Air keyboard in 3D space using augmented reality might have the same results due to the decreased decoupling between the motor space and display space.

5.1.1 Text-Entry Rate Modified-Shortest

Figure 5.2 shows the mean text-entry rates for each keyboard input device using the shortest-transcribed modification. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 54.6430$, p-value $= 2.5223e$-27, $(SD_{pooled} = 2.5)$, prompting the use of Tukey's HSD with multiple-compare for further analysis.

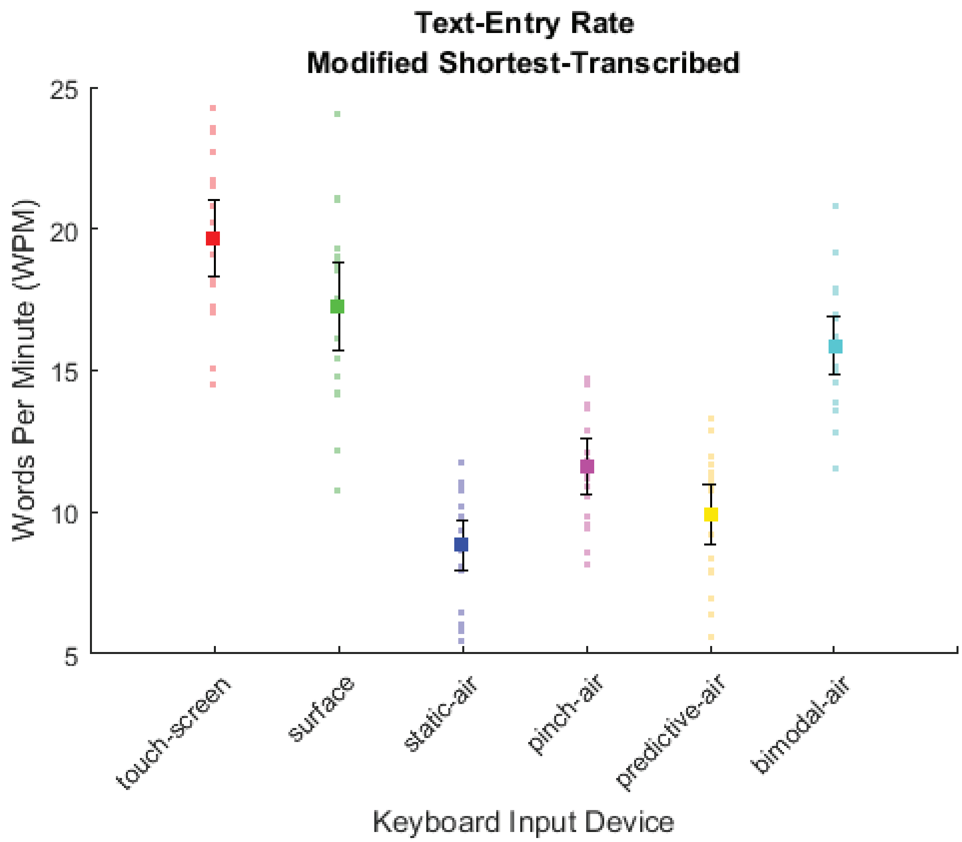


Figure 5.2. Mean Text-Entry Rates using the shortest-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

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

The results from using the shortest-transcribed modification are marginally better than those without, implying that errors were more likely corrected throughout typing the words rather than at the end. The cases where errors were made while still hitting all required letters seem to have very little impact on the duration it takes to complete words.

5.2 Error Rate

5.2.1 Minimum Word Distance

Figure 5.3 shows the mean Minimum Word Distance for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 16.5496$, p-value $=$ 6.1467e-12, ($SD_{pooled} = 18.5$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table B.3, revealed significant differences in Minimum Word Distance between the Touch Screen Keyboard and the Leap Static-Air, Pinch-Air, and Predictive-Air keyboards with p-values < 0.01. There were significant differences between the Leap Surface Keyboard and the Static-Air and Predictive-Air keyboards both with p-values < 0.0001. There were also significant

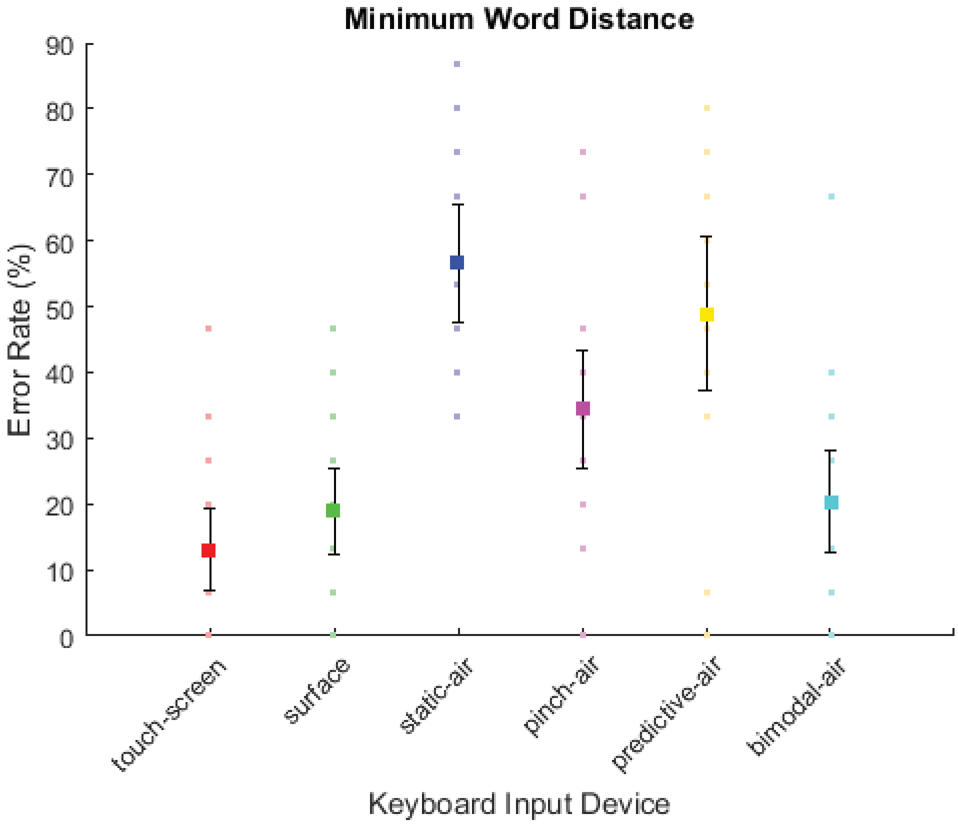


Figure 5.3. Mean Minimum Word Distance for each keyboard with error bars showing 95% confidence intervals.

differences found between the Leap Bimodal-Air and the Static-Air and Predictive-Air keyboards, both with p-values < 0.001. There was finally a significant difference found between the Leap Pinch-Air and Static-Air keyboards with p-value $= 0.0062$.

The error rates here were much higher than those seen in Vulture for Pinch or Touch (Markussen, Jakobsen, and Hornbæk 2014) because this is a modified version of MWD. Since participants were required to fix errors, the MWD from Vulture would have seen error rates of 0% so 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 is counted as erroneous. This MWD is showing us 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 and 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.

5.2.1.1 MWD Modified-Shortest. Figure 5.4 shows the mean Minimum Word Distance with the shortest-transcribed modification for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 8.5172$, p-value $= 9.0987e\text{-}07$, ($SD_{pooled} = 13.5$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table B.4, revealed significant differences 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 with p-values < 0.05. There were significant differences between the Leap Surface Keyboard and the Leap Static-Air, Pinch-Air, and Predictive-Air keyboards all with p-values < 0.05. There were also significant differences found between the Leap Bimodal-Air and the Static-Air and Predictive-Air keyboards, both with p-values $<$ 0.01.

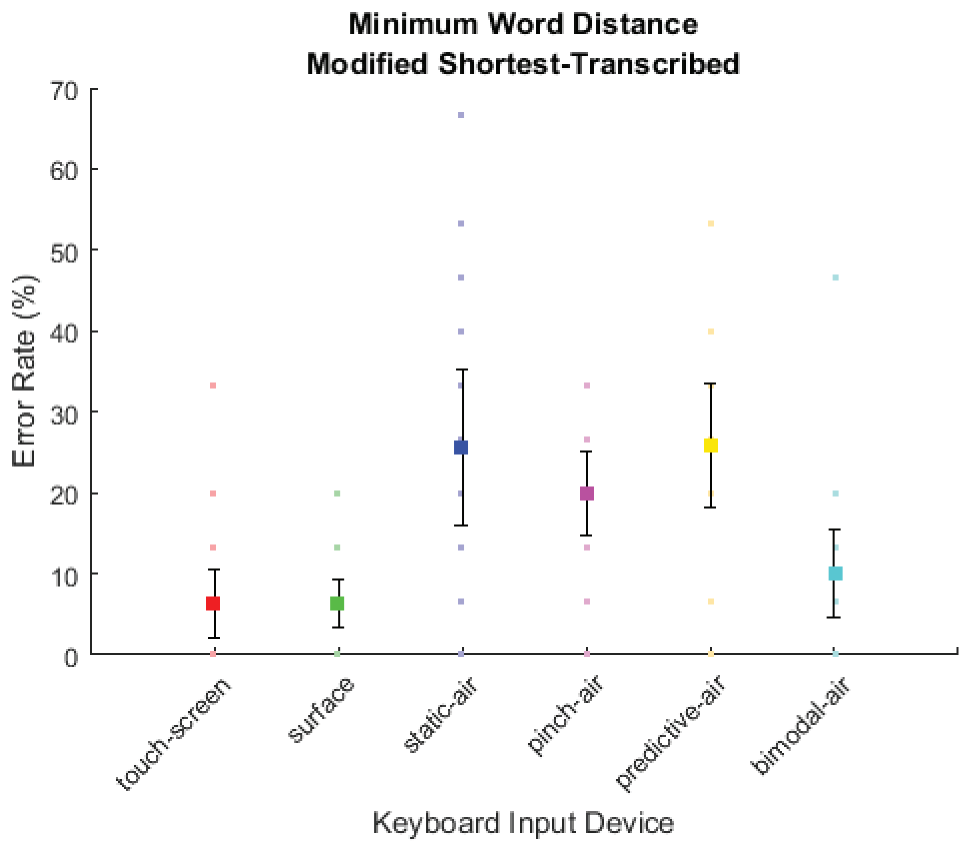
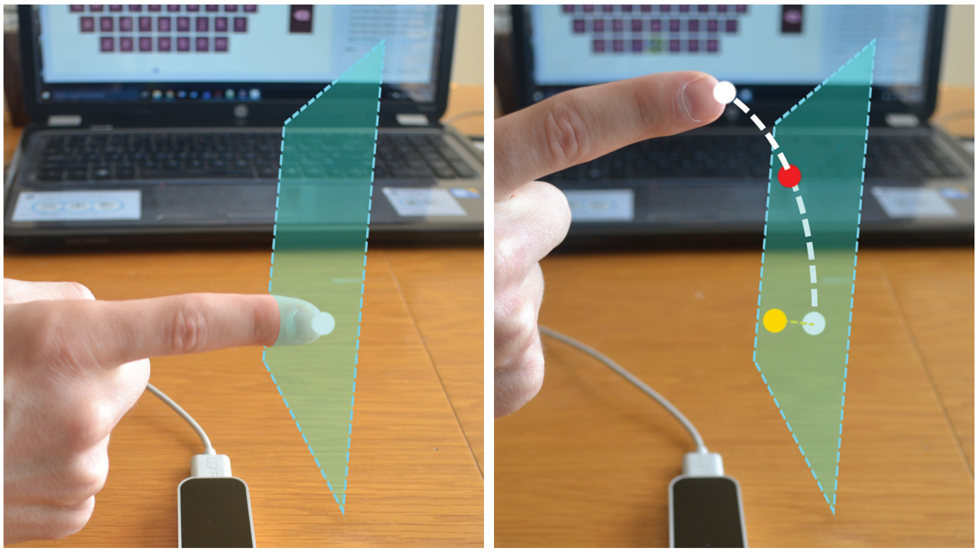


Figure 5.4. Mean Minimum Word Distance using the shortest-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

The huge reductions in MWD error rates, near 50% reduction for all keyboards, when using the shortest-transcribed modification indicates that many participants seem to be making errors after hitting all of the required letters for a word that they are entering. This may imply situations like Example 4 from Table 4.9, where participants are typing the wrong word, such as "fired" compared to "fire". More likely the case, for all of the Leap Motion-based keyboards it implies that many participants made errors on exiting the interaction plane. This kind of error, when exiting the interaction plane, was observed often by the researcher and is thought to be heavily influenced by participants pulling their hands away in an arcing motion, especially for participants who were resting their arm. Figure 5.5 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 best-guess implementation was more sensitive and less robust than word-recognition to deviations in a word's gesture-shape.



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

Figure 5.5. Examples of how the natural arcing motion generates erroneous input for 3-Dimensional interactions. **(a)** shows the user pressing the final key. **(b)** shows the intended release **yellow** and the detected release **red**.

5.2.2 Keystrokes Per Character

Figure 5.6 shows the mean Keystrokes Per Character for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 9.8827$, p-value $= 9.9960e$-08, ($SD_{pooled} = 0.49$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

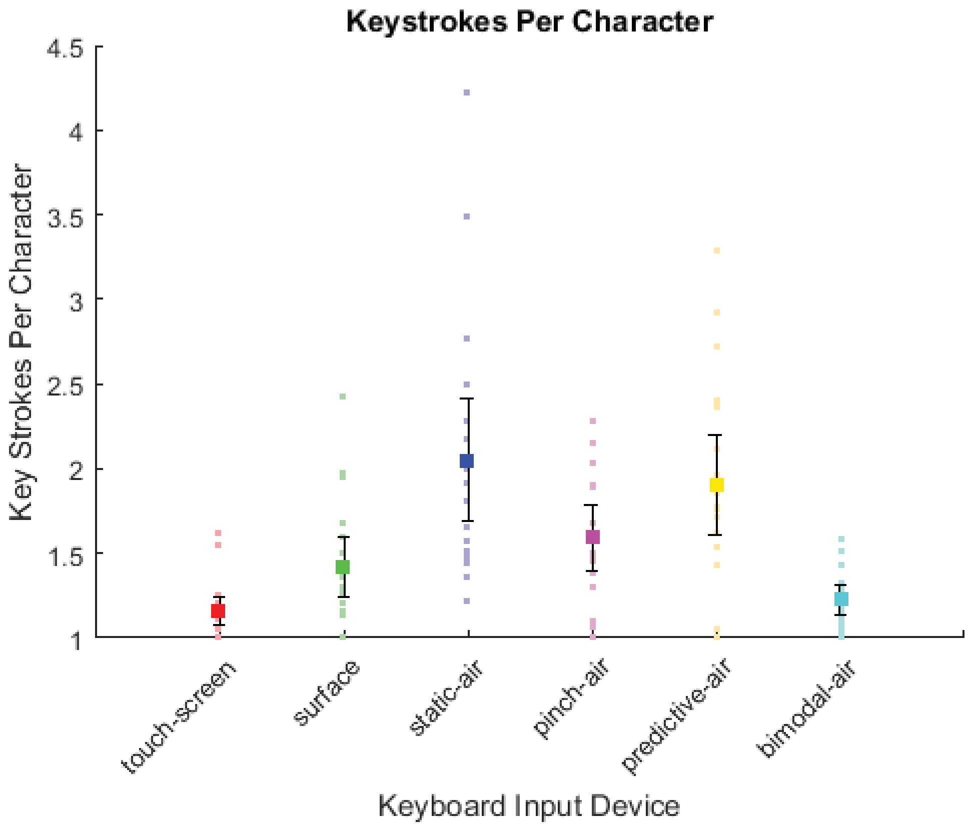


Figure 5.6. Mean Keystrokes Per Character for each keyboard with error bars showing 95% confidence intervals.

The multiple comparisons, seen in Table B.5, revealed significant differences in Keystrokes Per Character between the Touch Screen Keyboard and the Leap Static-Air and Predictive-Air keyboards, both with p-values < 0.001. There were significant differences between the Leap Surface Keyboard and the Static-Air and Predictive-Air keyboards both with p-values < 0.05. There were also significant differences found between the Leap Bimodal-Air and the Static-Air and Predictive-Air keyboards, both with p-values < 0.01.

Keystrokes Per Character was useful because with the best-guess implementation, words are produced at the character level rather than at the whole word level as with word-recognition. With KSPC, if words were input without any detected errors, then we would expect a KSPC of 1.0 keystrokes. 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 type 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 can be seen, a trend is developing in the dependent measures. This trend is expected to hold for all other variables.

5.2.2.1 KSPC Modified-Shortest. Figure 5.7 shows the mean Keystrokes Per Character with the shortest-transcribed modification for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 11.5949$, p-value = 7.0164e-09, ($SD_{pooled} = 0.32$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table B.6, revealed significant differences in Keystrokes Per Character with the shortest-transcribed modification between the Touch Screen Keyboard and the Leap Static-Air and Predictive-Air keyboards with

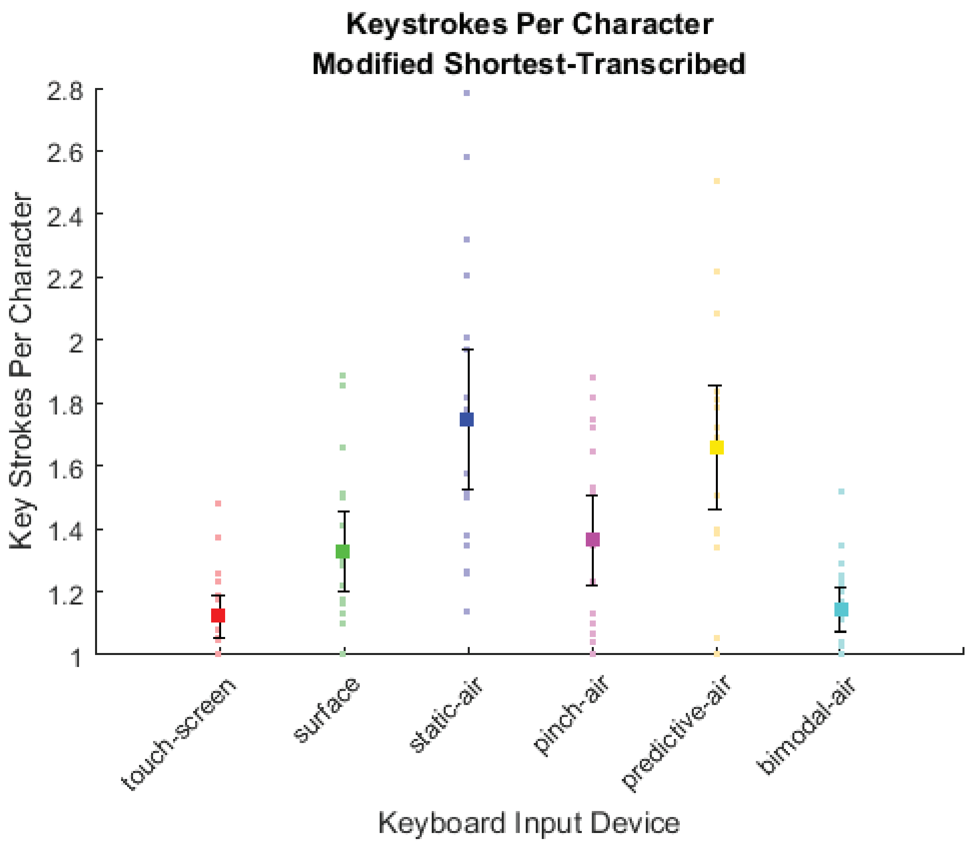


Figure 5.7. Mean Keystrokes Per Character using the shortest-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

p-values < 0.0001. There were significant differences between the Leap Surface Keyboard and the Leap Static-Air and Predictive-Air keyboards both with p-values $<$ 0.05. There were also significant differences found between the Leap Bimodal-Air and the Static-Air and Predictive-Air keyboards, both with p-values < 0.001. Finally there was a significant difference between the Pinch-Air and Static-Air keyboards with a p-value $= 0.0072$.

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

5.2.3 Minimum String Distance

5.2.3.1 MSD Modified-Shortest. Figure 5.8 shows the mean Minimum String Distance with the shortest-transcribed modification for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 6.8688$, p-value = 1.4591e-05, ($SD_{pooled} = 3.3$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table B.7, revealed significant differences 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 with p-values < 0.05. There were significant differences between the Leap Surface Keyboard and the Leap Static-Air and Predictive-Air keyboards both with p-values < 0.05. There were also significant differences found between the Leap Bimodal-Air and the Static-Air keyboards with a p-value $= 0.0067$.

The error rates seen for the Pinch-Air and Touch Screen keyboards for MSD with the shortest-transcribed modification are 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 of 3-Dimensions versus a physical 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

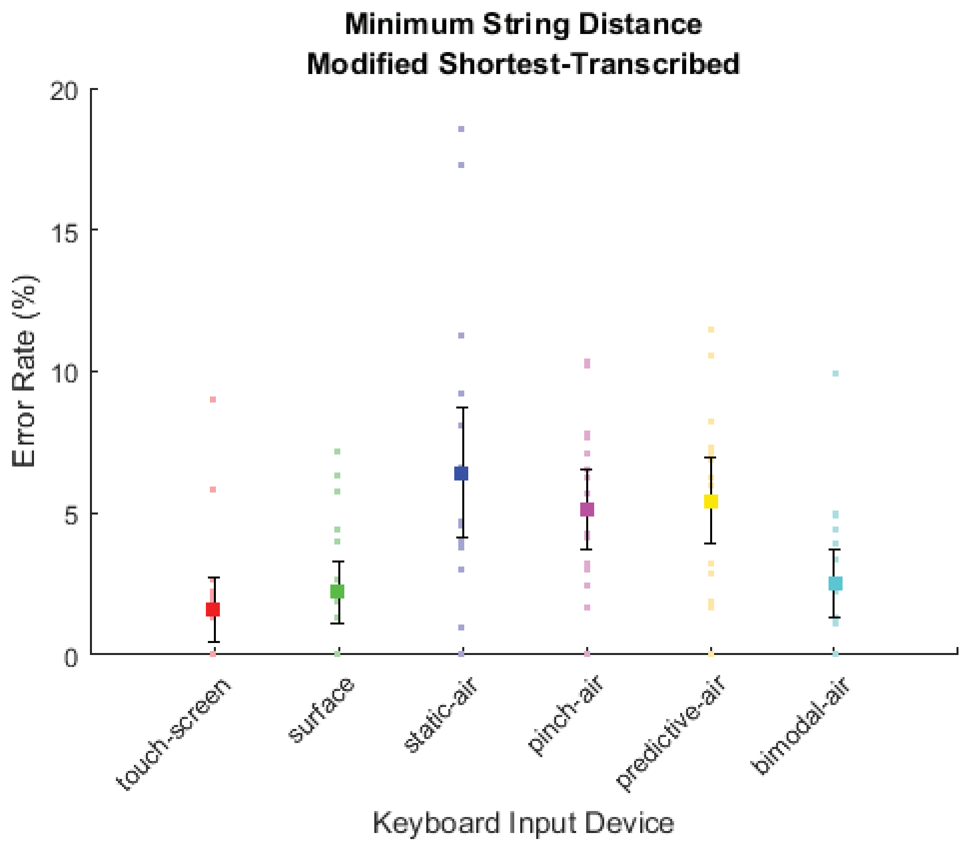


Figure 5.8. Mean Minimum String Distance using the shortest-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

by complexity, so far having never been significantly different than the Touch Screen or the Leap Surface keyboards.

5.2.4 Total Error Rate

Figure 5.9 shows the mean Total Error Rate for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 12.9381$,

p-value $= 9.4983e\text{-}10$, ($SD_{pooled} = 8.8$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

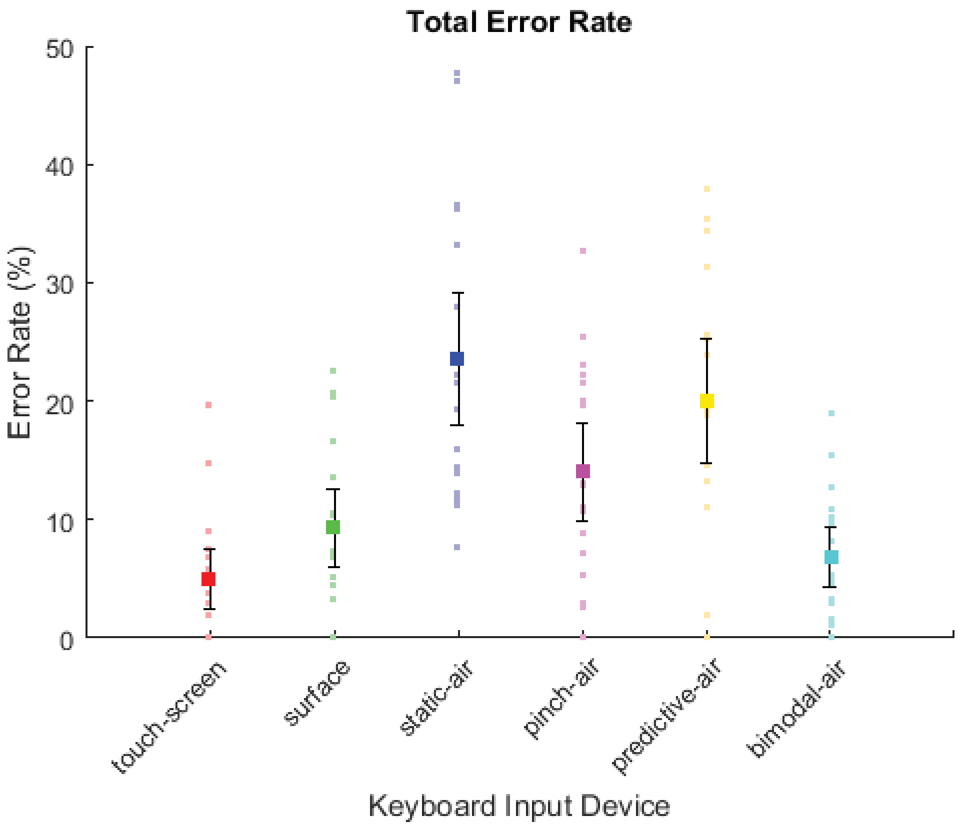


Figure 5.9. Mean Total Error Rates for each keyboard with error bars showing 95% confidence intervals.

The multiple comparisons, seen in Table B.8, revealed significant differences in Total Error Rate between the Touch Screen Keyboard and the Leap Static-Air and Predictive-Air keyboards, both with p-values < 0.0001. There were significant differences between the Leap Surface Keyboard and the Static-Air and Predictive-Air keyboards both with p-values < 0.01. There were also significant differences found between the Leap Bimodal-Air and the Static-Air and Predictive-Air keyboards, both with p-values < 0.01. Finally, there was a significant difference between the Pinch-Air and the Static-Air keyboards with a p-value $= 0.0191$.

Again, we see the keyboards following the same pattern. The Bimodal-Air and Leap Surface keyboards perform about 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, not significantly different from any of them.

5.2.4.1 Total Error Rate Modified-Shortest. Figure 5.10 shows the mean Total Error Rate with the shortest-transcribed modification for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 13.0717$, p-value = 7.8147e-10, ($SD_{pooled} = 8.6$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table B.9, revealed significant differences 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 with p-values < 0.05. There were significant differences between the Leap Surface Keyboard and the Leap Static-Air and Predictive-Air keyboards both with p-values < 0.01. 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 is a significant difference between the Pinch-Air and Static-Air keyboards with a p-value $= 0.0101$.

Surprisingly, unlike KSPC and MWD, the Total Error Rate didn't follow suit and show large decreases in error rate with the shortest-transcribed modification.

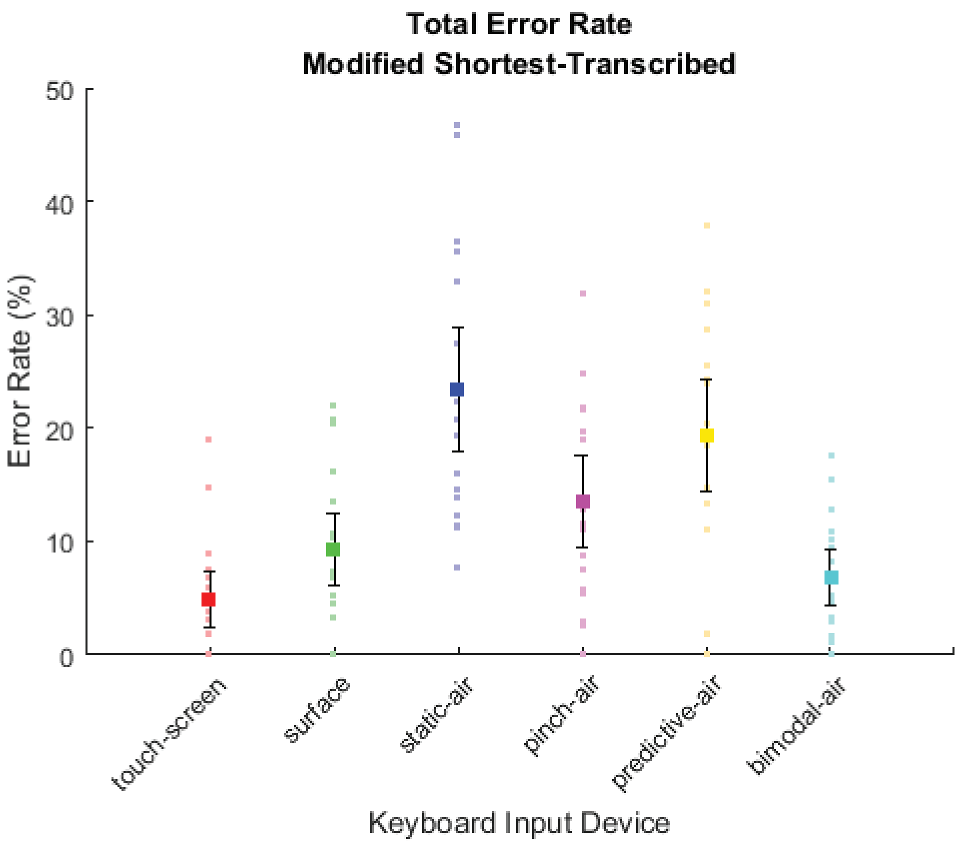


Figure 5.10. Mean Total Error Rate using the shortest-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

These observations do still however support what was seen text-entry rates also being relatively unaffected by the shortest-transcribed modification, implying that errors made by participants were more likely corrected throughout typing the words rather than at the end. This is a major downside of lacking word-recognition because gestures are interrupted and corrected rather than being followed through completely as is natural with word-gesture keyboards.

5.3 Correctness

5.3.1 Fréchet Distance

Figure 5.11 shows the mean Fréchet Distance for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 37.9416$, p-value $= 8.0562e$-22, ($SD_{pooled} = 69.4$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

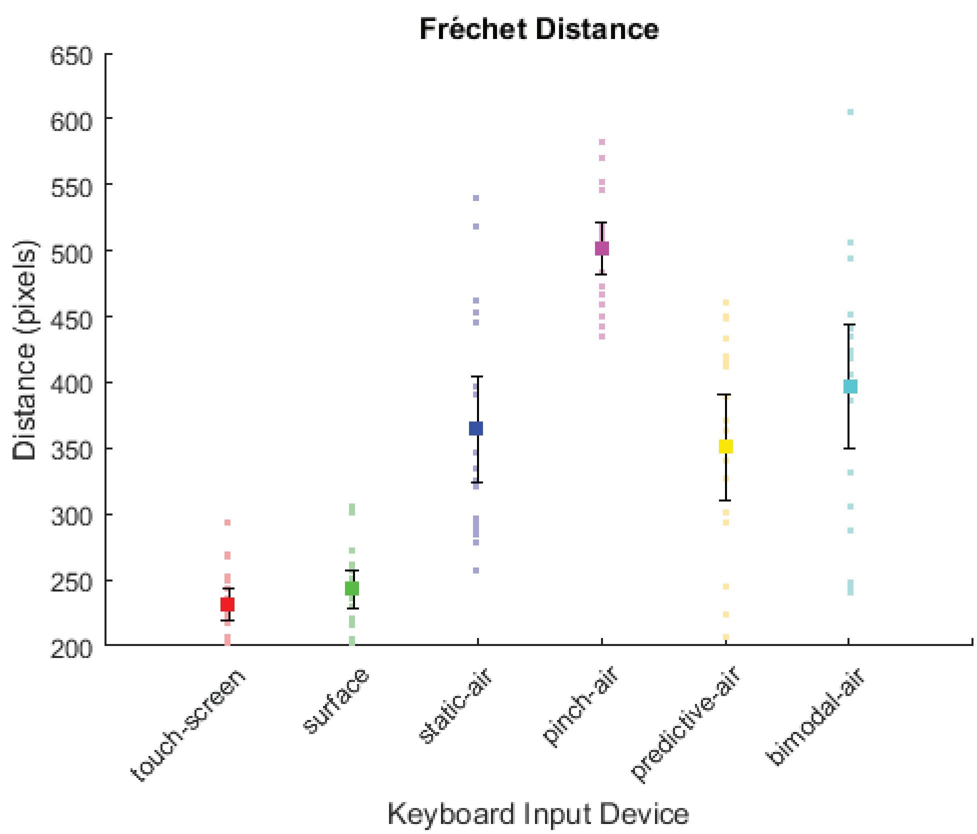


Figure 5.11. Mean Fréchet Distance for each keyboard with error bars showing 95% confidence intervals.

The multiple comparisons, seen in Table B.10, revealed significant differences in Fréchet Distance between the Touch Screen Keyboard and all of the mid-air keyboards with p-values < 0.0001. There were significant differences between the Leap

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

The Fréchet Distance is the only place where the trends are broken and the Bimodal-Air keyboard performs significantly worse than the Touch Screen and Leap Surface keyboards. The Pinch-Air Keyboard also performed significantly worse than all other keyboards. These results are expected to be caused by a combination of the word-gesture velocity seen in Section 5.4.2 and the affects of increase decoupling or complexity from working with the 3rd-Dimension.

The Leap Bimodal-Air seems 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. The Leap Surface and Touch Screen keyboards are relatively unaffected in correctness by their high word-gesturing velocities due to their decreased decoupling from the word-gesture motor space to the displayed screen.

The Pinch-Air Keyboard suffers from the combination of high word-gesture velocities and effects of increased decoupling, however it also seems to be affected heavily by it's deviation in its tracking method increasing it's overall distance traveled for each word-gesture, see Section 5.4.1. All other mid-air keyboards track the participant's pointer finger whereas the Pinch-Air keyboard tracks the participant's palm since their fingers move so sharply when creating a pinching gesture. This difference in tracking combined with high word-gesture velocities are the expected factors for this decrease in performance.

5.3.1.1 Fréchet Distance Modified-Shortest. Figure 5.12 shows the mean Fréchet Distance with the shortest-transcribed modification for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 65.7440$, p-value $= 2.3767e$-30, ($SD_{pooled} = 47.2$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

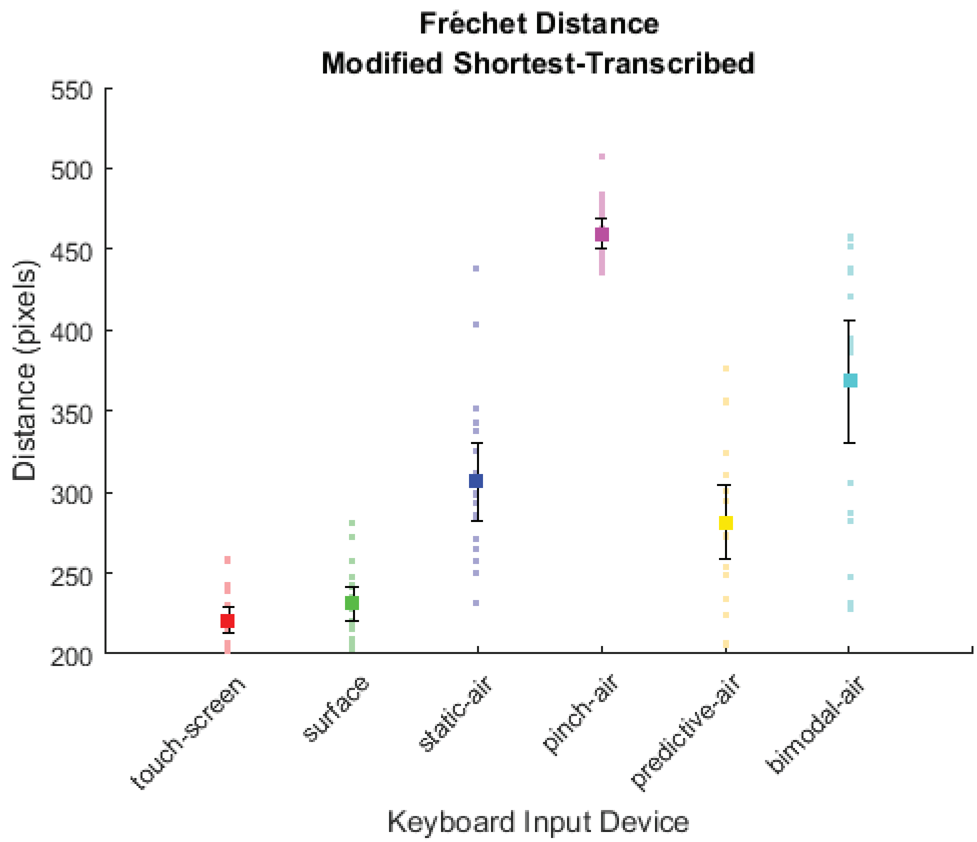


Figure 5.12. Mean Fréchet Distance using the shortest-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

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

Again, the increase in correctness for all keyboards, namely the mid-air keyboards, when using the shortest-transcribed modification implies that many errors are being produced when exiting the interaction plane. This is especially evident for the Static-Air, Pinch-Air, and Predictive-Air keyboards all seeing the largest improvements. For the Static-Air and Predictive-Air keyboards, these errors are caused by the addition of the 3rd-Dimension and participants having trouble entering and exiting the interaction plane accurately. The Pinch-Air Keyboard, again, sees a combination of high word-gesture velocity and a difference in tracking that is expected to produce these kinds of errors.

5.3.1.2 Fréchet Distance Modified-Backspace. Figure 5.13 shows the mean Fréchet Distance with the backspace-transcribed modification for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 97.1196$, p-value $= 3.5271e$-37, ($SD_{pooled} = 39.7$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table B.12, revealed significant differences in Fréchet Distance with the backspace-transcribed modification between the Touch Screen Keyboard and the Leap Static-Air, Pinch-Air, and Bimodal-Air keyboards with p-values < 0.01. There were significant differences between the Leap Surface

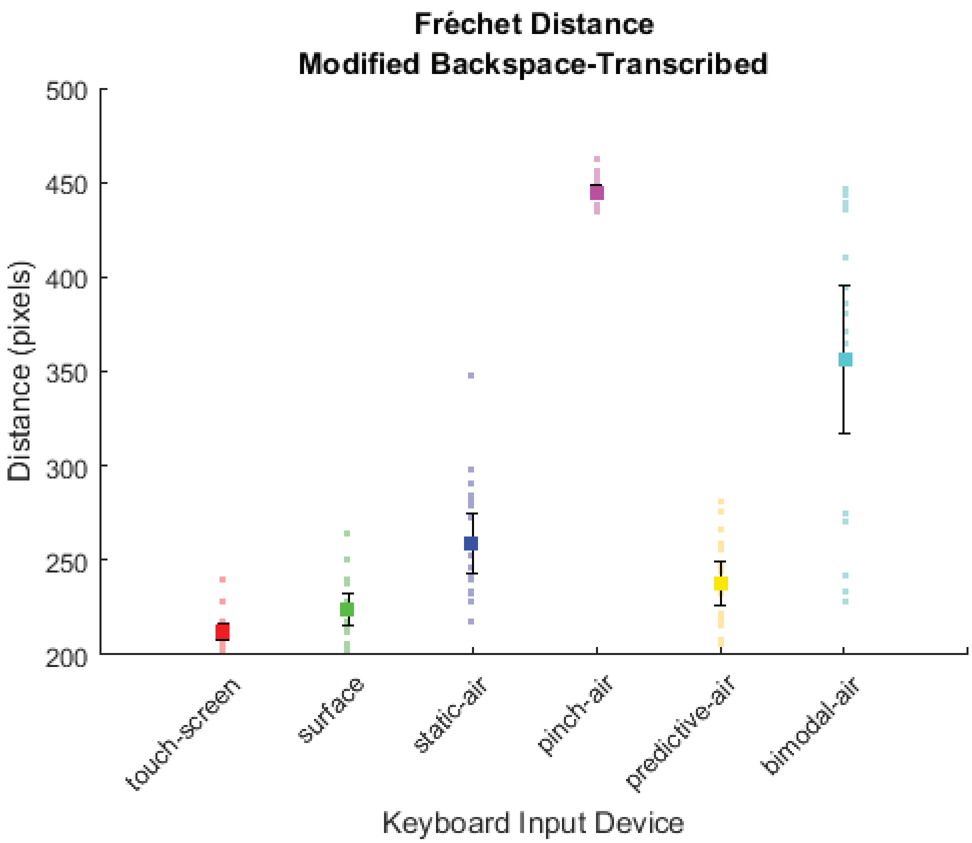


Figure 5.13. Mean Fréchet Distance using the backspace-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

Keyboard and the Leap Pinch-Air and Bimodal-Air keyboards both with p-values $<$ 0.0001. There were also significant differences found between the Leap Bimodal-Air and all other mid-air keyboards with p-values < 0.0001. Finally, there were significant differences between the Leap Pinch-Air Keyboard and the Static-Air and Predictive-Air keyboards with p-values < 0.0001.

If the backspace key is included in the path considered to be correct for Fréchet Distance, since participants were required to correct words, a boost in correctness is observed. The Bimodal-Air and Pinch-Air keyboards still suffer from the previously mentioned problems in Section 5.3.1. The Static-Air and Predictive-Air, however, see huge improvements in correctness because of the low word-gesture velocities and high error rates. The Predictive-Air Keyboard, in this case, is not significantly different than the Touch Screen and Leap Surface keyboards, implying that participants were relatively attentive to correcting errors for it.

5.4 Distance Measures

5.4.1 Word-Gesture Distance

Figure 5.14 shows the mean Word-Gesture Distance for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 13.9225$, p-value = 2.2937e-10, (SD_{pooled} = 398.6), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table B.13, revealed significant differences in Word-Gesture Distance between the Touch Screen Keyboard and all of the mid-air keyboards with p-values < 0.01. There was a significant difference between the Leap Pinch-Air and all other Leap Motion keyboards with p-values < 0.01.

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 is surprising, however, that the Leap Surface Keyboard seems to have 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 hand.

The Pinch-Air keyboard required a significantly longer word-gesture distance than all other keyboards. This is expected to be caused by the difference in tracking

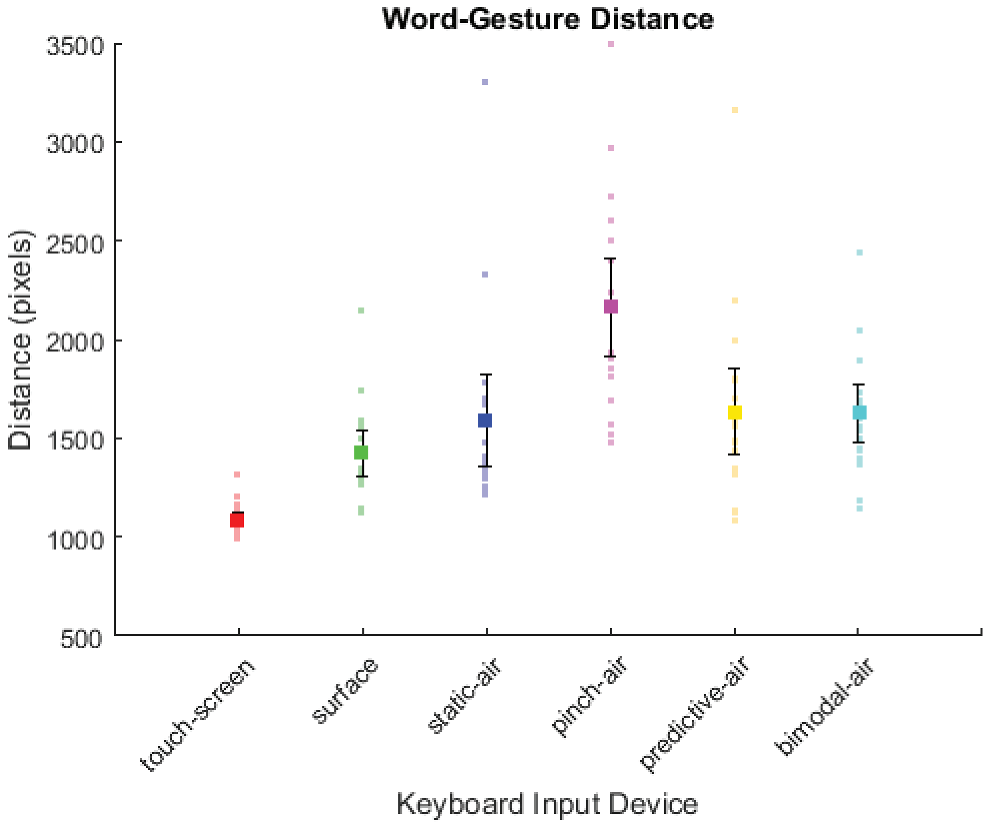


Figure 5.14. Mean Word-Gesture Distance for each keyboard with error bars showing 95% confidence intervals.

method as previously mentioned. Tracking the palm requires larger, less controlled movements than with tracking fingers or a stylus.

5.4.1.1 Word-Gesture Distance Modified-Shortest. Figure 5.15 shows the mean Word-Gesture Distance with the shortest-transcribed modification for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 14.403$, p-value $= 1.1613e$-10, ($SD_{pooled} = 325.1$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

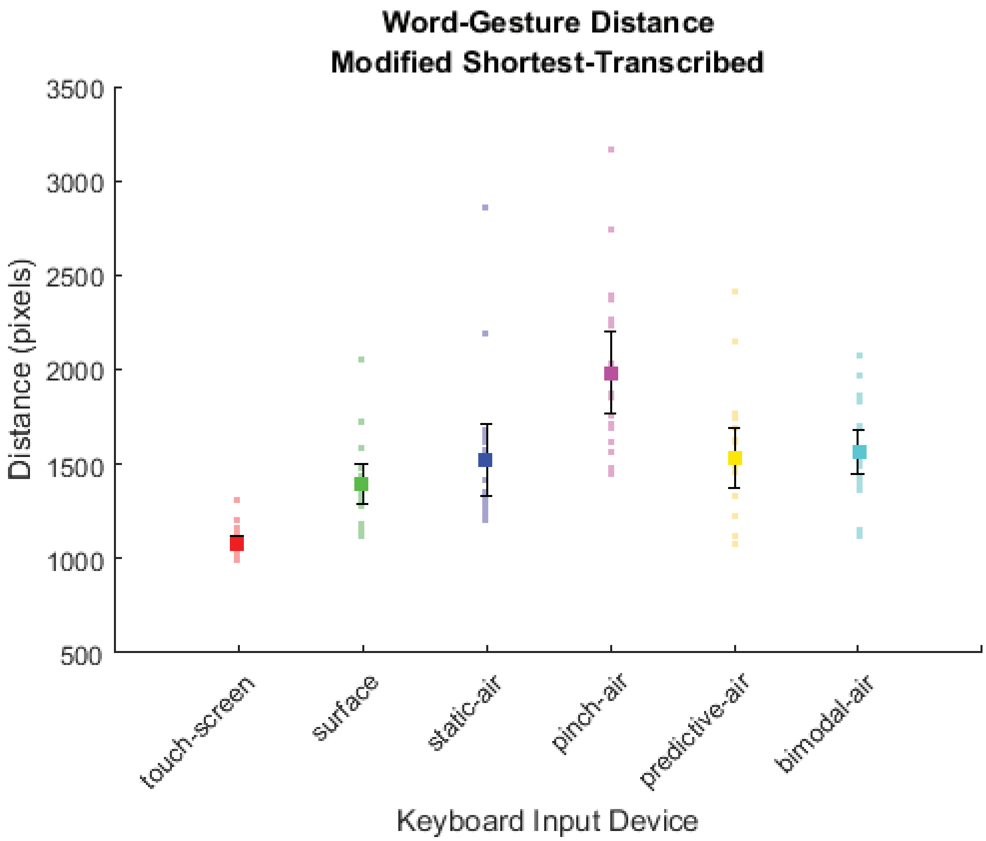


Figure 5.15. Mean Word-Gesture Distance using the shortest-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

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

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

5.4.2 Word-Gesture Velocity

Figure 5.16 shows the mean Word-Gesture Velocity for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 17.233$, p-value $= 2.4922e\text{-}12$, ($SD_{pooled} = 161.8$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

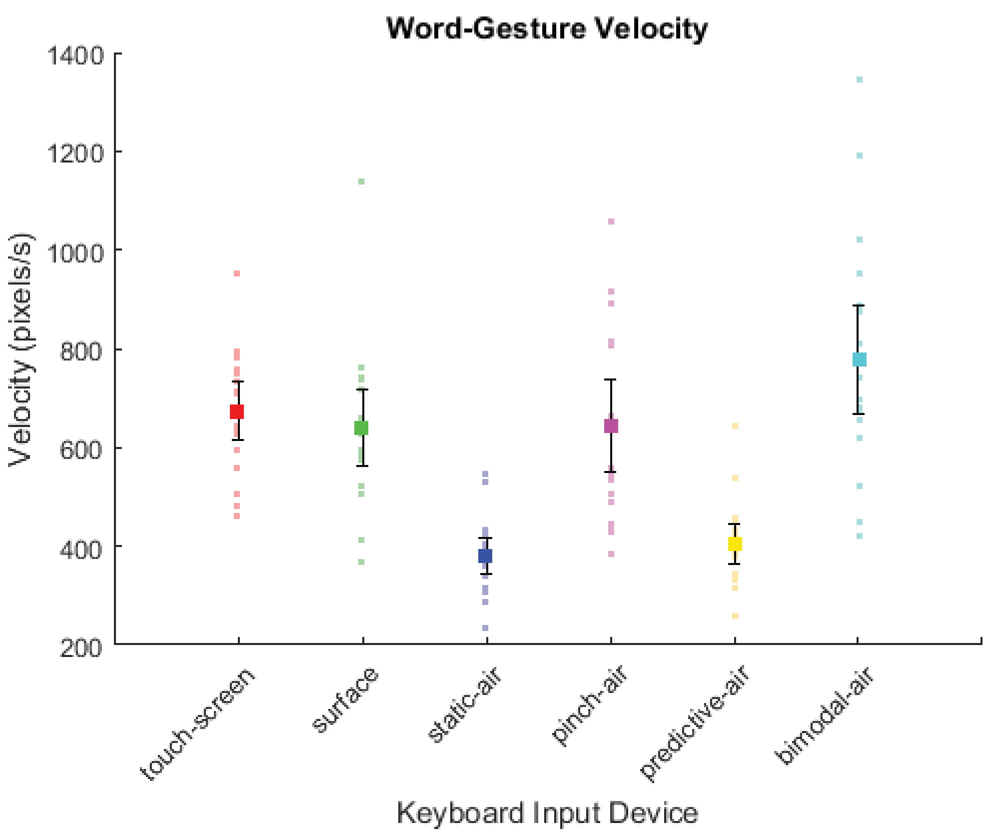


Figure 5.16. Mean Word-Gesture Velocity for each keyboard with error bars showing 95% confidence intervals.

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

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 to working with a 3rd-Dimension in mid-air. It is interesting, though, that these two keyboards also had the highest error rates across all error measures, implying a very high degree of difficulty, even with slower movements.

5.4.3 Hand Velocity

Figure 5.17 shows the mean Hand Velocity for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 121.9826$, p-value $= 2.0877e$-41, ($SD_{pooled} = 4.1$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table B.16, revealed significant differences in Hand Velocity between the Touch Screen Keyboard and all other keyboards with p-values < 0.0001. There were significant differences between the Leap Surface Keyboard and the Static-Air, Pinch-Air, and Predictive-Air keyboards with p-values $<$

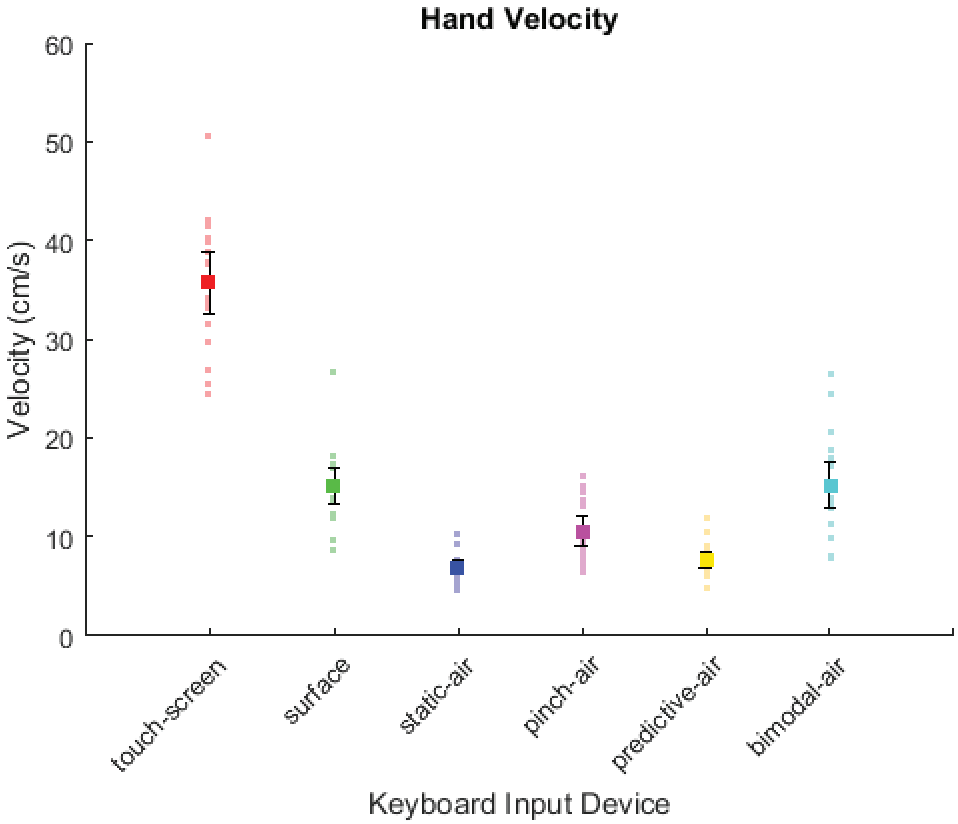


Figure 5.17. Mean Hand Velocity for each keyboard with error bars showing 95% confidence intervals.

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

The hand velocities are all similar to the word-gesture velocities in Section 5.4.2, except for the Touch Screen Keyboard. This result is simply because of the difference in size between the interaction planes that participants used and the physical size of the Touch Screen used, as mentioned in Chapter 3.

5.5 Timing Measures

5.5.1 Word-Gesture Duration

Figure 5.18 shows the mean Word-Gesture Duration for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 28.1205$, p-value $= 8.2440e$-18, ($SD_{pooled} = 0.81$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

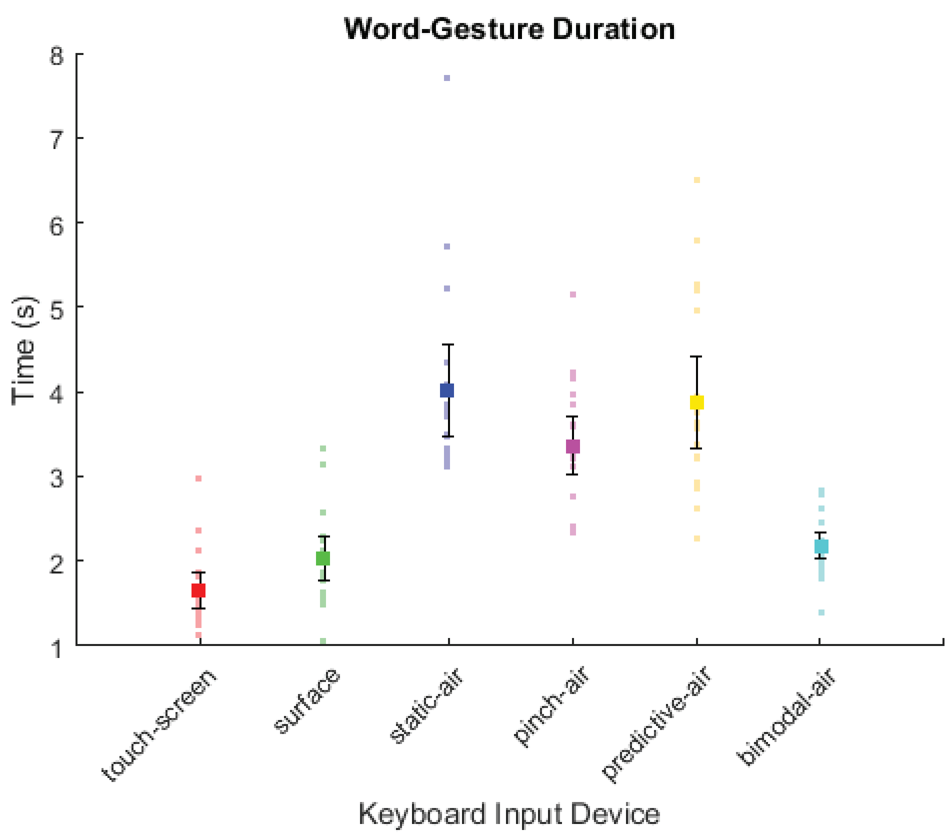


Figure 5.18. Mean Word-Gesture Duration for each keyboard with error bars showing 95% confidence intervals.

The multiple comparisons, seen in Table B.17, revealed significant differences in Word-Gesture Duration between the Touch Screen Keyboard and the Leap Static-Air, Pinch-Air, and Predictive-Air keyboards, all with p-values < 0.0001. There were significant differences between the Leap Surface Keyboard and the Static-Air, Pinch-Air, and Predictive-Air keyboards, all with p-values < 0.001. There were also significant differences found between the Leap Bimodal-Air and all of the other mid-air keyboards with p-values < 0.001.

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 mean that word-gestures take longer to perform, whereas for the Pinch-Air Keyboard, the longer distances traveled were the influence for the longer durations. The Bimodal-Air keyboard performed as well as the Touch Screen and Leap Surface keyboards as expected from it's prior trends.

5.5.1.1 Word-Gesture Duration Modified-Shortest. Figure 5.19 shows the mean Word-Gesture Duration with the shortest-transcribed modification for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 32.0355$, p-value = 1.7009e-19, ($SD_{pooled} = 0.67$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table B.18, revealed significant differences 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 with p-values < 0.0001. There were significant differences between the Leap Surface Keyboard and the Leap Static-Air, Pinch-Air, and Predictive-Air keyboards

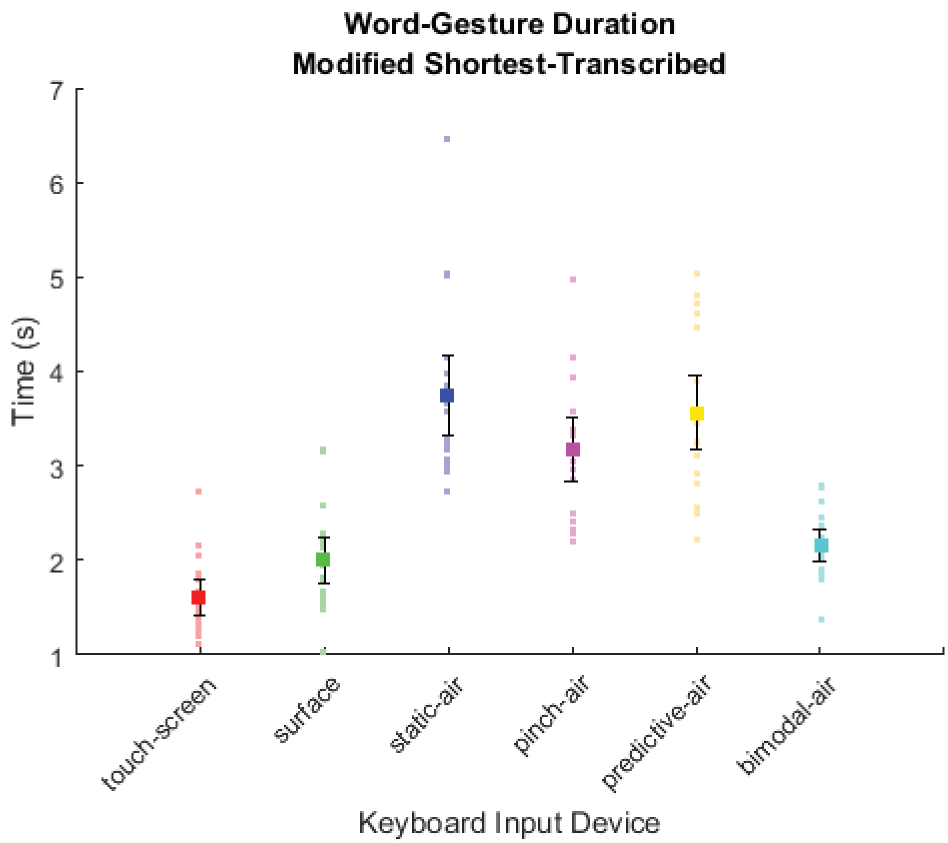


Figure 5.19. Mean Word-Gesture Dur using the shortest-transcribed modification for each keyboard with error bars showing 95% confidence intervals.

all with p-values < 0.0001. There were also significant differences found between the Leap Bimodal-Air and all of the other mid-air keyboards with p-values < 0.001.

As seen before, the shortest-transcribed modification for word-gesture duration had little effect on the total length of word-gesture durations. This again supports the idea that participants mostly corrected errors as they made them while transcribing a word.

5.5.2 Reaction Time to Errors

Figure 5.20 shows the mean Reaction Time to Errors for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 16.3385$, p-value $= 8.1487e$-12, ($SD_{pooled} = 0.32$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

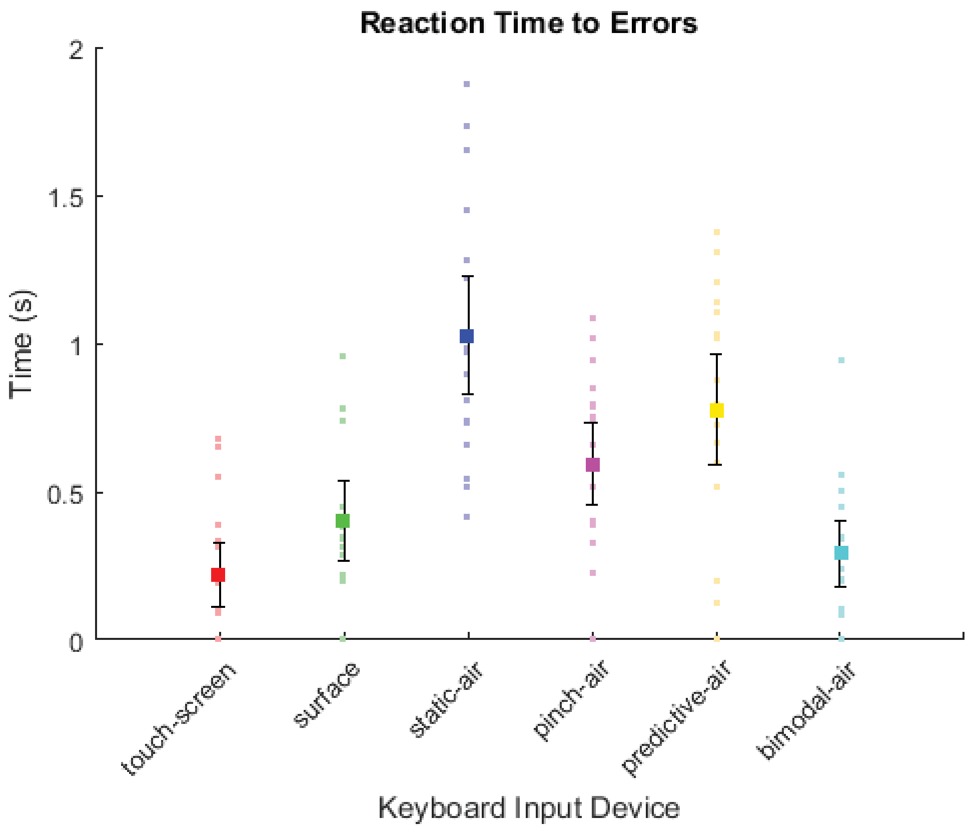


Figure 5.20. Mean Reaction Time to Errors for each keyboard with error bars showing 95% confidence intervals.

The multiple comparisons, seen in Table B.19, revealed significant differences in Reaction Time to Errors between the Touch Screen Keyboard and the Leap Static-Air, Pinch-Air, and Predictive-Air keyboards, all with p-values < 0.01. There were significant differences between the Leap Surface Keyboard and the Static-Air and

Predictive-Air keyboards both with p-values < 0.01. There were also significant differences found between the Leap Bimodal-Air and the Static-Air and Predictive-Air keyboards, both with p-values < 0.001. Finally, there was a significant difference found between the Pinch-Air and Static-Air keyboards with a p-value $= 0.0017$.

As expected, keyboards involving the 3rd-Dimension, increasing complexity, had significantly slower response times to errors than keyboards that did not. This implies that these keyboards required a higher level of focus as well as increased decoupling between the motor space and display.

5.5.3 Reaction Time to Simulate Touch

Figure 5.21 shows the mean Reaction Time to Simulate Touch for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 19.6476$, p-value $= 1.1541e\text{-}13$, ($SD_{pooled} = 0.54$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

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

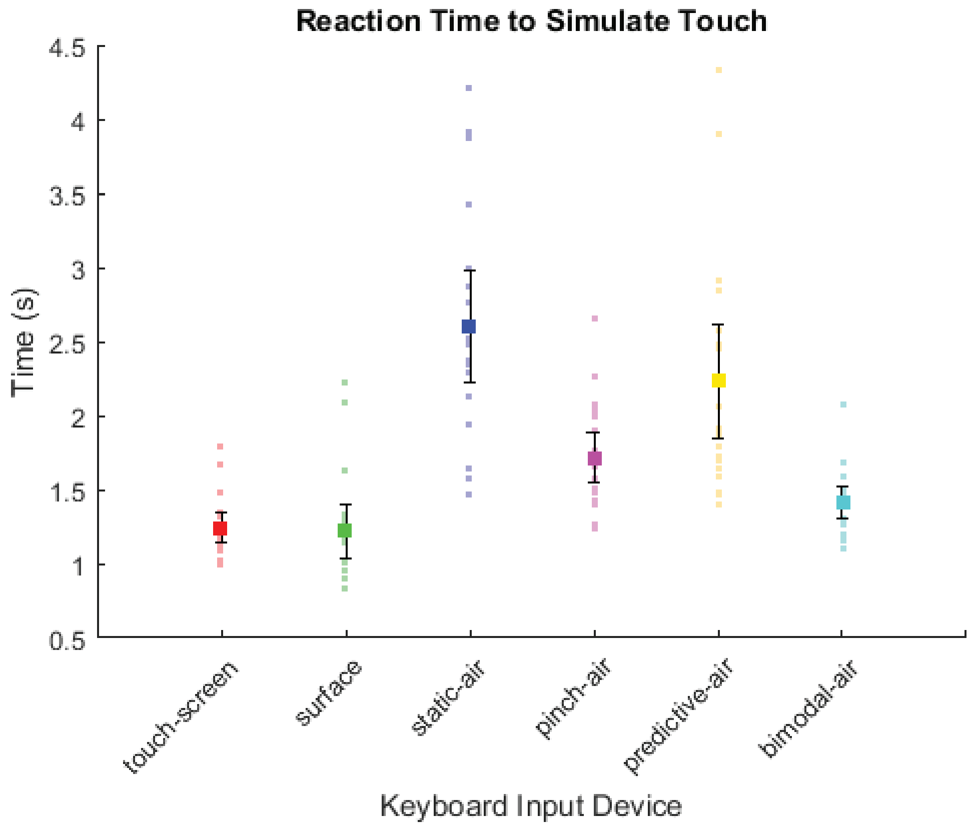


Figure 5.21. Mean Reaction Time to Simulate Touch for each keyboard with error bars showing 95% confidence intervals.

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.

5.5.4 Reaction Time for First Correct Letter

Figure 5.22 shows the mean Reaction Time for First Correct Letter for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) = 18.3416$, p-value $= 5.9470e\text{-}13$, ($SD_{pooled} = 0.95$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

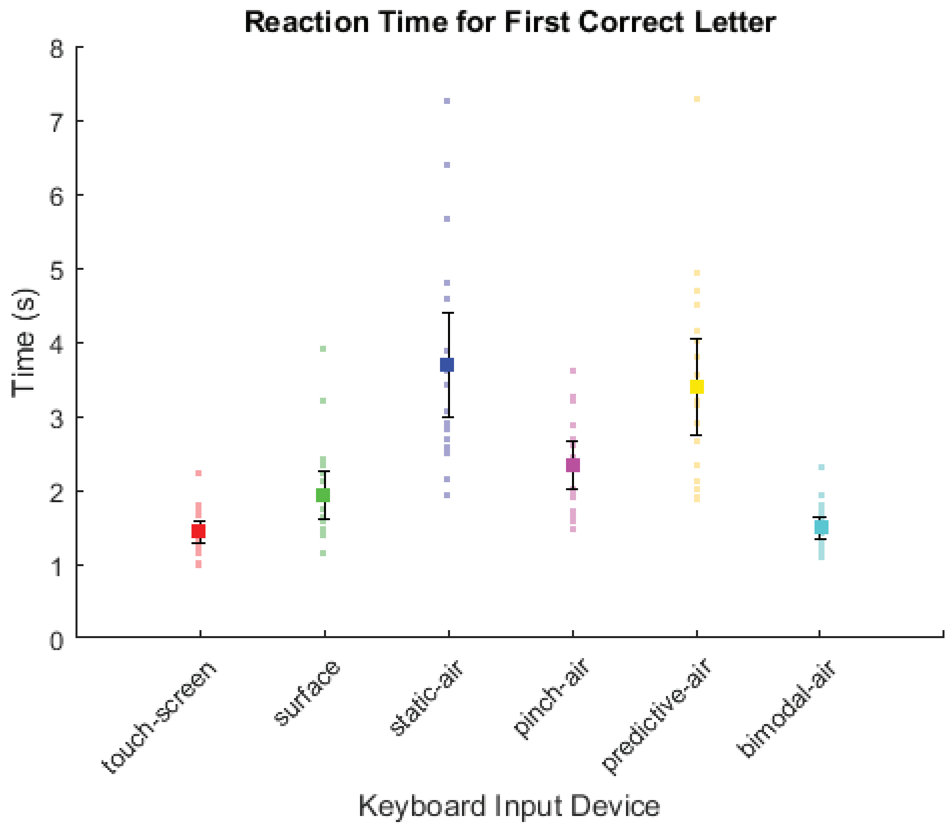


Figure 5.22. Mean Reaction Time for First Correct Letter for each keyboard with error bars showing 95% confidence intervals.

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

The Leap Bimodal-Air Keyboard performed at almost exactly the same rate as the Touch Screen Keyboard for hitting 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 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 and would have 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.

5.6 Quantitative Measures

5.6.1 Number of Touches Simulated

Figure 5.23 shows the mean Number of Touches Simulated for each keyboard input device. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means, $F(5, 102) =$

10.0290, p-value = 7.9270e-08, (SD_{pooled} = 1.28), prompting the use of Tukey's HSD with multiple-compare for further analysis.

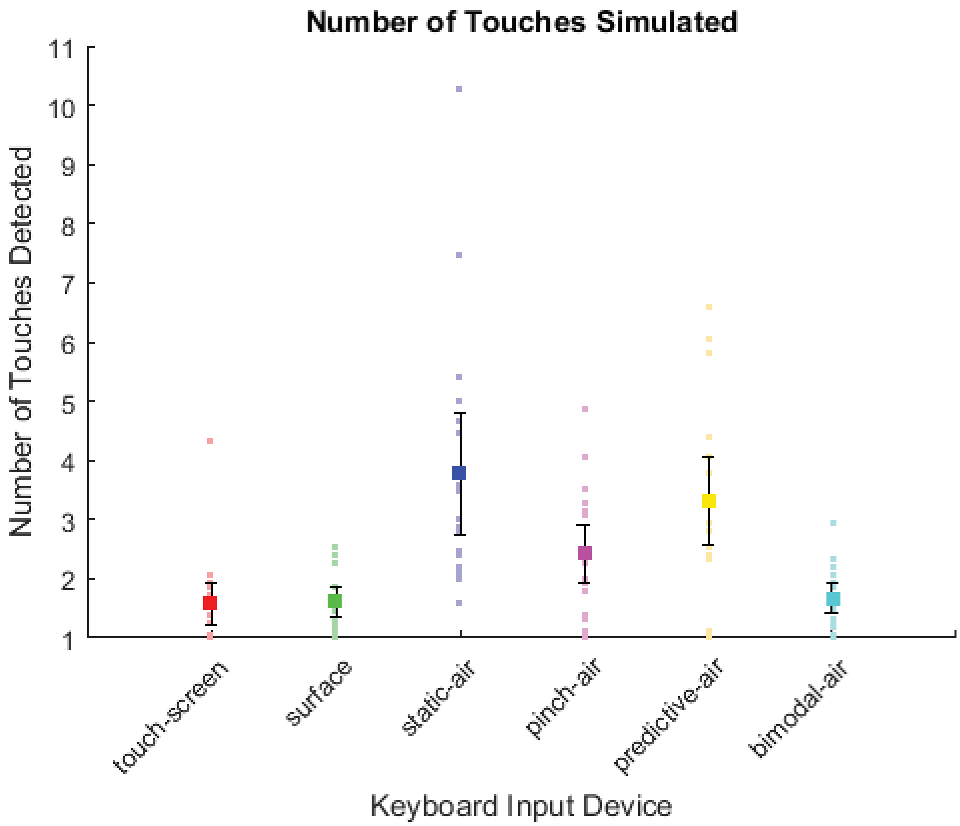


Figure 5.23. Mean Number of Touches Simulated for each keyboard with error bars showing 95% confidence intervals.

The multiple comparisons, seen in Table B.22, revealed significant differences in Number of Touches Simulated between the Touch Screen Keyboard and the Leap Static-Air and Predictive-Air keyboards, both with p-values < 0.01. There were significant differences between the Leap Surface Keyboard and the Static-Air and Predictive-Air keyboards both with p-values < 0.01. There were also significant differences found between the Leap Bimodal-Air and the Static-Air and Predictive-Air keyboards, again both with p-values < 0.01. Finally, there is a significant difference between the Pinch-Air keyboard and the Static-Air keyboard with a p-value = 0.0245.

Again, the same trends as before are seen. The Leap Surface and Bimodal-Air keyboards perform 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 by the researcher that the Static-Air keyboard suffered from a sort of "skimming" problem. The "skimming" problem is where participants would reach just 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, due to natural arcing in the hand motion, participants' hands would pull away from the interaction plane. This issue is expected to also be a culprit for some of the increased error rates.

5.6.2 Number of Practice Words

Figure 5.24 shows the mean Number of Practice Words for each keyboard input device. 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. Using a One-Way Analysis of Variance, 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 were participants who would perform a full set of 15 words for each keyboard. Rarely, participants felt the need to perform more than one practice set worth of words. The second category were participants who progressively performed less practice words for each consecutive keyboard, this is an example of why a Replicated Latin Squares design for counterbalancing was used.

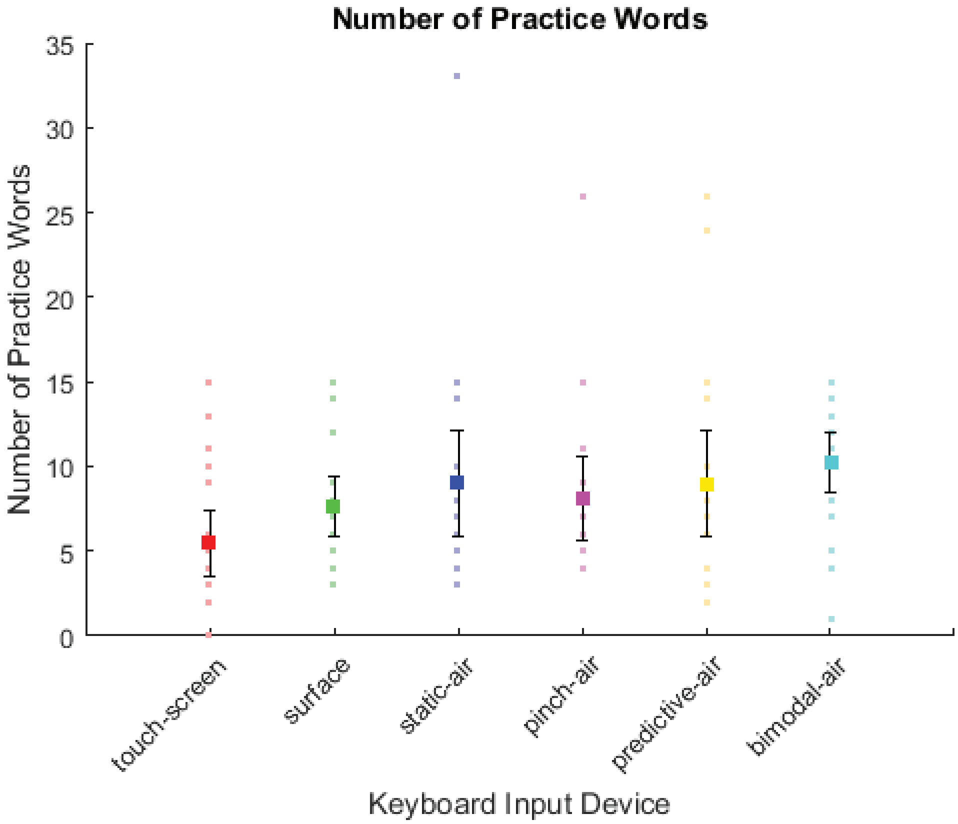


Figure 5.24. Mean Number of Practice Words for each keyboard with error bars showing 95% confidence intervals.

5.7 Qualitative Measures

5.7.1 Level of Discomfort

Figure 5.25 shows the median Level of Discomfort for each keyboard input device rated by the participants. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means for Level of Discomfort, $F(5, 102) = 7.5000$, p-value $= 4.9726e$-06, ($SD_{pooled} = 0.94$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

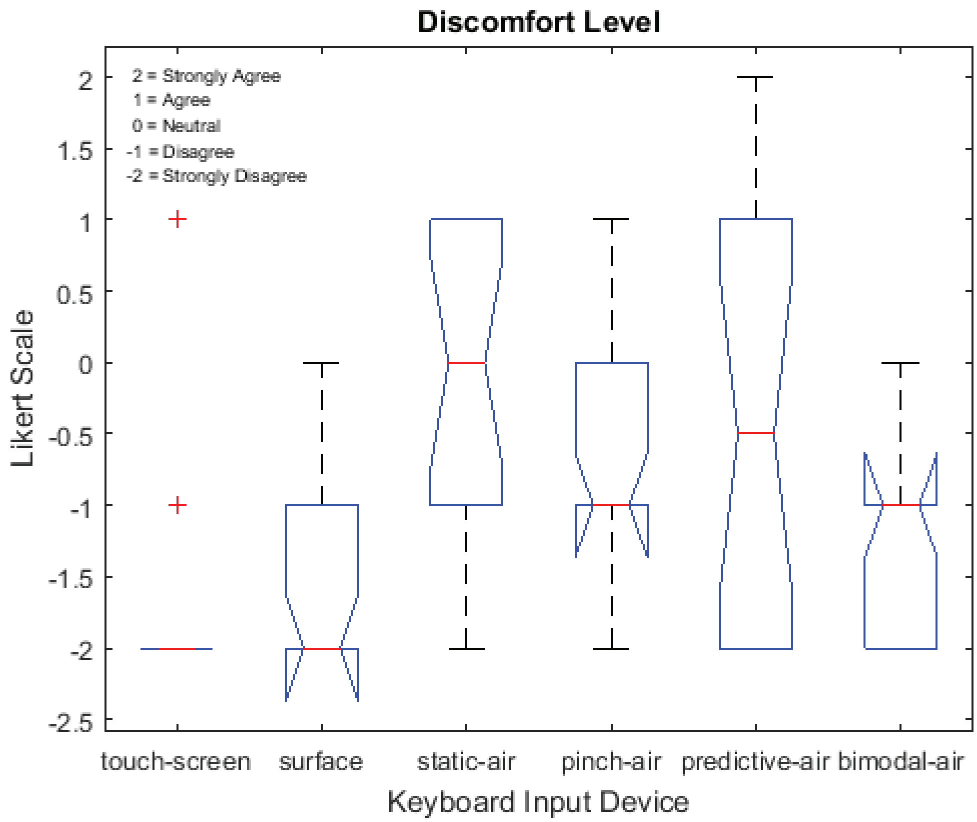


Figure 5.25. Median Level of Discomfort for each keyboard showing the 25th and 75th percentiles.

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

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 on the mid-air keyboards, and more extremely by the keyboards that utilized the 3rd-Dimension.

5.7.2 *Level of Fatigue*

Figure 5.26 shows the median Level of Fatigue for each keyboard input device rated by the participants. 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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means for Level of Discomfort, $F(5, 102) = 10.9910$, p-value $= 1.7659e$-08, ($SD_{pooled} = 1.12$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table B.24, revealed significant differences in Level of Fatigue between the Touch Screen Keyboard and the Leap Static-Air, Pinch-Air, and Predictive-Air keyboards, all with p-values < 0.01. There were significant differences between the Leap Surface Keyboard and the Static-Air, Pinch-Air, and Predictive-Air keyboards all with p-values < 0.05. There was also a significant difference found between the Leap Bimodal-Air and the Static-Air keyboards with a p-value $= 0.0043$.

As expected, the fatigue levels experienced for mid-air keyboards were typically higher than those that were not mid-air.

5.7.3 *Level of Difficulty*

Figure 5.27 shows the median Level of Difficulty for each keyboard input device rated by the participants. Keyboards reached a median Level of Difficulty of

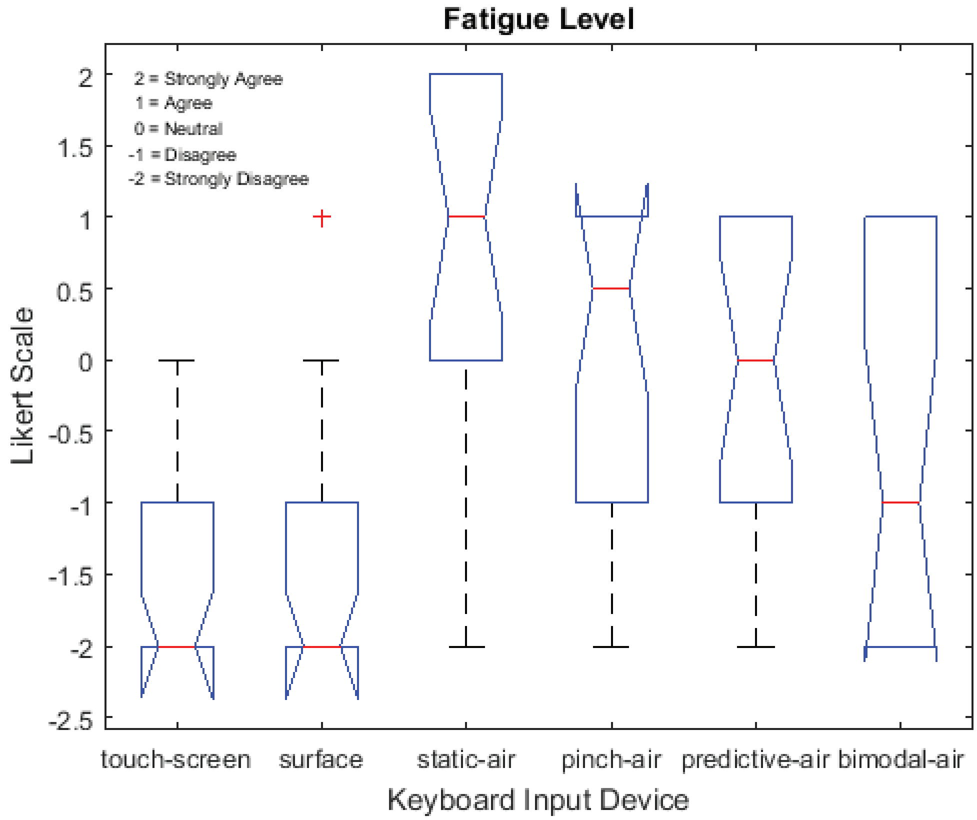


Figure 5.26. Median Level of Fatigue for each keyboard showing the 25th and 75th percentiles.

"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. Using a One-Way Analysis of Variance, a significant difference was detected between keyboard means for Level of Discomfort, $F(5, 102) = 6.0351$, p-value $= 6.2051e$-05, $(SD_{pooled} = 0.98)$, prompting the use of Tukey's HSD with multiple-compare for further analysis.

The multiple comparisons, seen in Table B.25, revealed significant differences in Level of Difficulty between the Touch Screen Keyboard and the Leap Static-Air, Pinch-Air, and Predictive-Air keyboards, all with p-values < 0.05. There was significant difference between the Leap Surface and the Static-Air keyboards with a p-values

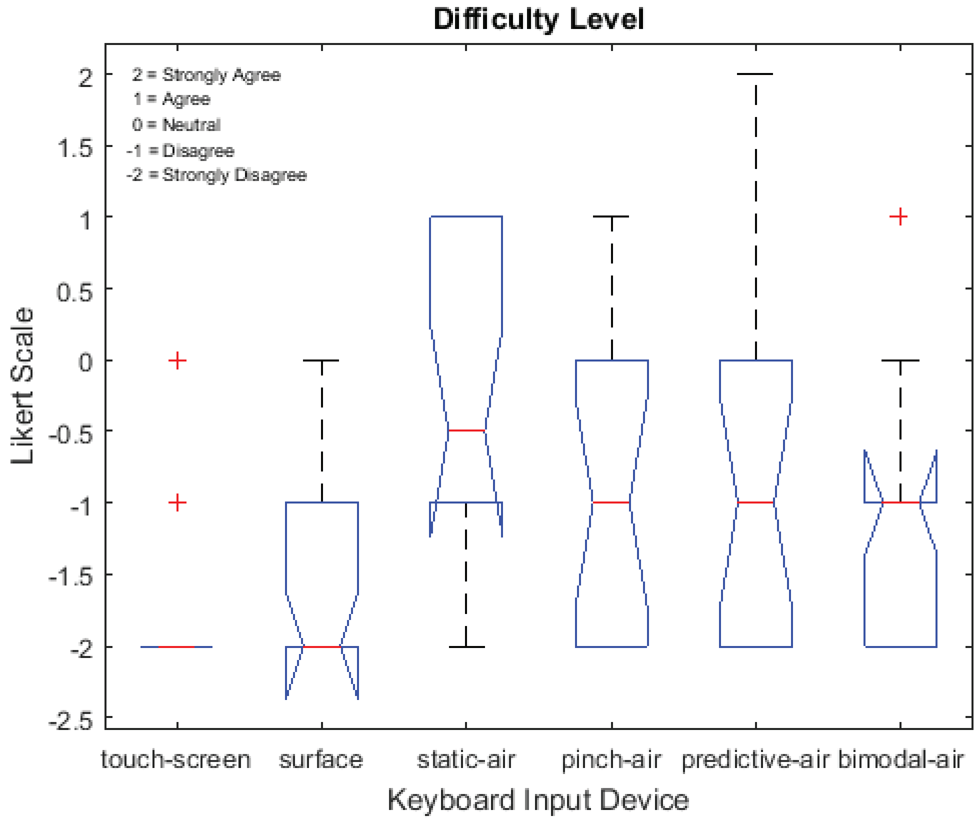


Figure 5.27. Median Level of Difficulty for each keyboard showing the 25th and 75th percentiles.

= 0.0042. There was also a significant difference found between the Leap Bimodal-Air and the Static-Air keyboards with a p-value = 0.0209.

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.

5.7.4 Preference Ranking

Figure 5.28 shows the mean Preference Ranking for each keyboard input device. 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. Using Friedman's rank-test, Analysis of Variance, a significant difference was detected between keyboard means, $\chi^2(5, 85) = 49.1429$, p-value $= 2.0749e$-09, ($SD_{pooled} = 1.87$), prompting the use of Tukey's HSD with multiple-compare for further analysis.

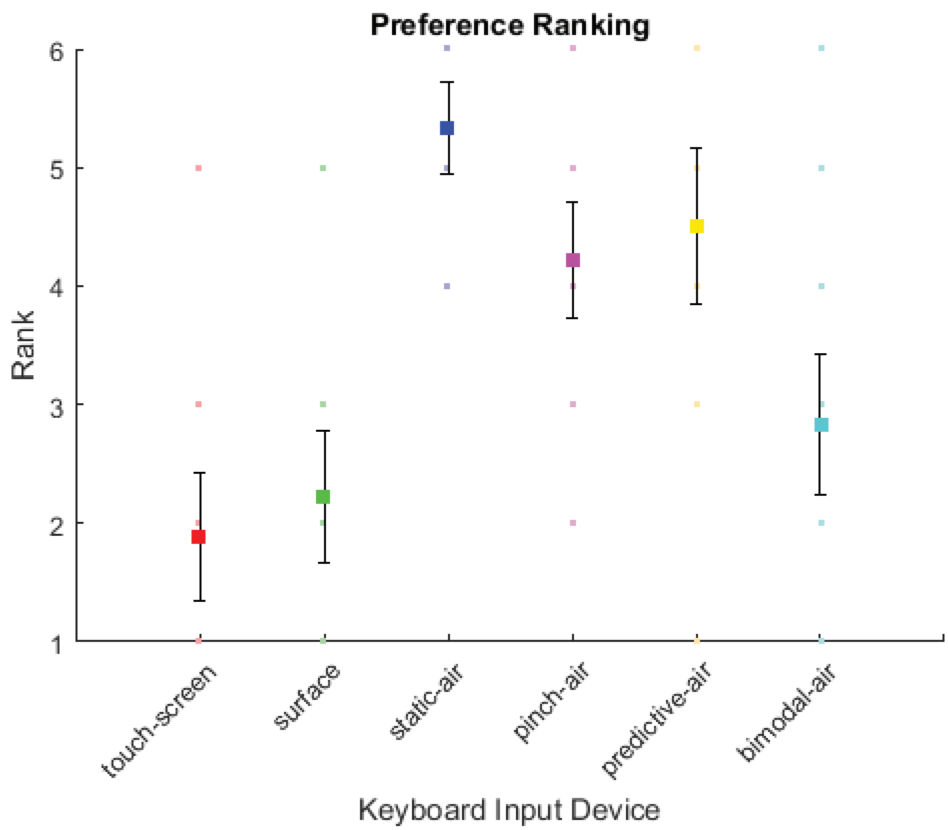


Figure 5.28. Mean Preference Rankings for each keyboard with error bars showing 95% confidence intervals.

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

The preference ranking of all of the keyboards very strongly reflects 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 SIX

Discussion, Future Work and Conclusion

6.1 Discussion

Vulture has shown that word-gesture keyboards are beneficial to mid-air text-entry by reaching a text-entry rate of 28 WPM, surpassing earlier work, however it utilizes pinching as a means of word separation (Markussen, Jakobsen, and Hornbæk 2014). This study aimed to find alternative solutions to word separation and interaction with a mid-air, word-gesture keyboard. Rather than pinching or inconvenient glove-use, this study utilized the 3rd-Dimension as well as a bimodal approach.

This study suffered from some of the same issues that were experienced in Vulture, text-entry rates were much slower for mid-air keyboards than the Touch Screen Keyboard. One reason for this is that participants are required to recouple the gestures in motor space with those on the display (Markussen, Jakobsen, and Hornbæk 2014). This affect was described in detail for many of the results in Chapter 5. Another reason, as seen in Vulture and in other studies (Markussen, Jakobsen, and Hornbæk 2014, Witt, Lawo, and Drugge 2008), participants heavily rely on the displayed feedback. This was seen mostly due to the small latency introduced when using the Leap Motion to track and display hand motions on the screen. Some participants commented having to slow down to ensure the display more accurately represented their hand movements. If movements were made that were too fast, the visual display would slightly lag behind and so some participants had a tendency to overshoot letters. A final reason, like in Vulture, is that many of the alternatives add to the complexity of text-entry by the forcing explicit delimiting of words.

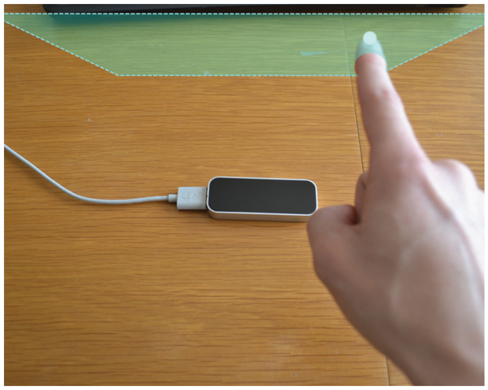
6.1.1 Pinching Interaction

This study utilized a pinching method using the Leap Motion in order to have a metric for comparison against Vulture's pinching since this study used a pseudo-implementation of word-gesturing, lacking word-recognition. 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). This gave a solid measure for alternative solutions to compare against.

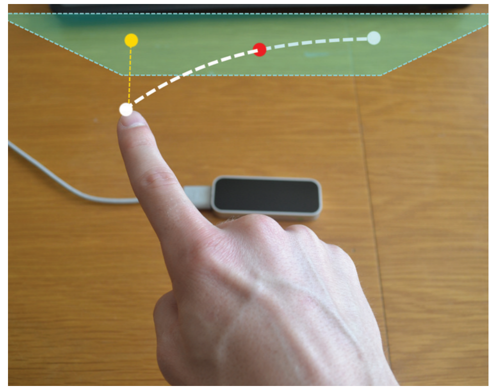
6.1.2 3-Dimensional Interaction

This study 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 the difficulty in mentally coupling word-gestures in motor space with those displayed on the screen with the added factor of having to move in the 3rd-Dimension and interact with an invisible plane to delimit words.

The Predictive-Air Keyboard was not significantly worse than the Pinch-Air Keyboard and needs to be investigated further 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 6.1. The "skimming" issue is when a participant reaches towards the interaction plane and simulates a touch but as they move their hand during the gesturing motion, natural arcing of the participant's hand causes them to lose the simulated touch. The natural arcing motion of the participants also seemed to cause a lot of errors after finishing writing a word because participants would pull their hand back in an arcing motion, especially if their elbow was rested. This would cause their finger to hit the key above the current key they were on and count it as a press, as seen in Figure 5.5.



(a) Pressing Key



(b) Moving to next Letter across Keyboard

Figure 6.1. 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. **(b)** shows the intended destination **yellow** and the accidental release **red**.

It's very interesting to note that the Static-Air implementation was used to project a mid-air keyboard onto flat surfaces, simulating a touch screen. This was the Leap Surface Keyboard. 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 it avoid having a decoupled motor space and display space. This hints that the Static-Air Keyboard may substantially benefit from an input plane displayed using augmented reality.

6.1.3 Bimodal Interaction

Bimodal mid-air interactions are shown to be very promising in this study and it should be noted that any other secondary input can 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 is a significant improvement over using pinching or utilizing the 3rd-Dimension. The Bimodal-Air Keyboard was also often times indistinguishable from the Touch Screen Keyboard for many other dependent measures, all detailed in

Chapter 5. The Bimodal-Air keyboard needs to be further investigated in a repeated-measures, multiple session study on a word-gesture keyboard implemented with word-recognition.

The Bimodal-Air isn't without some of it's own drawbacks though, as experienced with the other mid-air keyboards using the Leap Motion, there is an associated plane that the gesture is moving across as the participant moves their hand. This plane, if calibrated or oriented incorrectly, can lead to higher error rates and less precision overall.

6.1.4 Lacking Word-Recognition

There are some glaring limitations to using a pseudo-implementation of word-gesturing as opposed to using a full word-recognition implementation. A major differences was that the best-guess implementation analyzed the gesture as it was being created, showing participants real-time updates of what was happening to the path they were creating. As the gesture was being created, the system attempted to guess the current character being pressed based on the deviations in the gesture and then compared those deviations with the current characters that the participant was trying to achieve and against the previously detected deviation. The limitation that this imposes, seeing real-time character-level updates, is that most participants would interrupt their current word-gesture in order to fix the errors being created mid-gesture which is unnatural for word-gesturing. As well, the fidelity for making errors was much higher, allowing these interruptions to sometimes occur frequently. The lack or word-recognition means that words that aren't real could be produced through gesturing which again is counter-intuitive and once one mistake was made, the likelihood to see more errors increased.

6.2 Future Work

There are many things to consider when looking at improves to be made or additional studies that can be conducted.

6.2.1 Implement Word-Recognition

The most important change to this study that needs to be explored is to re-conduct the study using a standardized word-gesture keyboard implementation with word-recognition. This single improvement will give better and more standardized results across the board. There would also be no need for any modified variables, and less error measures could have been observed. There should also be an increase in text-entry rates since participants won't be distracted with errors mid-gesture.

6.2.2 Redesign Task

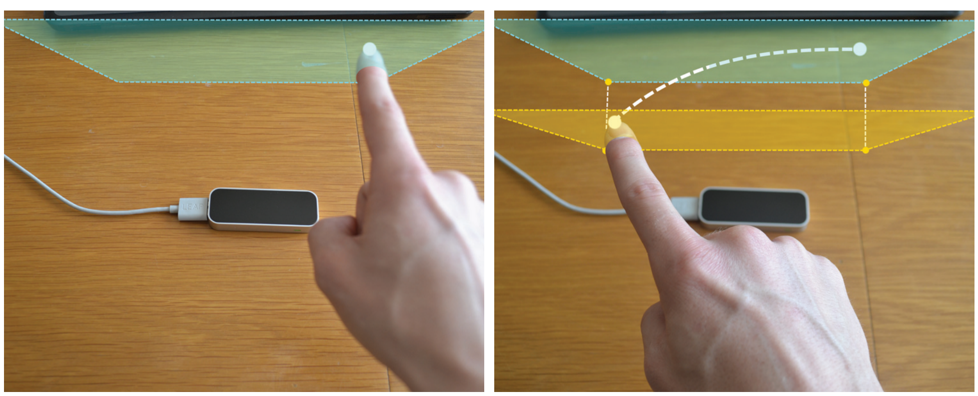
The task for the study needs to be substantially redesigned to fit a word-recognition implementation of the word-gesture keyboards. First, the trials need to be phrases instead of single words to better track text-entry and error rates. Next, there shouldn't be any requirement to fix words, it should just feel natural for the participant and they can fix errors where they feel it's necessary. The number of trials needs to be increased but a significant amount and as well there study needs to be designed for repeated sessions so that improvements in text-entry rate can be seen. With repeated sessions and actual word-recognition, the Bimodal-Air keyboard is expected to achieve text-entry rates superior to those found in Vulture.

6.2.3 Augmented Reality

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

6.2.4 The "Skimming" Issue

As a first thought to solve the "skimming" issue, there is a seemingly obvious solution to the problem. Simply, as shown in Figure 6.2, once a simulated touch has been made, increase the interaction plane's threshold in the participant's direction, enough so that the participant's hand doesn't leave the surface when moving from side to side or up and down. This change should see improved results for the Static-Air Keyboard since there would be many less times where the user accidentally exits the interaction plane. A similar approach could also be used for the Predictive-Air because sometimes quick side to side movements were seen as a touch being released.

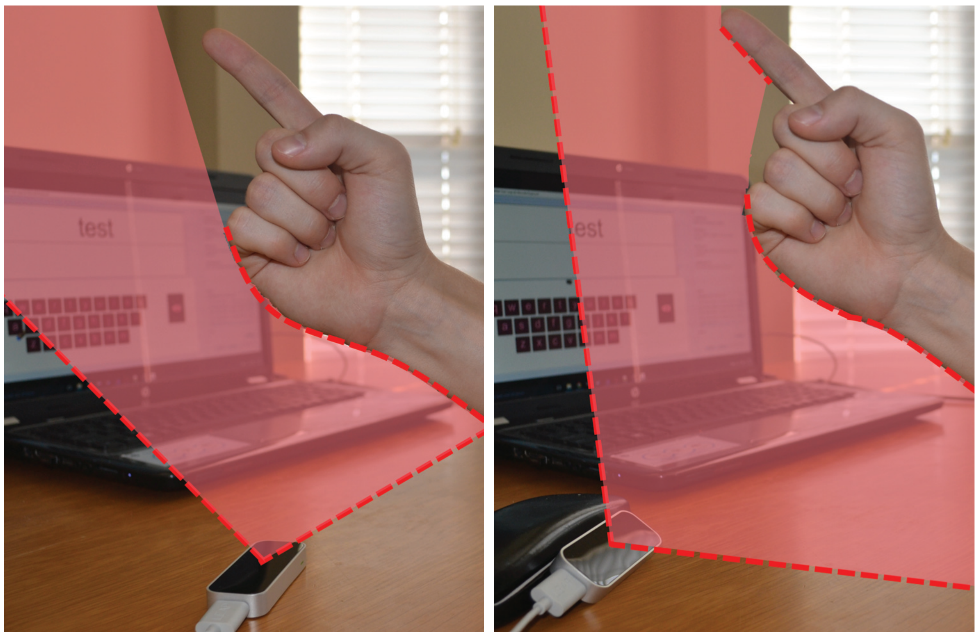


(a) Intersecting Plane to hit First Key (b) Increased Release Threshold

Figure 6.2. An example of how the "skimming" issue can be fixed by increasing the release threshold after touching. **(a)** shows the user pressing the first key. **(b)** shows the increased release threshold for the interaction plane as the user moves across to the next key.

6.2.5 Better Tracking

A simple observation that was made by the researcher is that there were many times that the Leap Motion had issues detecting people's hands and palm. The main reasoning for this was the positioning of the Leap Motion itself as well as some participants holding their hand in an upward position, like shown in Figure 6.3a. A simple solution to this issue is to angle the leap so that so that it is facing the participant at a better angle, as seen in Figure 6.3b.



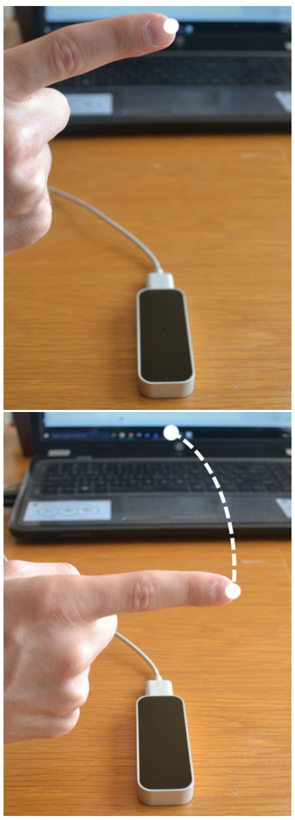
(a) Hand Blocking Finger (b) Angle Reveals Finger

Figure 6.3. 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.

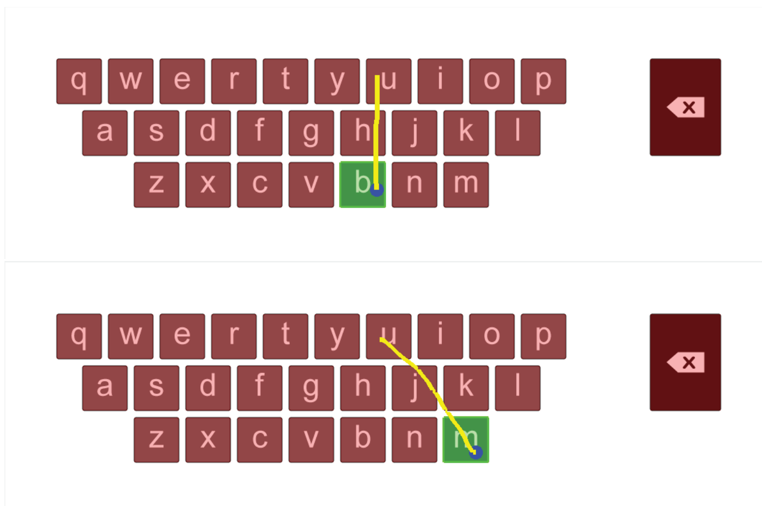
6.2.6 Alternative Interaction Plane

Something that was observed with all mid-air keyboards is that as the partcipants moved across the interaction plane, sometimes the side-to-side movements and up and down movements weren't represented as expected, as seen in Figure 6.4. Better approaches to calibration and plane-creation need to be used such as those used and mentioned in Personal Space and like methods (Hari Haran 2014, Guinness

2015). A spherical or even curved interaction plane designed for the individual participant would greatly benefit both the all of the mid-air keyboards especially so that participants would be able to rest their arms. 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.



(a) Movement



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

Figure 6.4. Example showing how a bad calibration affects movement. **(a)** shows a user trying to make a vertical movement. **(b)** shows how the actual path is off because of the calibration and orientation of the Leap Motion Controller.

6.2.7 Multi-functional Space

Utilizing the 3rd-Dimension for mid-air word-gesturing keyboards, when not touching the interaction plane to simulate touch, hand gestures can be used to control a cursor or for other gestures. This would allow users to fully interact with screens of any kind, including projections, and not limit the user to a keyboard and mouse.

6.2.8 Image Processing

After using the Leap Surface Keyboard and projecting a virtual keyboard over just a printed sheet, an addition that could be added to make the feature much more versatile is image processing. Being able to print and use any keyboard for word gesturing or otherwise could be interesting and highly beneficial.

6.2.9 Gaming Console Keyboards

An afterthought from using the Xbox Controller Keyboard in the pre-pilot and the pilot study, it would be very beneficial to apply what has been learned from this study and word-gesturing in general to gaming console keyboards. There are two options for implementing a word-gesture keyboard for gaming consoles. The first is to implement a mid-air word-gesture keyboard using the Xbox Kinect or using the Wii Remotes. The second option would be to transition from the standard console keyboards to a word-gesture keyboard just using standard console controllers. The word-gesture keyboard would most likely have to be bimodal. The user would hold the 'A' button while simulating touch and use the thumb stick to move a cursor around for word-gesturing. Either method would be a great improvement over single character text-entry currently seen on modern gaming consoles.

6.2.10 Accessibility

A major motivation for this research was the idea of applying it for amputees or those with disabilities that affect their performance using a standard keyboard. A proper study in accessibility should be performed to utilize the Bimodal-Air keyboard for users with disabilities.

6.3 Conclusion

Word-gesture keyboards show a promising means to efficient, mid-air text-entry by tracing word-gestures instead of slow, single-input text-entry (Markussen, Jakobsen, and Hornbæk 2014). This study, demonstrated alternative ways to delimit the separation of words for mid-air, word-gesture keyboards. 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 keyboard display. 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 complimented by an augmented reality design to recouple the gesture-space and keyboard display. Another method to lessen the severity of a decoupled gesture-space and keyboard display, the interaction plane needs to use new techniques to be better calibrated for each individual user and to display more accurate results and control (Hari Haran 2014, Guinness 2015). Finally, the empirical results from this study shows that using a bimodal technique as a means of word separation would greatly benefit mid-air, word-gesture keyboards. The Bimodal-Air keyboard was slower in text-entry rates than the Touch Screen Keyboard, however, the bimodal method was near indistinguishable from the touch screen for almost all other empirical results and was also better than pinching for nearly all results. This study needs to have a follow-up study with redesigned trials using a full word-gesture keyboard implementation with repeated measures and reoccurring sessions to further investigate bimodal techniques for mid-air text-entry. A bimodal approach just might be the future of mid-air text-entry for word-gesture keyboards.

APPENDICES

APPENDIX A

Surveys

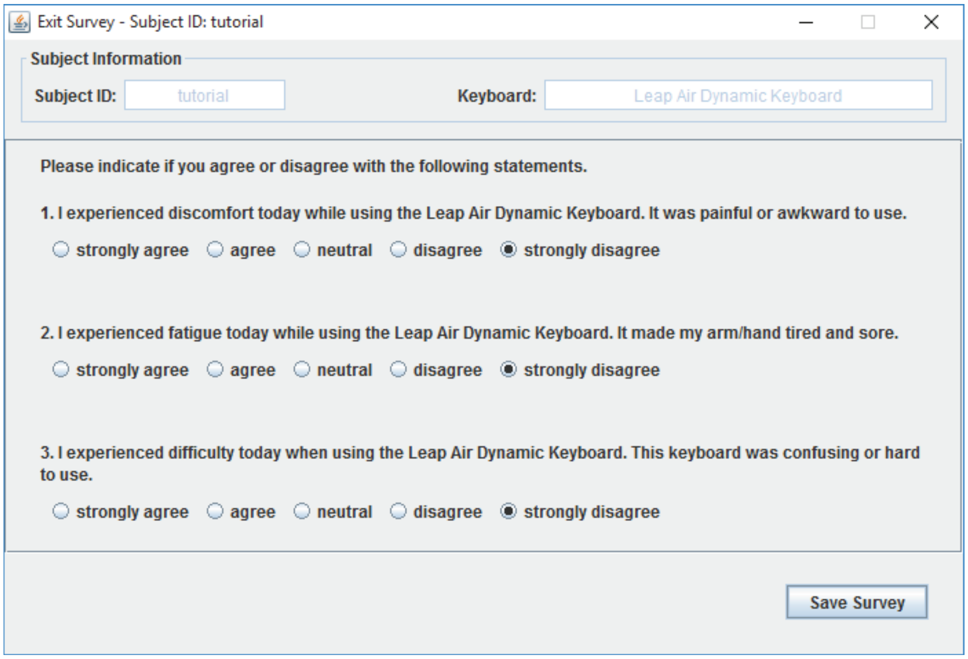
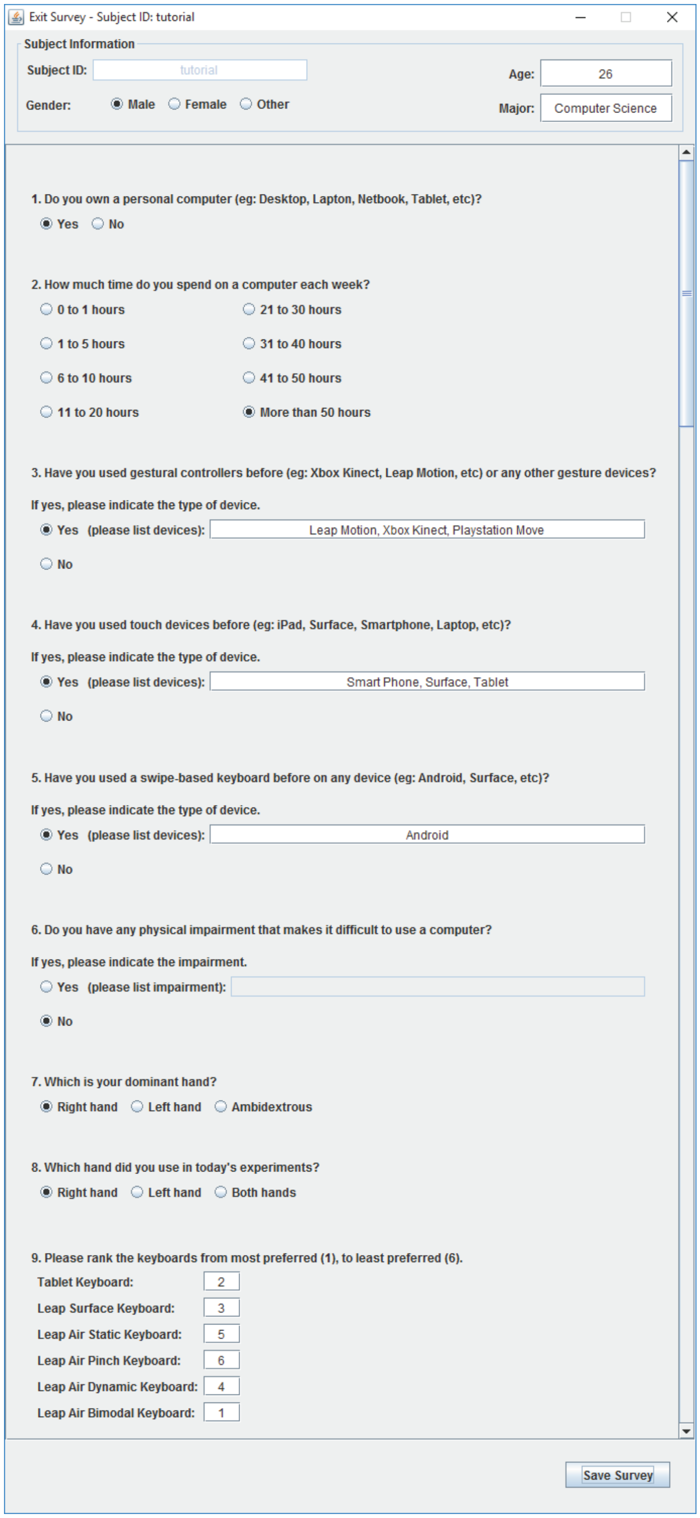


Figure A.1. Example of an intermittent survey that participants were required to fill out after finishing a task with a single keyboard.



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

APPENDIX B

Results Tables and Boxplots

The statistical methods used consisted of One-Way ANOVAs for each set of the dependent measures, followed by using Tukey's Honest Significant Difference (HSD) for multiple-compare for the post-hoc analysis. The ranking system from the exit survey used the Friedman's test for analysis in conjunction with Tukey's HSD for a post-hoc analysis.

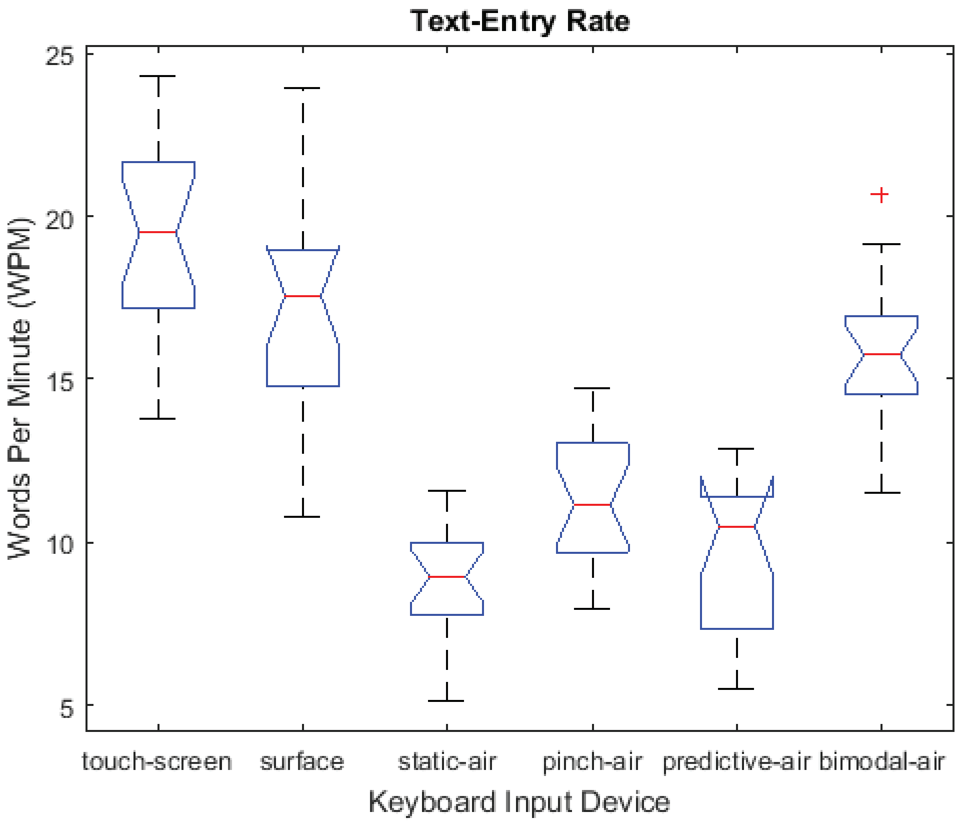


Figure B.1. Median Text-Entry Rates for each keyboard showing the 25th and 75th percentiles.

Table B.1. Tukey's Honest Significant Difference with multiple comparison for Text-Entry Rate.

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience 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

B.1.1 Text-Entry Rate Modified-Shortest

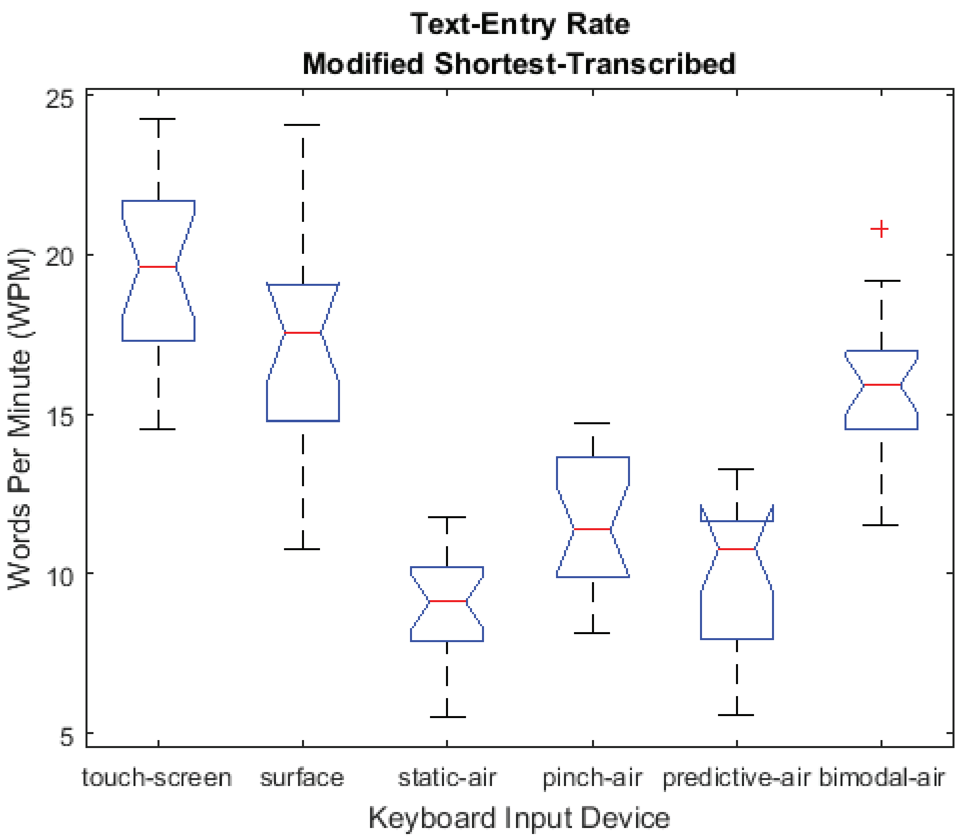


Figure B.2. Median Text-Entry Rates using the shortest-transcribed modification for each keyboard showing the 25th and 75th percentiles.

Table B.2. Tukey's Honest Significant Difference with multiple comparison for Text-Entry Rate modified with shortest-transcribed.

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience 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

B.2 Error Rates

B.2.1 Minimum Word Distance

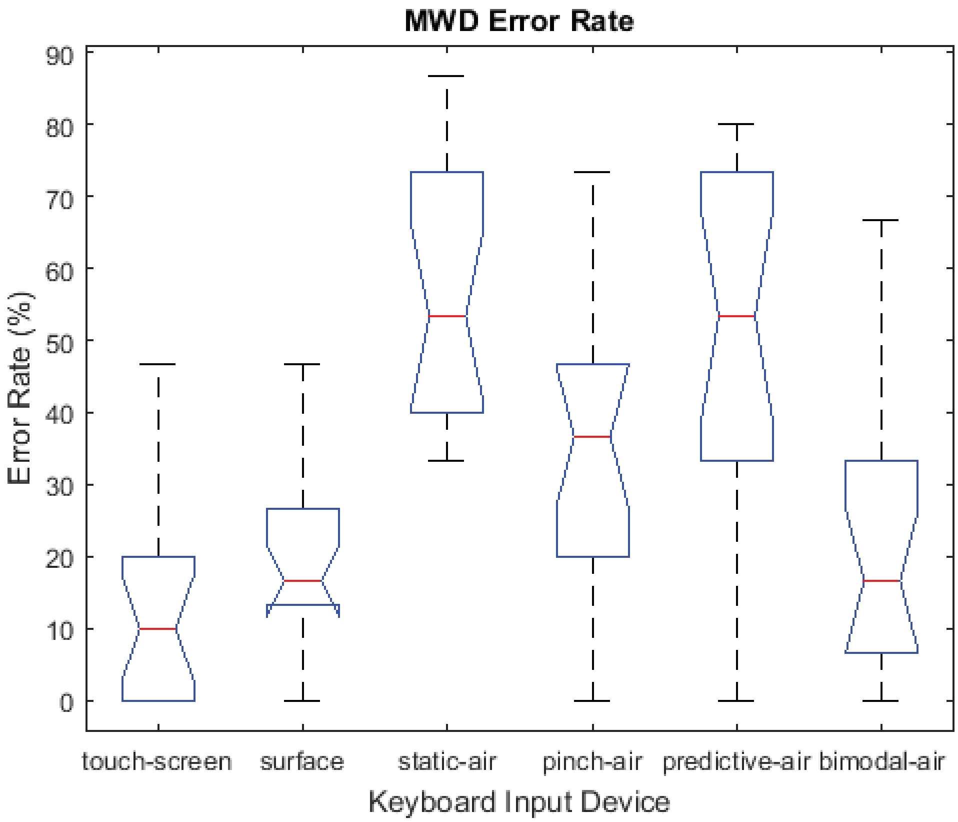


Figure B.3. Median Minimum Word Distance for each keyboard showing the 25th and 75th percentiles.

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

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience 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

B.2.1.1 MWD Modified-Shortest.

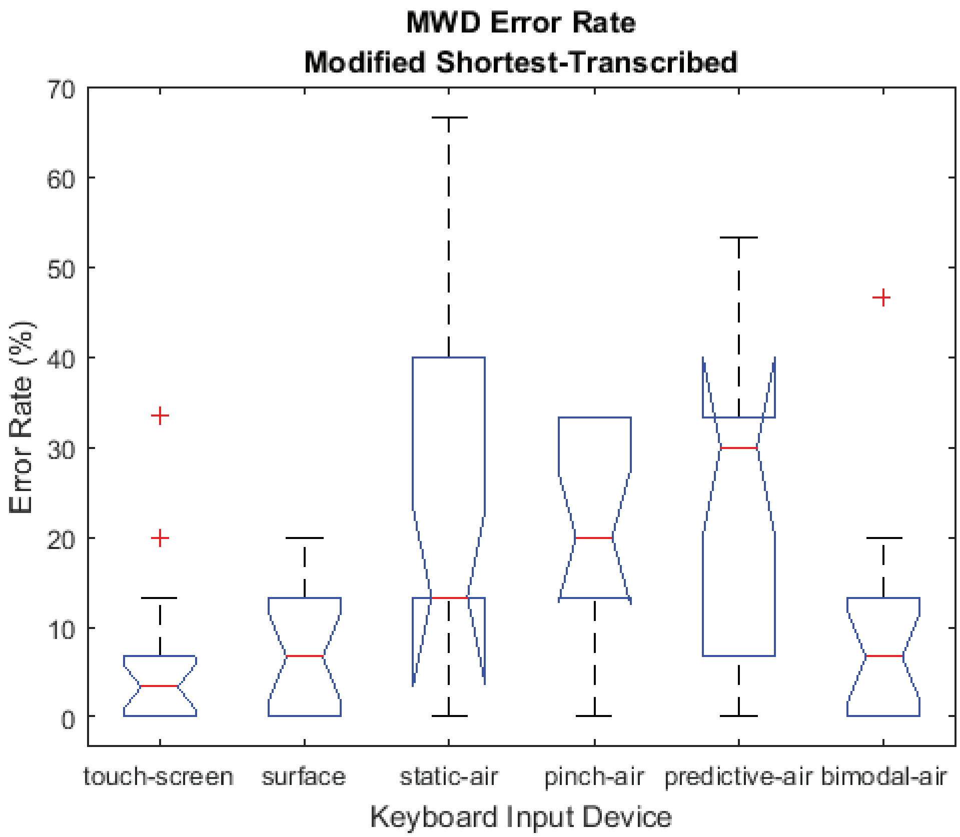


Figure B.4. Median Minimum Word Distance using the shortest-transcribed modification for each keyboard showing the 25th and 75th percentiles.

Table B.4. Tukey's Honest Significant Difference with multiple comparison for Minimum Word 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	-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

B.2.2 Keystrokes Per Character

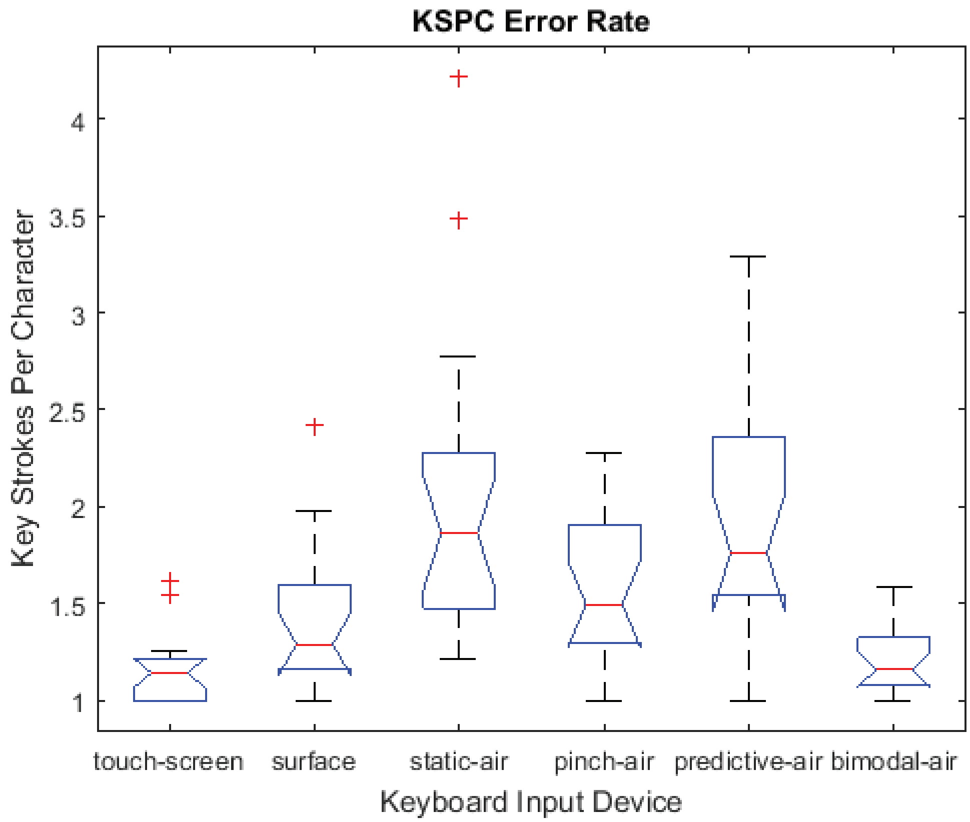


Figure B.5. Median Keystrokes Per Character for each keyboard showing the 25th and 75th percentiles.

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

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience 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

B.2.2.1 KSPC Modified-Shortest.

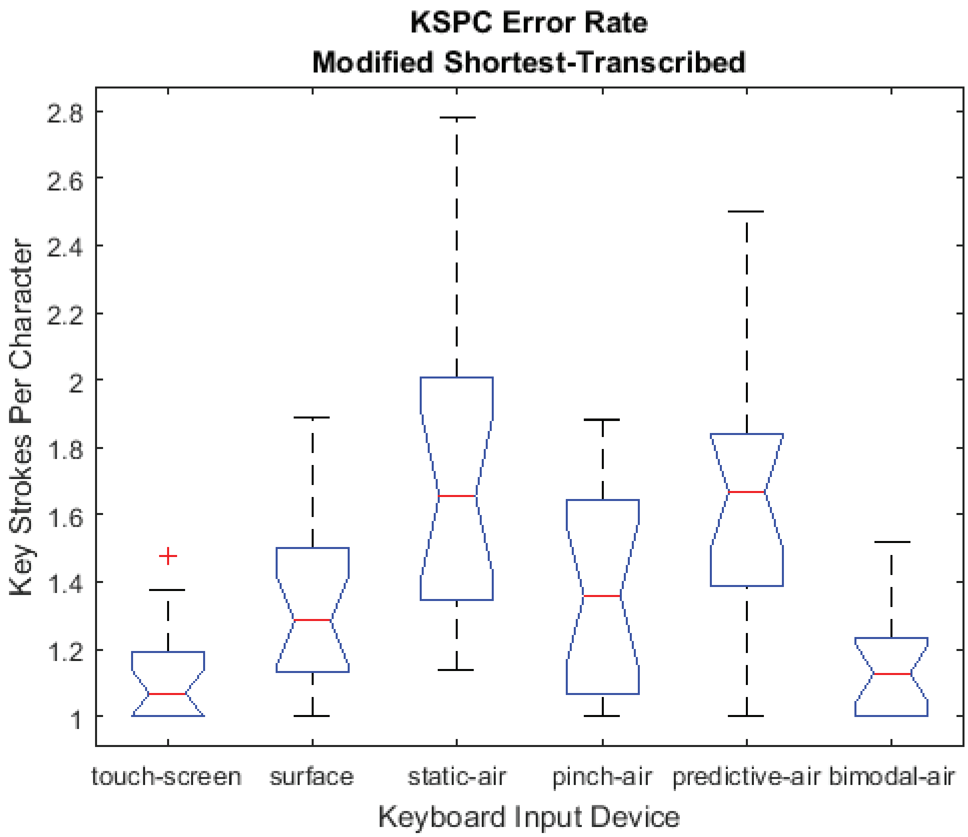


Figure B.6. Median Keystrokes Per Character using the shortest-transcribed modification for each keyboard showing the 25th and 75th percentiles.

Table B.6. Tukey's Honest Significant Difference with multiple comparison for Keystrokes Per Character 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	-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

B.2.3 Minimum String Distance

B.2.3.1 MSD Modified-Shortest.

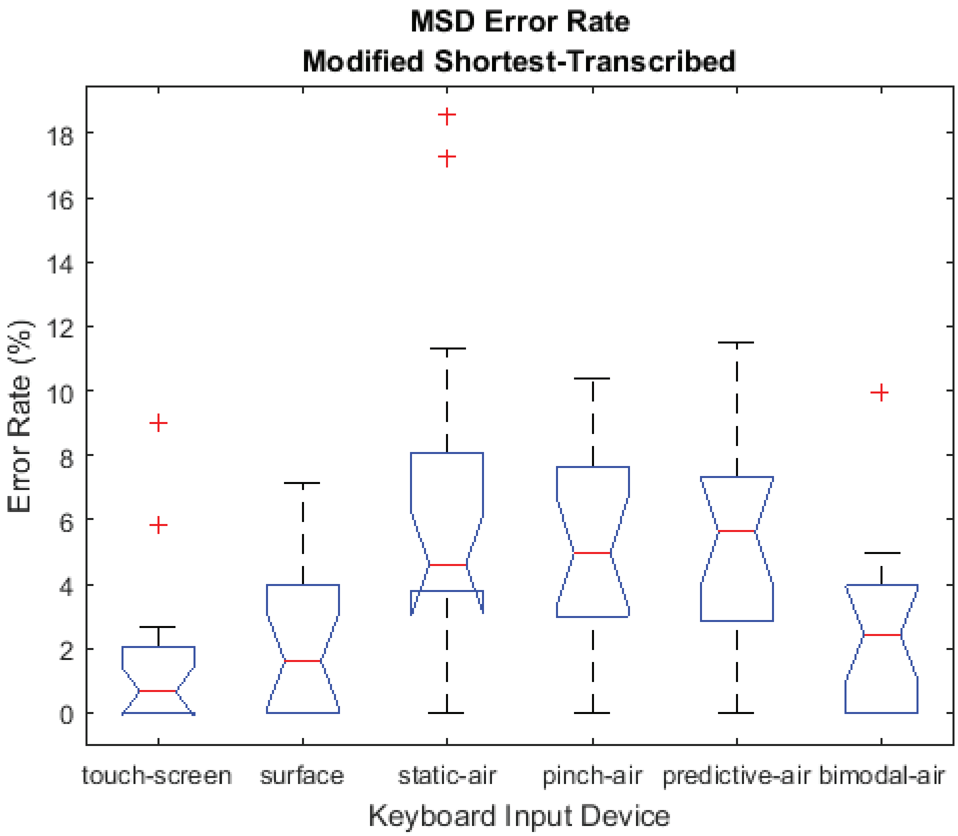


Figure B.7. Median Minimum String Distance using the shortest-transcribed modification for each keyboard showing the 25th and 75th percentiles.

Table B.7. Tukey's Honest Significant Difference with multiple comparison for Minimum String 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	-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

B.2.4 Total Error Rate

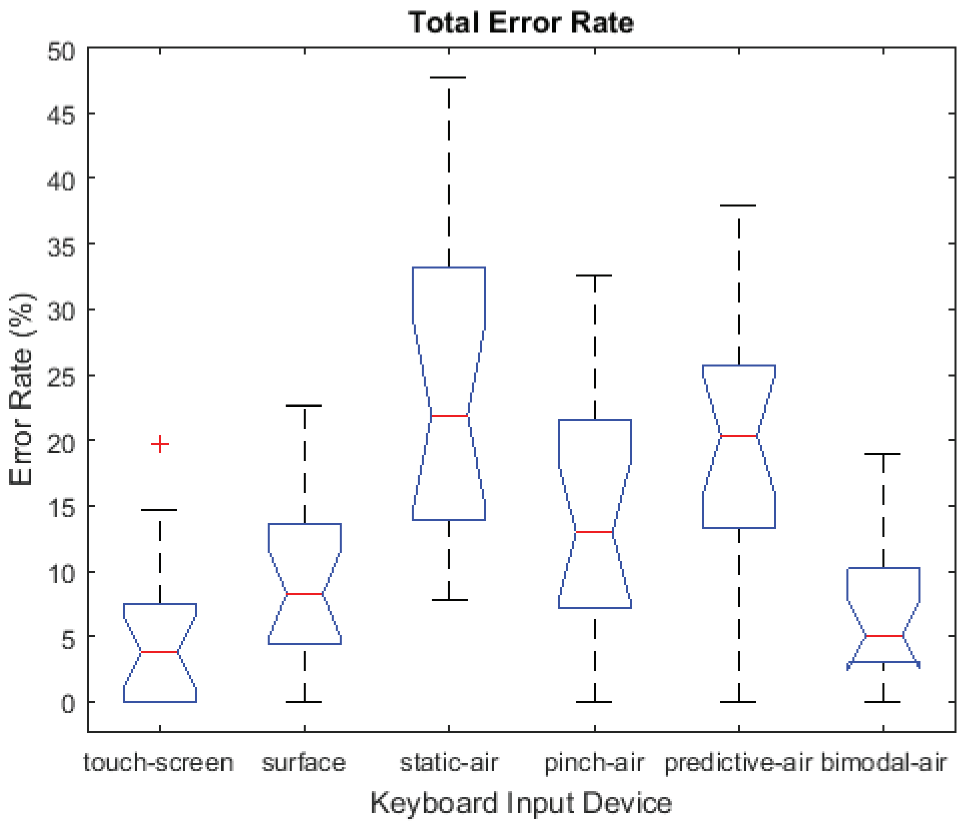


Figure B.8. Median Total Error Rate for each keyboard showing the 25th and 75th percentiles.

Table B.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

B.2.4.1 Total Error Rate Modified-Shortest.

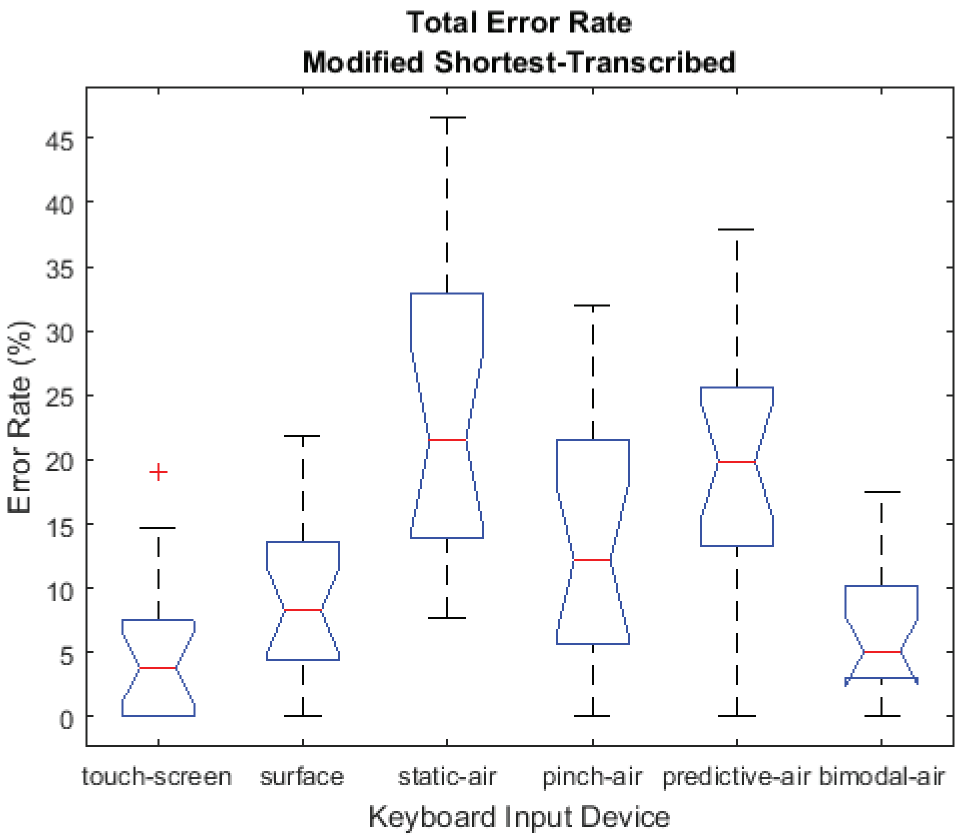


Figure B.9. Median Total Error Rate using the shortest-transcribed modification for each keyboard showing the 25th and 75th percentiles.

Table B.9. Tukey's Honest Significant Difference with multiple comparison for Total Error Rate 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	-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

B.3.1 Fréchet Distance

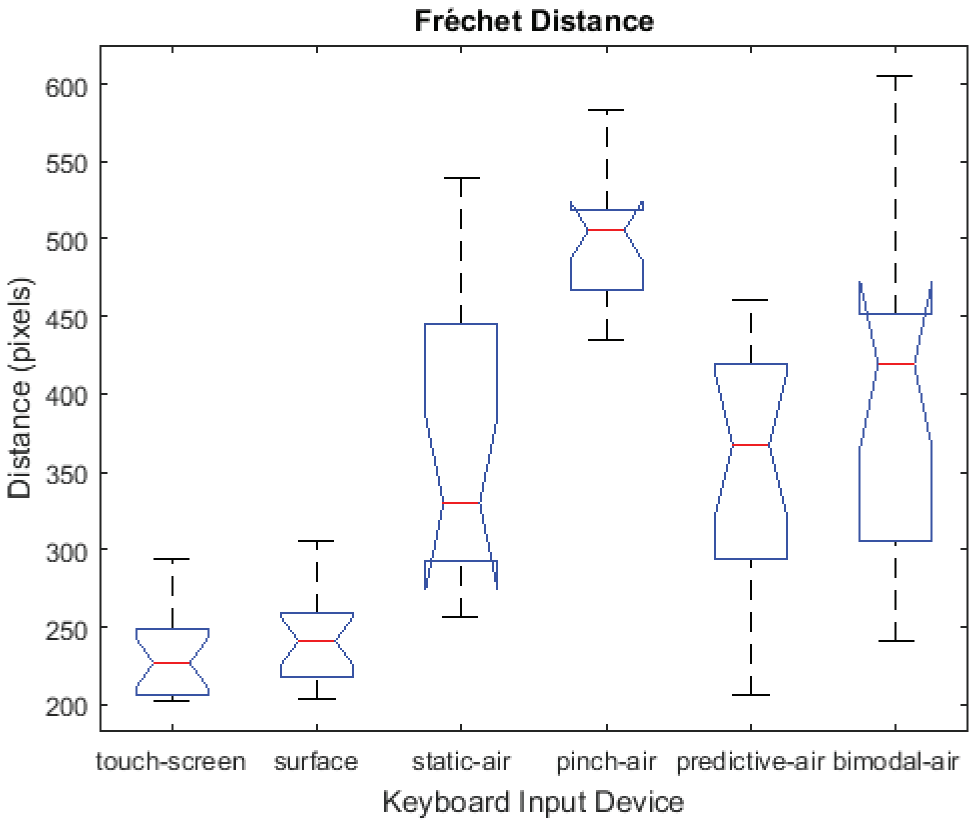


Figure B.10. Median Fréchet Distance for each keyboard showing the 25th and 75th percentiles.

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

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience 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

B.3.1.1 Fréchet Distance Modified-Shortest.

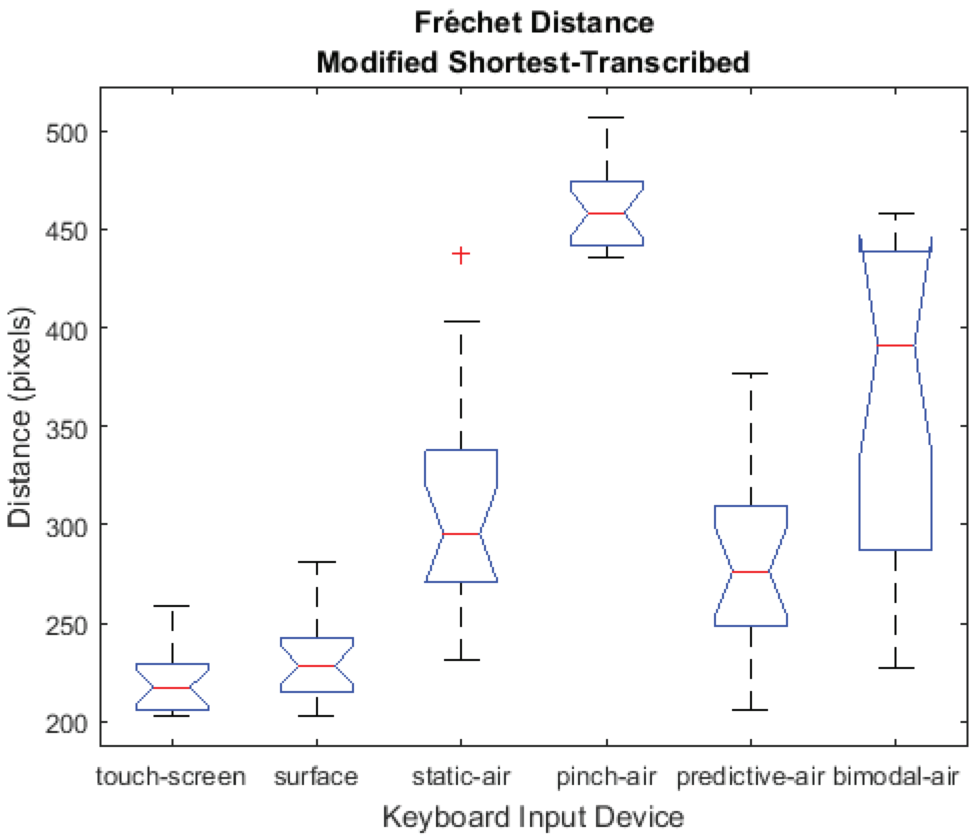


Figure B.11. Median Fréchet Distance using the shortest-transcribed modification for each keyboard showing the 25th and 75th percentiles.

Table B.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

B.3.1.2 Fréchet Distance Modified-Backspace.

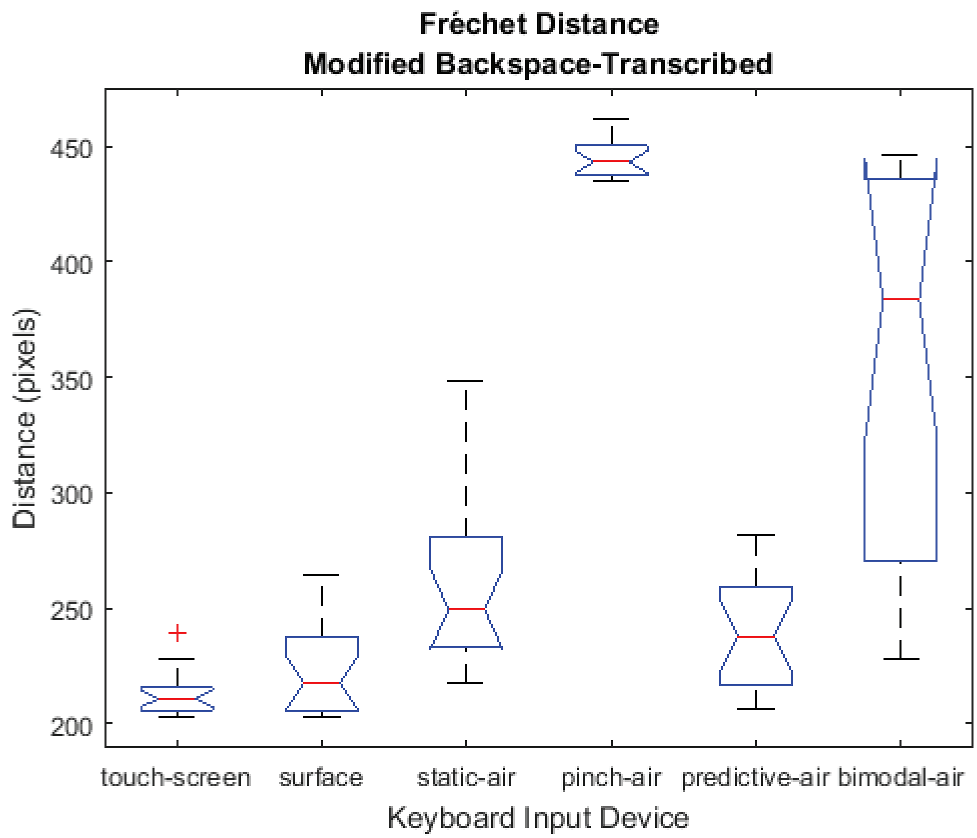


Figure B.12. Median Fréchet Distance using the backspace-transcribed modification for each keyboard showing the 25th and 75th percentiles.

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

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience 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

B.4.1 Word-Gesture Distance

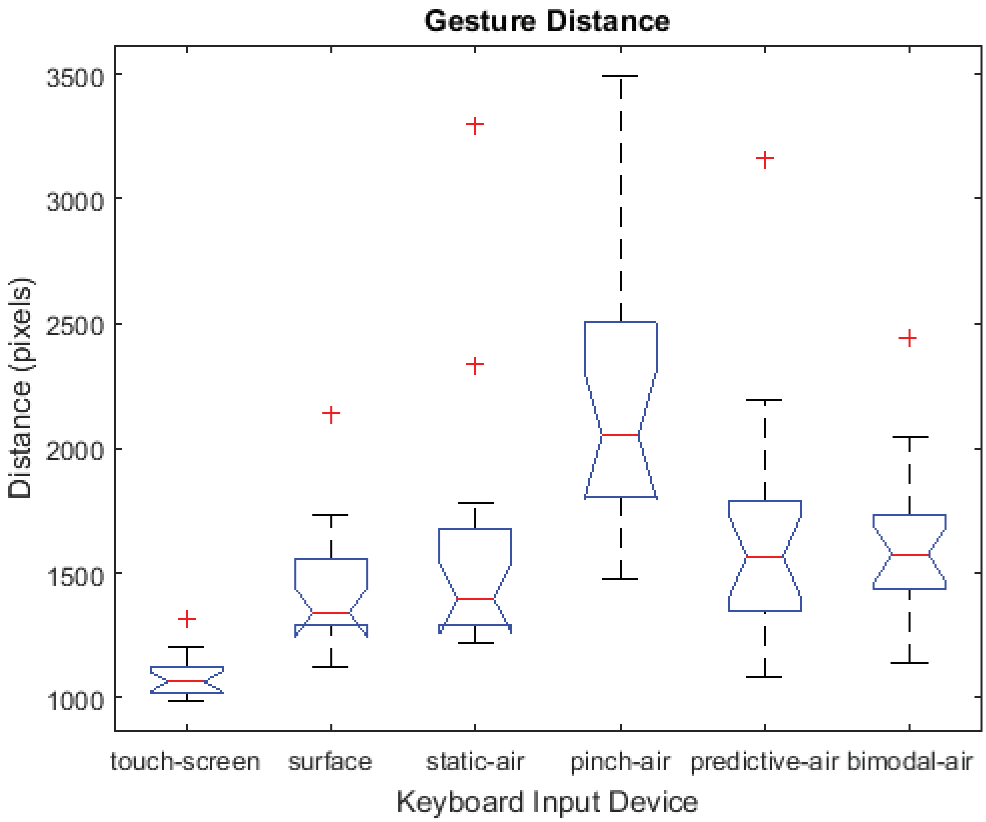


Figure B.13. Median Word-Gesture Distance for each keyboard showing the 25th and 75th percentiles.

Table B.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

B.4.1.1 Word-Gesture Distance Modified-Shortest.

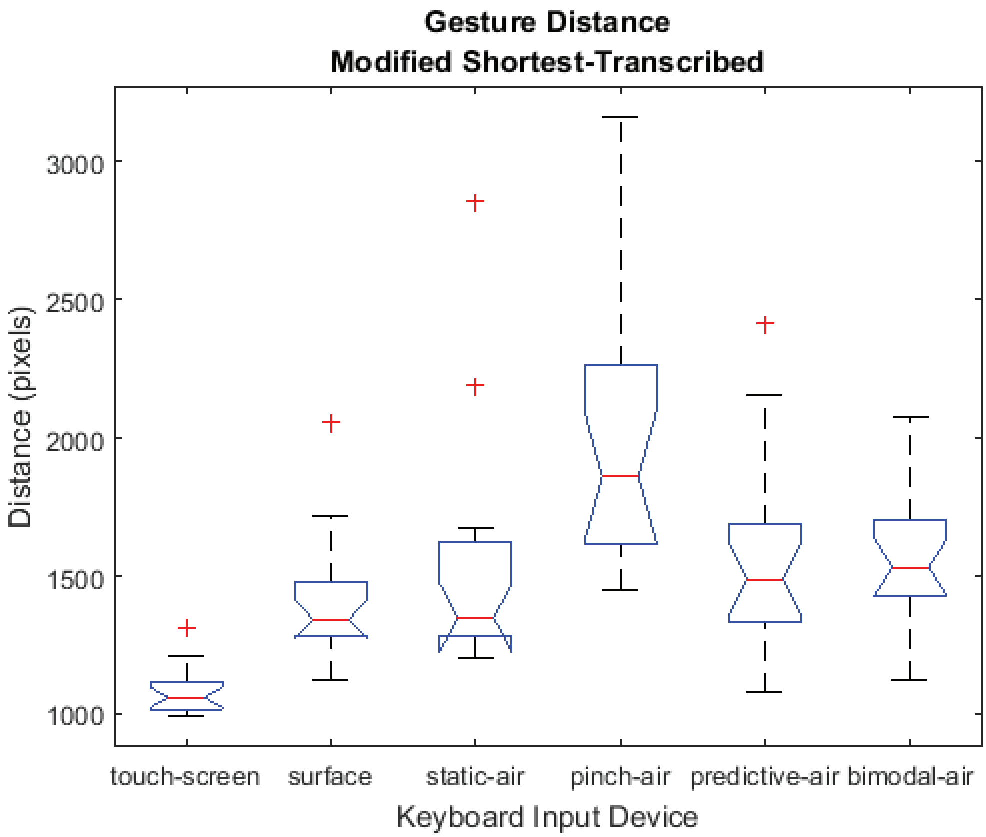


Figure B.14. Median Word-Gesture Distance using the shortest-transcribed modification for each keyboard showing the 25th and 75th percentiles.

Table B.14. Tukey's Honest Significant Difference with multiple comparison for Word-Gesture 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	-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

B.4.2 *Word-Gesture Velocity*

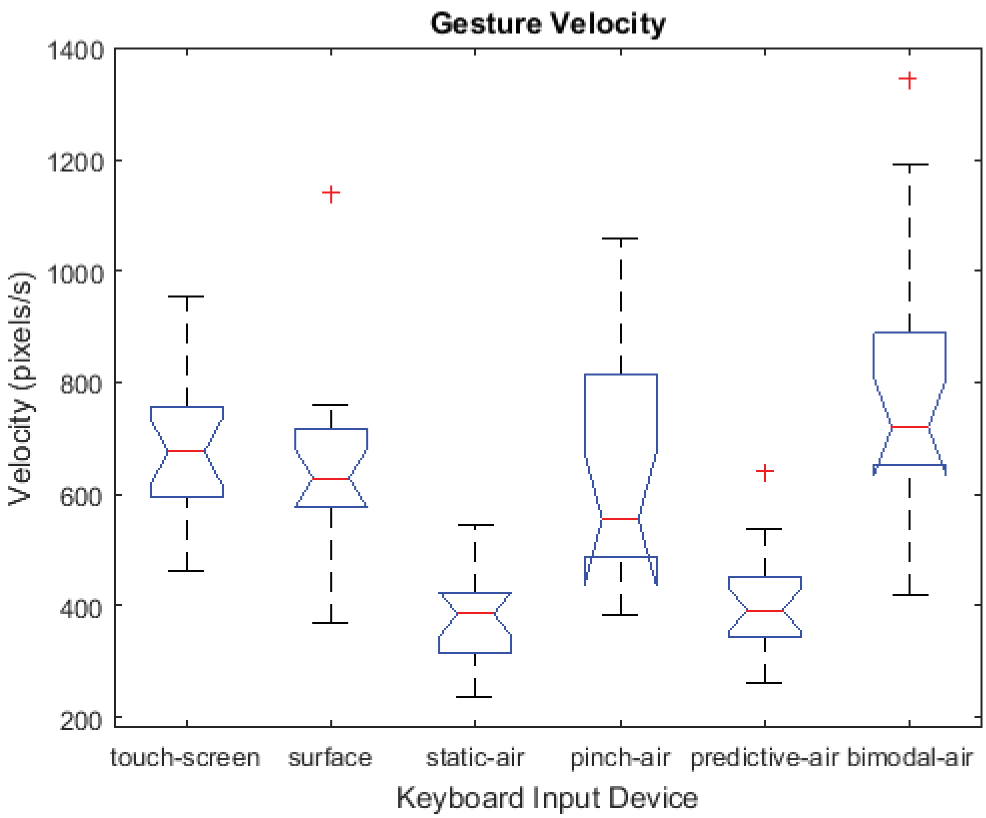


Figure B.15. Median Word-Gesture Velocity for each keyboard showing the 25th and 75th percentiles.

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

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience 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

B.4.3 Hand Velocity

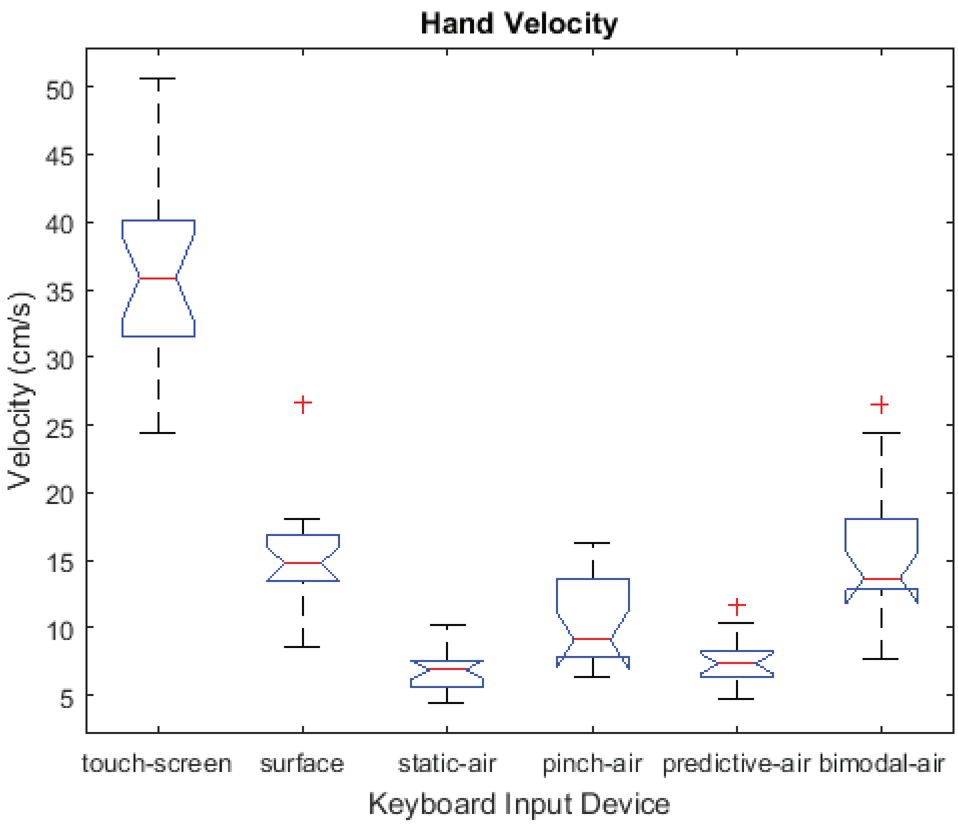


Figure B.16. Median Hand Velocity for each keyboard showing the 25th and 75th percentiles.

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

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience 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

B.5.1 Word-Gesture Duration

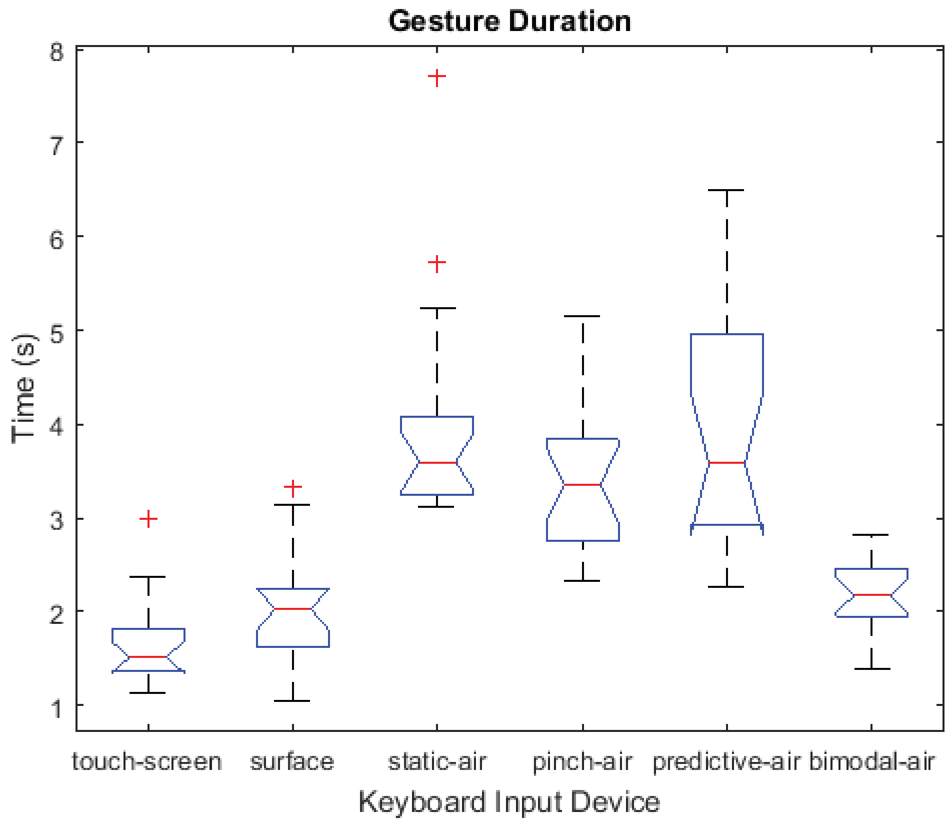


Figure B.17. Median Word-Gesture Duration for each keyboard showing the 25th and 75th percentiles.

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

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience 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

B.5.1.1 Word-Gesture Duration Modified-Shortest.

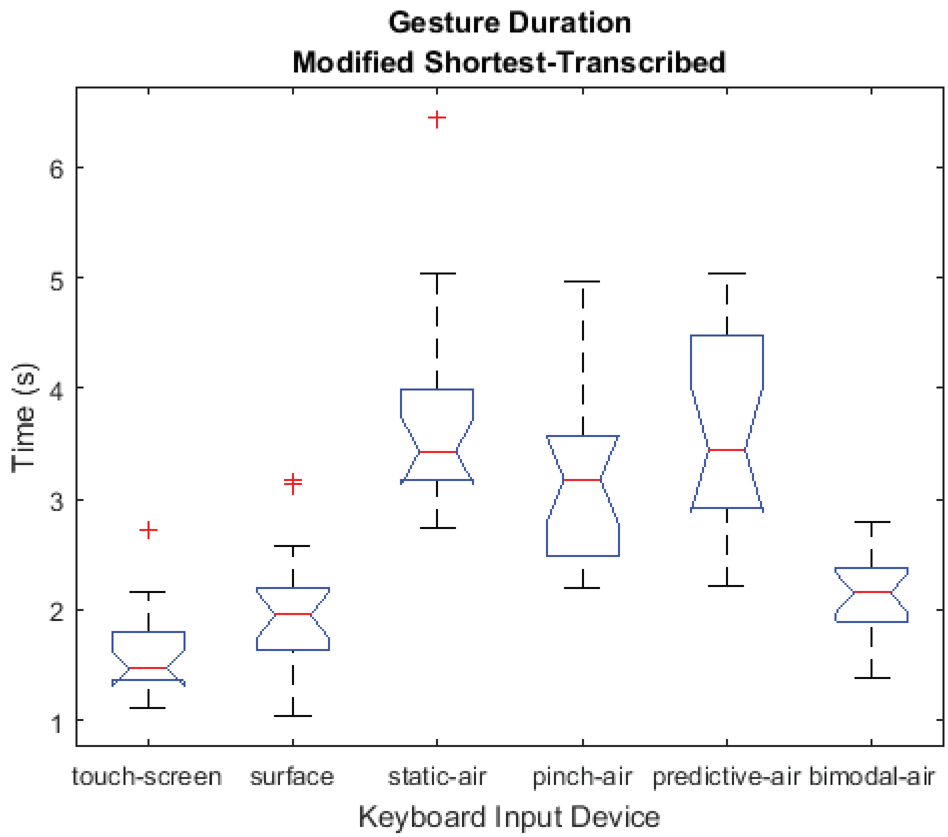


Figure B.18. Median Word-Gesture Duration using the shortest-transcribed modification for each keyboard showing the 25th and 75th percentiles.

Table B.18. Tukey's Honest Significant Difference with multiple comparison for Word-Gesture Duration 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	-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

B.5.2 Reaction Time to Errors

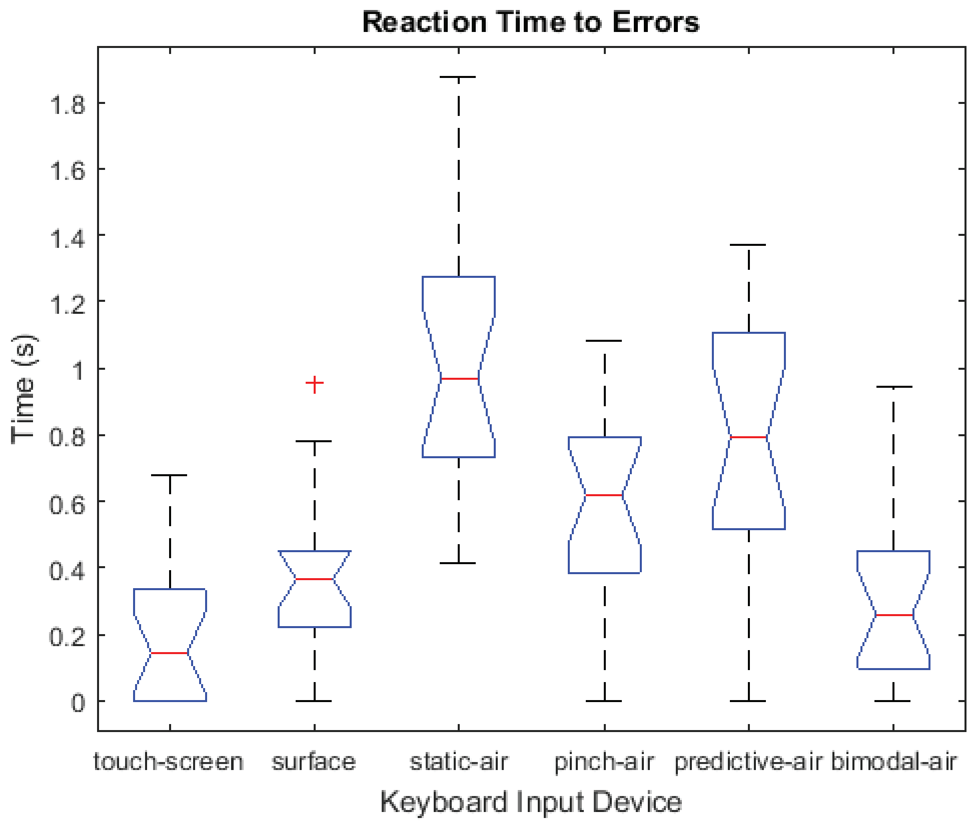


Figure B.19. Median Reaction Time to Errors for each keyboard showing the 25th and 75th percentiles.

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

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience 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

B.5.3 Reaction Time to Simulate Touch

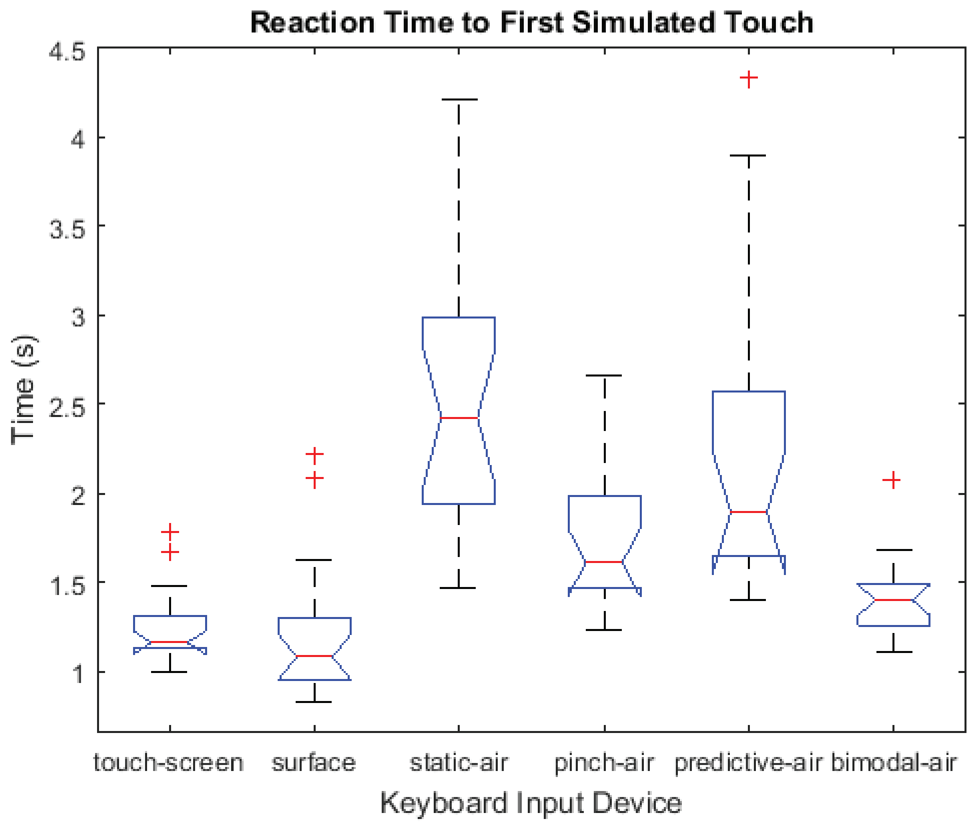


Figure B.20. Median Reaction Time to Simulate Touch for each keyboard showing the 25th and 75th percentiles.

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

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience 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

B.5.4 Reaction Time for First Correct Letter

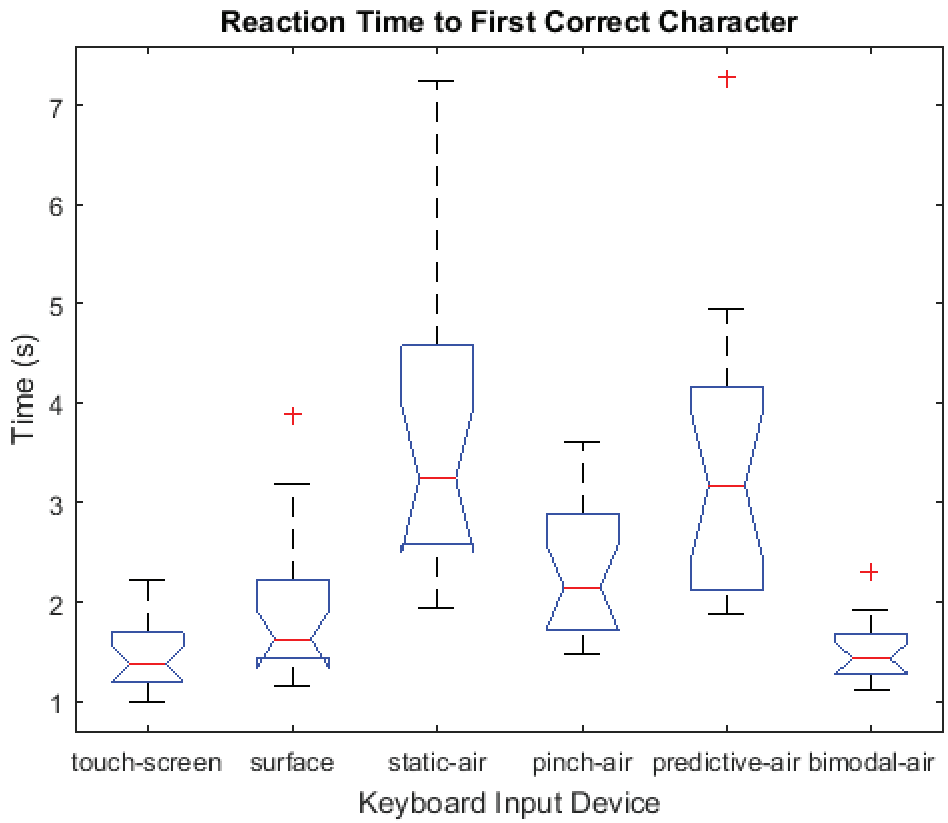


Figure B.21. Median Reaction Time for First Correct Letter for each keyboard showing the 25th and 75th percentiles.

Table B.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

B.6.1 Number of Touches Simulated

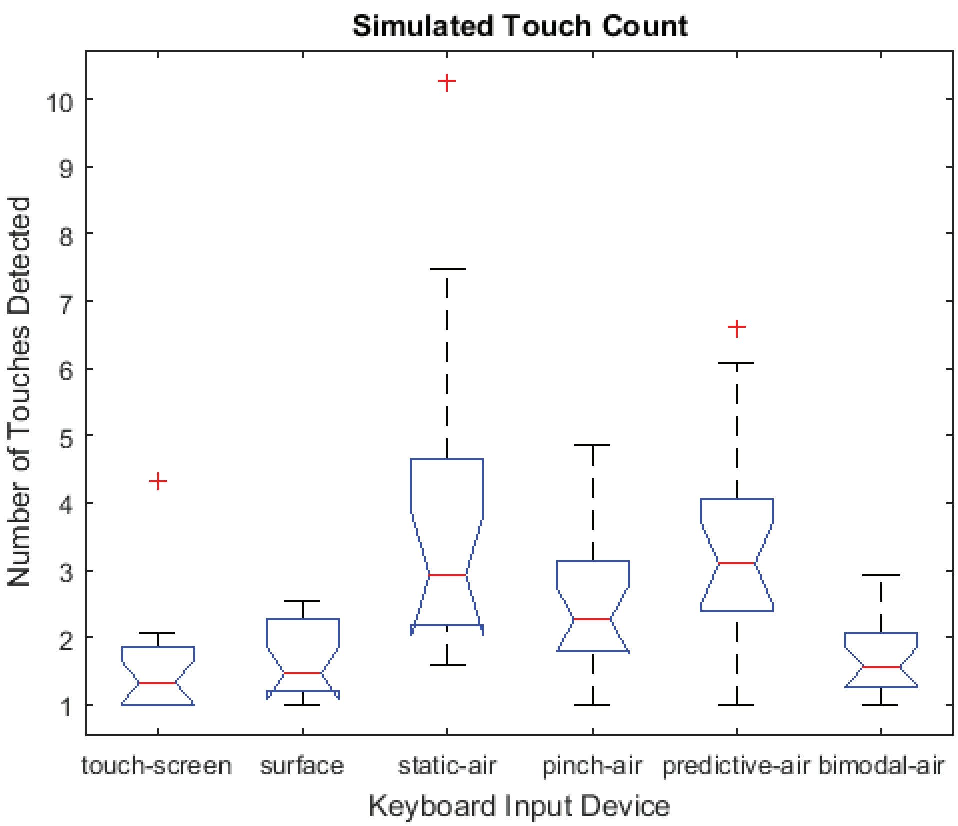


Figure B.22. Median Number of Touches Simulated for each keyboard showing the 25th and 75th percentiles.

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

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience 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

B.6.2 Number of Practice Words

.

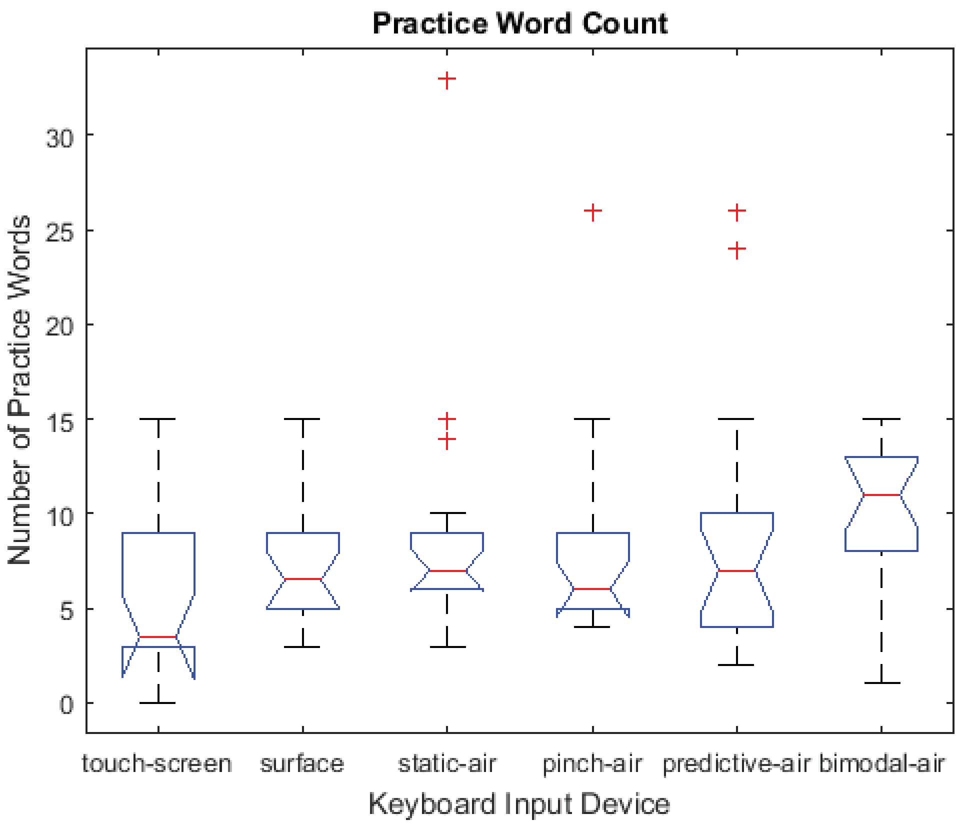


Figure B.23. Median Number of Practice Words for each keyboard showing the 25th and 75th percentiles.

B.7 Qualitative Measures

B.7.1 Level of Discomfort

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

Group 1	Group 2	Confidience Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidience 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

B.7.2 Level of Fatigue

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

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence 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

B.7.3 *Level of Difficulty*

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

Group 1	Group 2	Confidence Interval Lower End	Mean Difference $\mu_1 - \mu_2$	Confidence 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

B.7.4 Preference Ranking

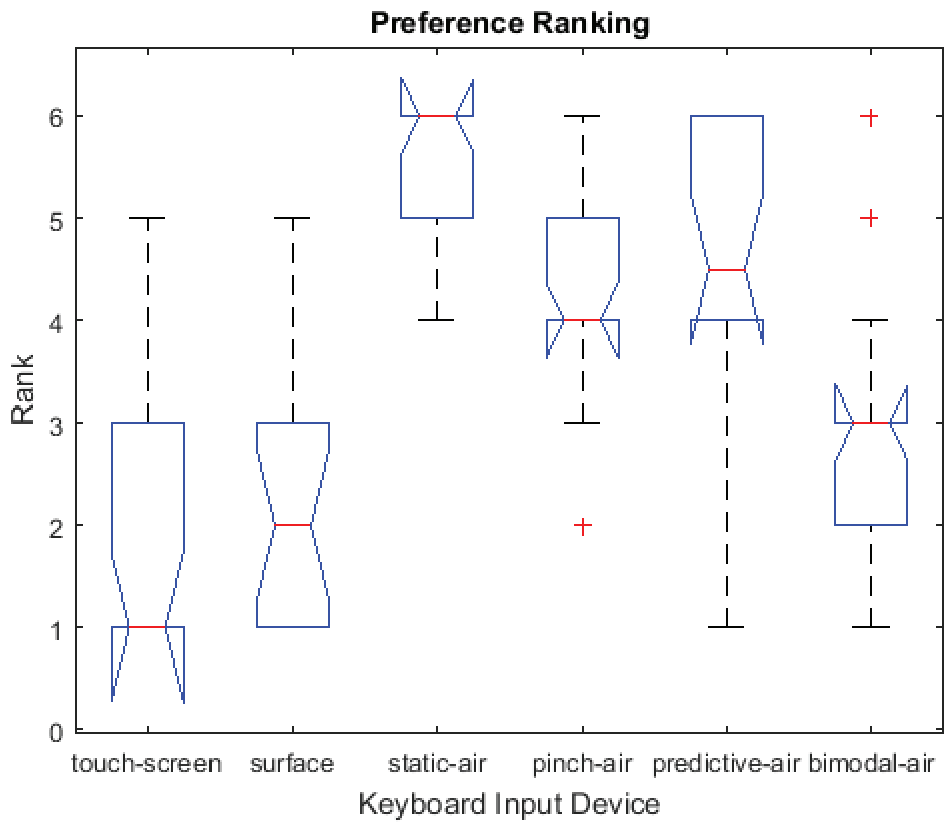


Figure B.24. Median Preference Ranking for each keyboard showing the 25th and 75th percentiles.

Table B.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.

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.

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.

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.

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.

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.

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.

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