

EMGRIE: Ergonomic Microgesture Recognition and Interaction Evaluation, A Case Study

by

David Way

Submitted to the Department of Electrical Engineering and Computer
Science

in partial fulfillment of the requirements for the degree of

Masters of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2014

© Massachusetts Institute of Technology 2014. All rights reserved.

Author
Department of Electrical Engineering and Computer Science
January 30, 2014

Certified by
Joseph Paradiso
Associate Professor
Thesis Supervisor

Accepted by
Prof. Albert R. Meyer
Master of Engineering Thesis Committee

EMGRIE: Ergonomic Microgesture Recognition and Interaction Evaluation, A Case Study

by

David Way

Submitted to the Department of Electrical Engineering and Computer Science
on January 30, 2014, in partial fulfillment of the
requirements for the degree of
Masters of Engineering in Computer Science and Engineering

Abstract

Given the recent success of hand pose gesture recognition via wrist-worn camera based sensors, specific hand pose interaction evaluation is needed. In order to evaluate such interactions, we built EMGRIE: a quick-prototype wrist-worn vision-based gesture recognition system that applies necessary feature extraction and training data collection techniques to facilitate user customization of specific hand pose gestures. We use EMGRIE to extract differences in microgesture task times, perceived effort, and perceived command associations across users, gestures, gesture performance iterations, and various applications. This thesis gives a summary of past wrist-worn gesture recognition systems and past gestural application design research, EMGRIE system implementation details and differences, free-hand microgesture choice rational, and sample case studies concerning Google Glass application design and associated usability experimentation. Given the results of our system and application experimentation, we find that specific hand pose gesture preferences may change drastically between users and propose hypotheses concerning hand pose usability.

Thesis Supervisor: Joseph Paradiso

Title: Associate Professor

Acknowledgments

I want to thank Prof. Joe Paradiso for agreeing to let me work on this project as his Research Assistant and helping me decide which directions to take the research.

I would like to thank Amna Carreiro, Linda Peterson, and the MIT Media Lab for helping me through the stress of securing funding for my tuition and research needs.

To the members of the Responsive Environments Group:

- Nan-Wei Gong, for providing reference and helping convince Prof. Paradiso to start the project.
- Nick Gillian, for providing the open source software that made my project possible and helping me review my conference research paper.
- Matt Aldrich, for teaching proper experimental design and statistical analysis in order to process data produced by such experiments.
- Gershon Dublon, for help in building Google Glass applications with Tidmarsh.
- Nan Zhao and Brian Mayton, for helping me produce my needed materials from the lab.

I would like to thank the numerous users who volunteered their free time to use my system such that I could record needed information.

Lastly, I would like to thank Chris Schmandt, Andrea Colao, and Ahmed Kirmani for allowing me to work on gestural interfaces as a UROP and providing the basis for this project.

Contents

1	Introduction	17
1.1	Motivations for Microgesture Study	17
1.2	Overview of Microgesture Study	18
1.2.1	Utilized Technologies	18
1.2.2	Target Gestures	19
1.2.3	Experimental Applications	19
2	Related Work	21
2.1	Wrist Worn Devices	21
2.1.1	Digits	21
2.1.2	The Image-Based Data Glove	22
2.1.3	RFID rings	22
2.1.4	AirTouch	22
2.1.5	Light Glove, Wrist Cam, and other gesture watches	23
2.2	Gesture Usability Studies	23
2.2.1	Stern et al.	23
2.2.2	Nielson et al.	24
2.2.3	Task Times	24
2.3	Sign Language Tracking	24
2.4	Skinput, Fingerpads	25
2.5	Interactive Machine Learning	25

3	Gesture Choice and Hand Pose Discretization	27
3.1	Sensible, Functional, Desirable	27
3.2	Hand Pose Categories	28
3.2.1	Finger lift and extend	28
3.2.2	Finger curls	31
3.2.3	Thumb to finger touches and pinches	32
3.2.4	External surface finger touches	34
3.2.5	Closed/Open gap gestures	35
3.3	Continuous Interaction vs Static Hand Pose	36
3.4	Combinations	37
3.5	Gesture Spotting	37
4	System Implementation	41
4.1	Computer Vision Feature Extraction	43
4.2	Finger Tracking	44
4.3	Gesture Recognition Toolkit	45
4.4	Microphone and Possible Sensing Modalities	45
4.5	Google Glass	46
4.6	Post Processing	47
5	User Integration and Learning Experiment	49
5.1	Fitts's Law Application	49
5.2	Initial Gesture Test and Experiment	50
5.3	Results	51
5.4	User test conclusions	52
6	Google Glass Text Entry Application	57
6.1	Introduction and Google Glass Usability Constraints	57
6.2	Text Entry Interface Design	58
6.2.1	First Iteration	58

6.2.2	Second Iteration	62
6.3	User test experiment	63
6.3.1	Randomized Display Experiment	65
6.3.2	Randomized Transition Experiment	70
6.3.3	Character Display Text Entry Experiment	77
6.3.4	Google Glass Display Text Entry Experiment	81
6.3.5	Expert User Learnability Experiment	84
6.4	Text Entry Application Design Discussion	86
7	Tidmarsh Focus Selection Application	87
7.1	Unity and Data Visualization Application	
Design Introduction		87
7.2	Interface Design I	88
7.3	Design I Experiment	89
7.3.1	Results	91
7.3.2	Result Discussion	93
7.4	Interface Design II	93
7.5	Experiment Design II	94
7.5.1	Results	95
7.5.2	Result Discussion	97
7.6	Focus Selection Design Discussion	97
8	Future Work	99
9	Conclusion and Discussion	101

List of Figures

3-1	The index lift. The MP joint bends the most, where the other joints in the finger remain loose. The pinky, ring, and middle lift are not shown for brevity.	29
3-2	The index extend. The MP joint bends the most, where the other joints in the finger remain loose. The pinky, ring, and middle extend are not shown for brevity.	30
3-3	The index curl.	32
3-4	The middle pinch.	33
3-5	The middle thumb middle phalanx touch.	33
3-6	The index external touch.	34
3-7	The index middle ring external touch.	35
3-8	The index middle together gesture	36
3-9	The index middle together ring lift gesture	37
4-1	The PMD camera inside the wrist mount.	42
4-2	The EMGRIE system attached to a user's wrist.	42
4-3	Image <i>A</i> shows the raw amplitude image returned by the TOF camera. <i>B</i> shows the thresholded image, and <i>C</i> shows the vertical edges returned by the Sobel derivative filter in the horizontal direction. . .	43
4-4	Finger centroids (green circles are midpoints of the finger edges, used to approximate the centroids), and finger tip detection (indicated by smaller green circles).	44

4-5	In addition to the camera attachment, we also used a microphone to aid in the detection of thumb to finger touches.	47
4-6	A user wearing Google Glass while using EMGRIE.	48
5-1	This box plot shows the task time spread for gestures over all users and all performance iterations.	53
5-2	Scatter-plot showing the gesture error rates for each user over each gesture (IL - index lift, ML - middle lift, RL - ring lift, IP - index pinch, MP - middle pinch, RP - ring pinch, PP - pinky pinch). Since task times were dependent on errors, the number of trials was varied across users. On average, each gesture was conducted about 12 times.	54
5-3	A table showing the user error rates (%) for each microgesture summed over each user. Clearly, the lift gesture group has lower error rates when compared with the pinch group. Since task times were dependent on errors, the number of trials was varied across users. On average, each gesture was conducted about 12 times.	55
6-1	Timing and error data for gesture sequences for an expert user. . . .	61
6-2	Displayed gesture graphic indicating "Open Palm" to the user.	66
6-3	Displayed gesture graphic indicating "Middle Lift" to the user.	67
6-4	Scatter plot showing task times times for randomly displayed gestures. Each gesture was performed 5 times.	68
6-5	A Box plot comparing gesture groups. The red line in each box indicates the median, where the edges of the box show the quartile boundaries. Outliers are also displayed.	69
6-6	User 2 random entry error matrix	70
6-7	User 7 random entry error matrix	71

6-8	The user viewed this screen while conducting experiments. The current gesture graphic was displayed to the left of the camera video and gesture prompting was displayed underneath. The current gesture in text form was displayed at the bottom. At the top, we showed the current sequences of vision computational preprocessing.	72
6-9	Random transition average group gesture rates and error rates. We see that the pinch gestures were more difficult to perform in transition between other gestures.	73
6-10	A box plot describing random transition time for each group in gesture per second.	74
6-11	A box plot describing random transition errors for each group in incorrect gesture per second.	74
6-12	User 3 average random transition times	75
6-13	User 3 random transition errors	75
6-14	User 5 random transition errors	76
6-15	Text entry input rates for each user and gesture group. The data is incomplete, since some users had other commitments.	78
6-16	Text entry input rates (gesture/second) for each group over all users .	79
6-17	Text entry error rates (error/second) for each group over all users . .	79
6-18	User 1 text entry error trends	80
6-19	Google Glass text entry input rates	82
6-20	Box plot describing the correct gesture rate for each gesture group over all users	82
6-21	Box plot describing the incorrect gesture rate for each gesture group over all users	83
6-22	Glass entry user 1 text entry error trends	83
6-23	Expert user correct gesture input rates over sessions (one session every few days for one month.	85

7-1	Tidmarsh application interface for discrete camera positions. The highlighted green number in the upper left indicates the current focus position. The views the area from the birds-eye perspective.	90
7-2	The box plot for each gesture and the corresponding association for 6 users in a user survey. A 3 indicates a level of strong association, and 2 indicates a medium level of association.	91
7-3	The box plot for each gesture and the corresponding effort level on the Borg scale for 6 users in a survey.	92
7-4	The box plot for each task and the corresponding discrete task times.	92
7-5	Tidmarsh application interface for cont camera positions	94
7-6	The box plot for each gesture and the corresponding association. A 3 indicates a level of strong association, and 2 indicates a medium level of association.	95
7-7	The box plot for each gesture and the corresponding effort level on the Borg scale.	96
7-8	The box plot for each task and the corresponding task times.	96

List of Tables

3.1	We describe each category here for convenience. Each category has continuous microgesture analogs and can be used in combination with each other to yield more complicated microgestures.	39
6.1	Text Entry Application I	59
6.2	Text Entry Application II	64
6.3	Experimental Groups	65
6.4	Variance and mean over all users (in seconds).	67
6.5	Text entry strings for each gesture group. Since character mappings were chosen in accordance to gesture, proper sentences were not used.	77
6.6	Variance and mean for correct / error gesture rates (gesture / second) over all users for each gesture group.	78
6.7	Glass entry variance and mean for correct / error gesture rates (gesture/second) over all users	81
7.1	The gesture to action mapping for the discrete Tidmarsh application	89
7.2	The gesture to action mapping for the continuous Tidmarsh application	94

Chapter 1

Introduction

In this manuscript, we present a novel methodology for microgesture recognition (small movements of the fingers and thumb), several applications designed for microgesture input, and usability experimentation examining task times, perceived effort, and fit-to-function. We give a summary describing past research in microgesture recognition and application design, a rationale for the microgestures we chose to explore, and a summary describing hypotheses that future research should explore.

1.1 Motivations for Microgesture Study

We believe that ubiquitous gestural user interfaces requiring large movements of the whole arm and shoulder, while powerful, do not reach full potential in gestural application usability. These interfaces require considerable volume and can quickly tire users (commonly referred to as "gorilla arm" [1]). In public settings, hand waving in the space in front of the user can be social awkward and consuming of personal space (eg. such gestural interfaces may be difficult to use while on a crowded train or when out for a walk). Hence, we believe that gestural user interfaces will eventually move away from the space directly front of the user's chest to the relatively physically small, immediate, free-hand space down by the user's side or in the user's lap. We will see that this gesture space requires little effort, provides potential for intuitive fit-to-function, and is efficient. In the past, most work in gestural applica-

tion design and sensor framework construction centers around gesture requiring large movements, whereas only few robust systems designed for small movement recognition gained notoriety. We believe that such systems did not possess feasible form factors, or present useful applications, to warrant proper attention. In this manuscript, we argue that despite limitations in form factor, usability experimentation results are needed in the space of microgesture and associated usability, since the physical hand wave required will soon shrink in size and footprint enough to make microgestural applications practical.

1.2 Overview of Microgesture Study

When studying microgestures, we were faced with limitations in our choices of possible gestures, sensing modalities, and application design.

1.2.1 Utilized Technologies

Many past microgesture recognition systems use computer vision and Time of Flight (TOF) sensors to extract information needed to determine hand pose, and we agree that such sensors are powerful allies in microgesture recognition. PMD Technologies produces an off-the-shelf small TOF sensor, for which we constructed a wristband to hold sensor.

In order to leverage the vision sensor in detecting finger to thumb taps, we utilized a microphone pressed against the wrist to detect bone vibrations created by microgestures.

Given the data provided by these sensors, we use open-source machine learning software to help build a pipeline in which data is collected, preprocessed, and produced in a format that a classifier can easily interpret.

1.2.2 Target Gestures

Since our sensor is positioned on the inside of the wrist, it cannot detect all possible hand positions. This restricts our study to microgestures that require a line of sight from the inside of the wrist - we will provide a summary of past research explorations of hand pose concerning this limitation and offer our own interpretation of possible microgesture categories.

1.2.3 Experimental Applications

After examining possible useful microgestures and constructing a robust method for detection, we designed and tested several applications.

Text Entry

Individuals will still be texting on the next generation of computer platforms, although with smarter context engines, texting will probably not be as limiting as it is now. Accordingly, the first application we examined closely was methods for text input. There are many different characters and hence many possible commands, causing many inherent trade-offs in application design. Nonetheless, we find text entry not requiring a keyboard or panel of buttons to be intriguing and potentially powerful.

Tidmarsh Focus Selection

More likely to be adopted than text entry modalities, we believe interaction with data visualization and focus selection (or "alternative choices") applications will leverage interfaces based on microgestures. In this case, we adopted the Responsive Environments Tidmarsh project's [2] Unity [3] build to produce a case study concerning on-site interaction and data navigation within a highly instrumented natural outdoor setting.

Chapter 2

Related Work

In this section, we will give brief summaries describing past research in a number of related multi-disciplinary areas.

2.1 Wrist Worn Devices

There are a number of wrist worn devices used for gesture recognition that are worth mentioning.

2.1.1 Digits

Microsoft Digits [16] is a wrist-worn system designed for extracting hand pose, and is functionally equivalent to the system described in this manuscript. However, there are extremely important differences. Digits was designed from the ground up with the objective of building a virtual model of the hand through estimating all joint angles found in the hand, and the in-article goes on to test the level of accuracy behind the system. Our system was built from the ground up only to extract gesture classification labels, and hence utilizes a simplified methodology to extract only gesture classifications in order to quickly examine usability aspects of an application and interface.

We hypothesize that our system uses less processing power to extract gesture,

but is inherently less powerful than extracting a full virtual model of the hand. In Chapter 4, we will further explore the benefits and trade-offs of our methodology for extracting gesture classifications.

2.1.2 The Image-Based Data Glove

Both the Image-Based Data Glove [26] and Sixth Sense [23] use finger tags and an RGB camera to extract hand pose. There are many other similar systems that use finger tags, and we feel that such research developed needed interest in the area. The Image-Based Data Glove’s camera location is equivalent to Digits (on the inside of the wrist) and Sixth Sense’s camera worn around the neck. In terms of usability, systems requiring fingertip tags may be cumbersome for the user to attach or wear when the user wants to use the system.

2.1.3 RFID rings

A non-vision methodology described by Bainbridge [6] uses RFID ring tags and a wrist reader module to detect the finger distances. As the study by Bainbridge clearly demonstrates, we find that non-vision methodologies such as [6] could be easily leveraged with vision based systems and result in a powerful gesture recognizer. Vision systems require line-of-sight and are not accurate when determining minute contact differences, so sensing modalities such as RFID may improve shortcomings in many areas.

2.1.4 AirTouch

The AirTouch system [18] utilizes synchronized tactile feedback as well as a vision-based gesture recognizer. The system is situated on the wrist with the camera pointed directly normal to the wrist where the other hand can input gestures. In order to confirm gestures, the hand with the wristband performs a gesture which vibration sensors on the wristband can detect. This solves the ‘gesture-spotting’ problem: the design prevents true negatives, as in the system does not detect gestures when the

user 'accidentally' performs them. We discuss this problem in detail in Chapter 3. In all, AirTouch is another good example of using alternate sensing modalities to improve small gesture recognition.

2.1.5 Light Glove, Wrist Cam, and other gesture watches

The Light Glove [14], Wrist Cam [32], PinchWatch [20], GestureWrist and GesturePad [27], and Gesture Watch [17] are all vision based systems. The systems mentioned are limited in the number of gestures they can detect. Specifically, the PinchWatch, GestureWrist, and Gesture Watch require large movements of one or both hands to satisfy the line-of-sight requirement inherent in vision systems. The Light Glove and Wrist Cam are designed only to detect one dimensional movements of the index, middle, ring, and pinky fingers. Many of these systems can be modified to improve upon these limitations, and we demonstrate such possibilities in Chapter 4 in our own implementation.

2.2 Gesture Usability Studies

The main goal to developing a microgesture recognition system was to explore possibilities of a universal microgesture language for wrist-worn sensing modalities. In order to explore, we experimented with usability aspects such as task times, effort perception, and fit-to-function perception. The following studies offer guidance on how to conduct such explorations.

2.2.1 Stern et al.

In the article by Stern [31], the authors present a case study that tests a number of gestures against a number of possible actions those gestures represent. The user is asked to rate the association between the gesture and action, as well as the effort required to perform the gesture. We take a similar approach to our own experimentation, but do not test all gesture to action combinations. The gestures examined in

[31] do not align with the microgestures we are interested in.

2.2.2 Nielson et al.

The article by Nielson [25] presents a methodology for designing gestural applications. We follow many similar steps, but decided to primarily focus on task times, effort, and fit-to-function when exploring usability and benchmarking. We explain our intuition in choosing microgestures for applications, and much of our thought process stems from the process described in [25].

2.2.3 Task Times

Reporting task times can be difficult. In past studies, it has been found that all task times are skewed in the positive direction and are not distributed normally. In the article by Sauro [28], it is clear that the geometric mean of the task time over users is the best way to compare averages. Through ad-hoc comparison tests in Chapter 5, we use statistical packages to compare such task time data sets through the geometric mean, as well as many other factors (median, variance, etc).

2.3 Sign Language Tracking

In the paper by Starner[30], the authors describe a method for detecting sign-language based hand gestures through hidden Markov models. While this is very useful, sign-language gestures were designed for the line-of-sight to extend away from the user. For someone to see communication clearly, gestures must be located our in front of the users. In our case, we are studying microgesture from the perspective of the wrist. We can use sign-language and methods to recognize gestures as examples, but we should apply designed gestural application solutions to the microgesture space.

2.4 Skinput, Fingerpads

Some interesting applicable multi-modal user interfaces are FingerPad [9] and SkinPut [13]. FingerPad uses magnetic sensing to detect minute movements between the thumb and index finger and can track the relative position of the thumb on millimeter scale. SkinPut detects vibrations the arm makes when struck with the opposite hand's index finger. Both of these system are wearable devices, but do not capture a gesture space as large as the hand pose. Nonetheless, they are very interesting user interface modalities that can be used in conjunction with a hand pose gesture recognition system.

2.5 Interactive Machine Learning

Interactive Machine Learning is a relatively new research space that studies how the human reacts to a machine learning user interface system and changes his/her input. For instance, if the user performs a gesture that is misclassified, he or she can alter their performance until they find that the system correctly classifies the gesture. Caramiaux touches on this in his paper Machine Learning of Musical Gestures [8] and Merrill examines such effects extensively in FlexiGesture [22]. Learnability is a usability aspect that is mainly mentioned in our Future Work section, but we do test for learnability on a short scale through task times experimentation and do a pilot study with an expert user over a series of weeks.

Chapter 3

Gesture Choice and Hand Pose Discretization

3.1 Sensable, Functional, Desirable

In the discussion by Benford [7], the authors explore various design constraints for building applications with multi-modal input capabilities. The authors state that input methods must be sensible by the technology involved, functional (intuitive and sensical) and appropriate to the given the application parameters, but also desirable to the user. Each of these constraints is applicable to the design of microgestural applications. The gestures involved must be sensible by the camera or other sensors on the wrist band (microphone, gyroscope, etc). In this case, most all microgestures must satisfy the line-of-sight requirement and/or be noticeable by the gyroscope or microphone. For example, lateral movements of the fingers when the fingers and palm are flat are not recognized, because the movements are not in the camera's line of sight. Given the specific application, gestures must be intuitive and have good fit-to-function - in Chapter 7, we will see how gestures should relate to a specified action. Finally, microgestures should be easy (desirable to the user). Certain hand positions are more difficult for others, thus not as desirable as an input modality. We must keep each of these constraints in mind when exploring which possible hand poses to use in applications.

3.2 Hand Pose Categories

In the study by Wolf [33], non-free hand gestures were explored by discretizing the set of all possible gestures in simple and intuitive categories. In order to follow a similar methodology to [33] we attempt to discretize all sets of possible free-hand pose into simple categories as well. While following the physical constraints set by extensive studies in the Grasping Hand [15] and line-of-sight constraints imposed by the wrist-worn sensor, we believe the space should divide into 5 categories: finger lifts and finger extends, finger curls, thumb to finger touch, external surface finger touch, and adjacent finger open/close gap categories. The categories stem from past studies and clear physical dimensions in which the thumb and fingers move. In all cases, we define a microgesture motion as originating from a base hand pose (the shape the hand takes when one lets their arm and hand hang loose by their side) to forming the defined hand pose, and then moving back to the base gesture. In the base pose, we found that that the fingers naturally curve towards the normal of the palm and the thumb comes over the palm towards the index finger to varying degrees over individuals. Since this hand pose requires no conscious movement or energy, the base gesture represents the easiest hand pose possible. For each category, we will discuss the movement required to move from the base position to perform the static or dynamic gesture. Our goal is to explore the set of design principles and constraints for classifying microgestures and therefore aid application development. Throughout the next sections, figures detail each microgesture category. Pictures of microgestures were taken from similar perspectives of a wrist-worn camera.

3.2.1 Finger lift and extend

To perform a lift or extend, the user must first very slightly move the thumb to the right side of the palm. This allows the sensor to have a clear view of fingers and gives each finger full range of motion. The distance the thumb needs to move varies over user-unique base positions. The thumb should remain extended and not curled. This thumb motion applies to the other categories except for thumb to finger touch. While



Figure 3-1: The index lift. The MP joint bends the most, where the other joints in the finger remain loose. The pinky, ring, and middle lift are not shown for brevity.

this thumb motion may be fatiguing, it is necessary in this setting. The WristCam study [32] and light glove [14] focus on this gesture.

When the fingers have full range of motion, the user flexes the defined finger towards the normal of the palm (lift) or away from the normal of the palm (extend) such that the finger only bends at the first joint (MP joint) (Figures 3-1 and 3-2). All other joints in the finger remain naturally loose. We found that it is common to move the index and middle fingers independently, allowing for discrete hand poses. Movement of the ring and pinky fingers is not independent, yet both can move independently of the index and middle fingers. While it is possible for gesture recognition to detect movement differences between the ring and pinky, large deviations between the ring and pinky vary in comfort over individual. The distance the finger bends depends on what is comfortable for the user (about 15 - 40 degrees in either direction from the base pose). For instance, a hand pose may involve the middle finger lifted while all other fingers are left in their base position or a hand pose where all fingers are completely extended such that the hand is flat.



Figure 3-2: The index extend. The MP joint bends the most, where the other joints in the finger remain loose. The pinky, ring, and middle extend are not shown for brevity.

When fingers lift, they naturally come closer together such that adjacent finger sides often touch. When this occurs, the gesture also falls under the closed gap category. The user must take precaution to counteract this natural occurrence and slightly increase separation between fingers. For instance, it is very difficult to lift all fingers and maintain clear separation between them. Due to this limitation, the difficulty level of finger lifts is larger than that of finger extensions. However, extensions often invoke a reflex: when the index finger extends the middle finger may lift to counteract the motion of the index. This reflex can be suppressed easily, but increases the cognitive load of the extension gesture.

The second limitation involves the complexity of finger lift/extension combinations. Each of the 3 finger sections (pinky/ring pair, middle, and index) can presume 3 discrete states (lifted, base, and extended). The lifted and extended states require conscious movement and therefore increase the difficulty of pose. While there are 27 various possible gesture poses, it is clear that only 6-8 of those poses has minimal

cognitive load: 1 or 2 fingers extended or lifted at one time.

We believe that discretizing the position of a single finger into 3 states is most beneficial to the user. Rather than consciously adjusting the level of movement required to match one of 4 or more possible states of finger position, the user simply decides which direction in which to move from the base position and applies a comfortable burst of movement. We aim for gestural interaction to be as consistent as possible to common user interfaces while maintaining simplicity: this discretization implies that there are two imaginary buttons on the topside and undersides of the finger. We leave side to side motion of the fingers to the adjacent finger closed/open gap category. Finger lifts and extends can also be used for non-discrete modes of interaction, such as sliders.

3.2.2 Finger curls

A finger curl involves all joints of the finger to bend (Figure 3-3). Unfortunately, this involves relatively more intense strength applied to all parts of the finger as opposed to the finger lift. The smaller joints increase the dexterity of the finger yet do not yield without noticeable strain. While the fingers can curl independently of each other, the user will notice increased discomfort after repetition and maintaining pose for long amounts of time.

There are many different positions a curled finger can assume. However, we decide to group all positions of a curled finger into one gesture to decrease performance difficulty level. The level of difficulty involved in performing a specific finger curl will only increase by discretizing a finger curl into 2 or more curl positions. We hypothesize that the curl gestures will not prove useful given hypothesized undesirability. However, the index curl, or common trigger motion, may prove useful for various applications.



Figure 3-3: The index curl.

3.2.3 Thumb to finger touches and pinches

Thumb to finger touch gestures encompass the entire space of tapping and sliding the thumb over a single finger or multiple fingers. In general, thumb to finger touches are easy to perform. The base gesture naturally draws the thumb over the middle of the palm and decreases the amount of energy required to move the thumb toward a finger. While these gestures are easy to perform, there are only select ways to discretize motions of the thumb touching a finger.

The thumb can tap the finger in many different areas. It is possible to discretize each finger into three areas in which the thumb can tap: the fingertip (third phalanx), the second phalanx, and the first phalanx. Difficulties of which phalanx to tap by the thumb varies over individual. The fingertip tap may prove to be one of the most natural and intuitive forms of discrete microinteraction. With this design, there are twelve total possible discrete gestures in which the thumb can touch the finger.

The thumb may want to slide between two different areas on one finger or two different fingers. This gesture allows for continuous interaction, effectively allowing



Figure 3-4: The middle pinch.



Figure 3-5: The middle thumb middle phalanx touch.



Figure 3-6: The index external touch.

the thumb to draw on or across fingers. While thumb drawing is a very powerful mode of interaction, there are only select on-the-go cases where continuous modes of interaction apply. The second limitation involves the difficulty to thumb draw over areas close to the palm, limiting the range of user experience.

3.2.4 External surface finger touches

When the hand is resting or hanging naturally, an external surface is often not far away. For instance, the side of the leg is on the palm side of a hanging hand. A table top is often on the palm side of a resting hand. The hand and wrist must move such that the fingers are close enough to the surface to touch. If on a tabletop, the wrist must elevate to allow the sensor a view of the fingers and tabletop. Apart from the wrist and hand movement limitations, the finger to external surface touch adds another useful set of discrete gestures.

Depending on the quality level of data processing and the smoothness of the surface, various continuous areas of the surface may be drawn on or tapped. To



Figure 3-7: The index middle ring external touch.

draw, the finger must curl to access other areas as opposed to moving the whole hand or wrist. The sensor position is relative the wrist, and cannot track its own relative position to an external smooth surface to allow static hand pose drawing. In future works, additional sensing technology may help solve this issue. Since the finger must often hold uncomfortable positions to access other areas of the external surface, we currently emphasize discrete finger taps as the most prominent mode of external surface interaction.

3.2.5 Closed/Open gap gestures

We have the ability to exhibit lateral motion in our fingers at the MP joint. We can extend this motion to create the last category: gestures that involve side to side finger touches. Difficulties of various lateral movements the fingers perform varies over individuals. There are 8 discrete finger side touch gestures (3 gaps, each with 2 possible states, closed or open). Depending on level of dexterity in the hand, users may experience various levels of performance difficulties.



Figure 3-8: The index middle together gesture

We found that the extreme version of the closed gap gesture (adjacent fingers crossed) is extremely difficult to produce as well as detect. Given the level of difficulty, it is clear that the fingers crossed gesture should not be used in an application.

3.3 Continuous Interaction vs Static Hand Pose

In the previous sections, we often discussed continuous interactions analogous to the discrete hand positions of each category. Given the parameters of the application, the continuous mode of interaction may be quite useful. In the case of sliders (volume sliders, page sliders), thumb to finger touch sliding is a clean way to mimic manipulating a scroller or slider. These modes of interaction are easily available in the hand and may prove very powerful in certain application designs. Chapter 7 gives a very good example!



Figure 3-9: The index middle together ring lift gesture

3.4 Combinations

To increase the power of an application through increasing the number of possible gestures to recognize, the mentioned categories may be used in conjunction with one another. For instance, the user can close the gap between the index and middle fingers and lift the ring finger. We define this gesture as index middle together ring lift, and use the formula *Specifier + Category + Specifier + Category + ...* to describe the entire gesture combination, where *Specifier* describes the fingers involved in the gesture and *Category* describes the action they take in correspondence with the categories mentioned.

3.5 Gesture Spotting

Gesture spotting is the ability of the system to pick out gestures from a continuous stream of data or movement. Many experts are skeptical of gestural user interfaces, as gestures are so natural and common that systems may confuse any movement as

input and register a true negative (a gesture that was not meant to be an input). Plus, for some gestural vocabulary, it may be much faster to transition directly between active gestures rather than forcing a return to the base gesture. We argue that this problem should be solved at the application level: a gesture that is not natural should be used to tell system to start recognizing all gestures used with the application. This way, users can specify their own preference for which gesture they find suitable as a "start" gesture and no true negatives will hinder usability. We suggest a combination gesture as a "start" gesture, such as index middle together ring lift, as an unusual yet easy to perform gesture fits the requirements. The system may also be activated by a deliberate, non-finger gesture; eg. tapping or scratching the sensor wristband. This may save significant power, as the camera will only be activated when gesture capture is engaged.







Microgesture Classes	Graphic	Description
Finger Lift		Bend the finger at the MP joint towards the center of the palm.
Finger Extend		Straighten the finger at the MP joint to point away from the palm.
Finger Curl		Bend all joints in the finger arbitrarily towards the palm.
Finger Pinch		Touch the thumb and finger together at specified locations.
External Surface Touch		Touch the specified finger against any surface.
Closed Gap Touch		Press the specified fingers together.

Table 3.1: We describe each category here for convenience. Each category has continuous microgesture analogs and can be used in combination with each other to yield more complicated microgestures.

Chapter 4

System Implementation

Since the main focus of this manuscript was to explore the usability of hand pose microgestural interaction, the software implementation involved is very straightforward. We have deemed the system EMGRIE (Ergonomic Microgesture Recognition and Interaction Evaluation). It primarily uses open-source software and off-the-shelf hardware. Applications were designed first by deciding which gestures were to be recognized, then implementing only the computer vision techniques needed to extract those features for a bag-of-words machine learning model, and finally then building the data sets in which to use for gesture classification. Once this pipeline was initially constructed, we found that it was very simple to upgrade, remove, and add new components in order to make changes to how gestures were ultimately classified. While our system does not recognize all possible hand positions currently, developers can easily modify the bag-of-words model inherent in the software to include the features needed for target gestures. We argue that in the low-power context of wearable electronics, additional computation for additional gesture classification should be optional or application specific. Overall, computation should be power-optimal and running on customized hardware. The bag-of-words model is suited for such cases. At our 50 frames per second, the camera we currently use consumes 200-400 mA, which is prohibitively high for frequent mobile use. Again, we present this system as an early incarnation. Power will decrease as this technology becomes tailored for mobile applications [29].

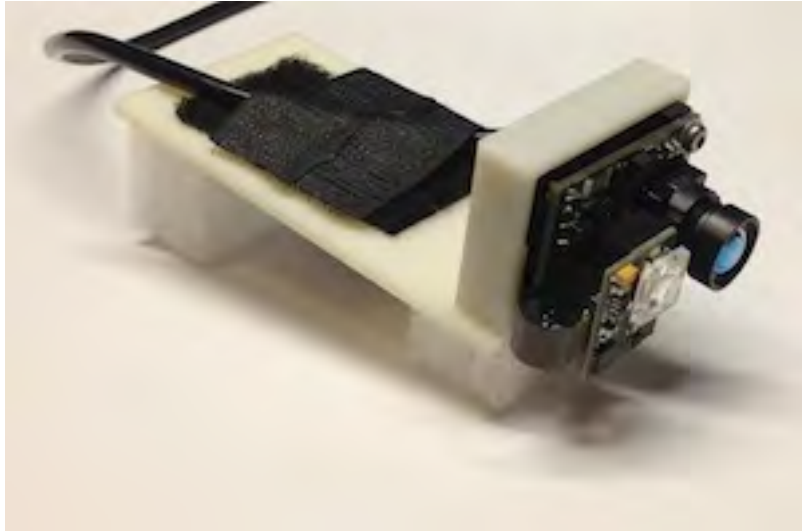


Figure 4-1: The PMD camera inside the wrist mount.



Figure 4-2: The EMGRIE system attached to a user's wrist.



Figure 4-3: Image *A* shows the raw amplitude image returned by the TOF camera. *B* shows the thresholded image, and *C* shows the vertical edges returned by the Sobel derivative filter in the horizontal direction.

4.1 Computer Vision Feature Extraction

The Time of Flight camera used was the PMD Technologies CamBoard Nano (figure 4-1). Note that, although it can be easily wrist-worn, this camera is large for this application. The camera design, however, was made to be generic and work at longer ranges - a range camera designed for this limited application should be able to grow significantly smaller. The camera allows the developer to set the 'integration' time, which specifies the amount of time the camera waits to receive infrared light from the foreground. We found that by setting the integration time to be less than the default allowed objects beyond the fingers to not be included in the gray scale image and improved the performance of the computer vision pipeline by about 50 percent. The image returned by the camera is a gray scale image in the form of an array of 32-bit float values from 0 to 1. Values closer to 1 represent areas in the image closer to the camera and values closer to 0 represent areas further away.

To manipulate the image during processing, we use OpenCV. To alleviate noise in the image, we first conduct an 'open' procedure (dilation then erosion) to remove small black pixel blobs. To create a binary image (the hand silhouette) and remove more noise, we experimented with a few threshold values (40 - 65 on the 0-255 scale). After extracting the silhouette, we use a Sobel kernel to find edges in the horizontal and vertical directions. Edges running the vertical direction indicate finger edges, and edges running in the horizontal direction indicate the thumb, fist, or palm edges. Based on the relative positions of the vertical edges, we can determine the centroids

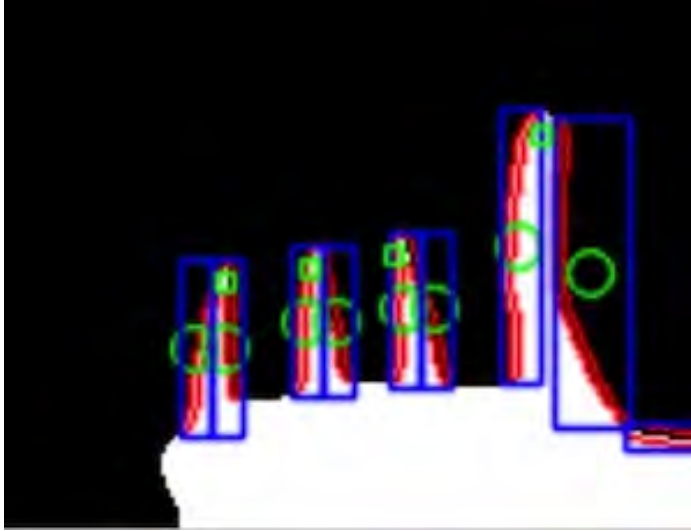


Figure 4-4: Finger centroids (green circles are midpoints of the finger edges, used to approximate the centroids), and finger tip detection (indicated by smaller green circles).

of each finger.

4.2 Finger Tracking

Through tracking the centroids of each finger, we can determine to which finger the vertical edge belongs. If the algorithm finds 8 edges, then we know for certain where the centroids of each finger reside. If the algorithm finds less than 8 finger edges, then we use the previous finger centroid locations and the label assigned to each finger to determine which finger the edge belongs. Based on these labels, we construct the feature vector from features found at each finger blob, such as the finger length, width, amplitude (mean distance from the camera). For detecting thumb to finger touches, we can similarly find the distances between the fingers and the thumb edge and include such features in the feature vector. If the set of features in the feature vectors cannot distinguish between two target gestures, then the developer must implement a methodology for extracting a feature from the image that allows classification.

4.3 Gesture Recognition Toolkit

Using the feature extraction pipeline section, we can collect data for each microgesture and train a sigmoid SVM classification model. Using Gesture Recognition Toolkit [12], it is very easy to collect data, save to disk, and train an SVM. Each user performs the corresponding static pose for about 10 seconds (about 500 data points, one for each frame at 50 frames per second) where EMGRIE learns the specific gesture the user wants to label as index lift, etc. This gives the user freedom to customize gestures and still appear to 'follow directions' given by the gesture and application. In general, we found that the size and shape of hands, especially when performing gestures, varies greatly over individuals. Hence, we only use individually trained SVM models as opposed to a universal model. We hypothesize that one SVM classifier per individual greatly increases the usability of gestural applications when gestures are very particular, as such is the case here.

4.4 Microphone and Possible Sensing Modalities

In order to aid in the detection of finger to thumb taps, we employed a microphone pressed to the user's skin. The TOF camera cannot detect the difference between the hand pose described by the thumb hovering just above the index finger and the thumb touching the index finger - the silhouettes produced are identical between the two gestures and the depth data difference produced is not significant between the two gestures. Through the microphone, we can detect the discrete moment in which the finger makes contact with the thumb. The contact makes a subtle vibration in the hand which is picked up by the microphone.

Through fast-Fourier transform, we noticed that the hand makes easily distinguishable lower frequencies (60 - 100 Hz) when the user taps fingers and the thumb together. By summing the amplitudes of the lower frequency ranges, we were able to use a threshold determined by another SVM to classify finger to thumb taps. We attempted various spots in which to put the microphone, but found that the best

position for listening to bone vibration was not the most comfortable position for the user. The most comfortable position we found was on the inside of the wrist. However when the microphone was placed on the inside the of the wrist, the microphone picked up interfering noise whenever the hand moved at all. Regardless of extraneous noise, the microphone was still able to recognize when the finger touched the thumb.

We found that sometimes the microphone failed. The microphone would shift locations such that it was not able to recognize vibrations, or the specific user produced a sound which did not trigger the classifier. This led to the user forcefully attempting to tap the finger and thumb together. To alleviate this issue, we slightly modified the vision system to also recognize the gesture without the need of the microphone. For instance, if the vision system classifies a pinch gesture lasting more than a certain large number of frames, the it can be assumed that the user is attempting to perform a pinch gesture. This way, each form of sensing is leveraged so there is minimal dependent form of input.

We would like to note further solutions to microphone failure. Further product engineering is required to ensure microphone location and pressure on the wrist. Continued research into data produced by hand vibration may lead to beneficial pattern recognition. Such methods may yield more proper results.

In [4], a similar method was used to detect interactions. We can extend such methodologies by using our camera with a string of microphones across the wrist for more robust modes of detection or adaptive gain/threshold setting. Through multiple microphones, signal transference distinction may be significant between various gestures.

4.5 Google Glass

Often during user tests, we asked users to wear a Google Glass and use an Android application side loaded onto the Google Glass. While watching users interact with the novelty of Google Glass was engaging, many users found the device slightly frustrating. Often, the user was farsighted and could not read the projection or had

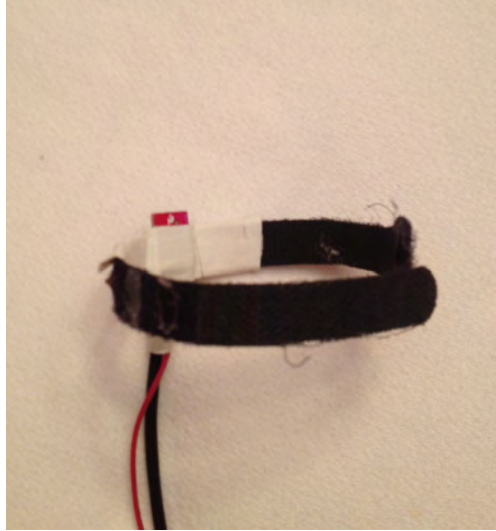


Figure 4-5: In addition to the camera attachment, we also used a microphone to aid in the detection of thumb to finger touches.

trouble adjusting Glass in order to see the display. While the user was using Google Glass, a background service `PowerManager` would decrease the brightness of the display after 30 seconds or shut the display off entirely. Since Google Glass is largely undocumented, it was difficult to find the way to remove this background service.

4.6 Post Processing

Upon classification, we can instill post processing techniques to reduce noise in the gesture recognition pipeline. Before recognizing a new gesture, we can count the number of frames that is classified as the same gesture. This reinforces the probability that the gesture being performed is the correct gesture. We can also employ a sliding window in which the gesture that is recognized the most over a certain number of frames (window width) is considered the output of the gesture recognition pipeline.



Figure 4-6: A user wearing Google Glass while using EMGRIE.

Chapter 5

User Integration and Learning

Experiment

The main goal of this research is to examine the potential for a universal gestural language and present possible free hand gestures to construct such language through pilot user studies. There has been longstanding desire to develop such a universal gestural language [10] and we hope to evaluate a series of gestures here. There are infinite possible hand poses and infinite possible gesture to action associations for various applications, but we can use previous and related work to decipher which hand poses should merit examination in a user study. In this chapter, we will explain why we chose to examine the gestures we did and why we chose to examine task times in our various experiments detailed in this and subsequent chapters.

5.1 Fitts's Law Application

Fitts's Law states that the time required to rapidly move to a target is a function of the distance to that target and the size of that target [11]. It is difficult to test Fitts's Law properly in gestural experiment contexts, as accurate measurements of the distance traveled by the various components of the hand to reach a certain hand pose and accurate measurements of the size of the space of correct hand poses are required. Even though we cannot properly apply Fitts's Law to a gestural task time

experiment, we can still use it as a way to hypothesize which aspects of microgesture affect task times. Since the distance various hand components have to travel to reach a hand pose and the size of space of correct hand poses is potentially greatly dependent on the gesture itself, we can hypothesize that the task time to complete a gesture is dependent on that gesture. We would like to note that since free-hand microgesture spaces are more physically close than full-body gesture spaces [15], this hypothesis should not be taken for granted, cannot be proven trivially, and requires a user study and standard usability statistical analysis. Hence, we believe that it is meritable to hold such a user study that examines the relation between gestures and task times.

5.2 Initial Gesture Test and Experiment

Since there are many hand poses (as described in Chapter 4), it is easy to fall into the experimental design trap of attempting to test 15+ gestures in a single user study. While some of our initial experimental designs entailed this volume of gestures, it is not feasible statistically and usability-wise. Users had difficulty learning more than 7 gestures at a time, and increasing the amount of within-user factors decreases the statistical power of the experiment. Hence, we decided to test two very different yet basic free-hand microgesture groups: the finger lifts and the finger pinches. Our motivation for choosing the finger lift group stems from the ability of the wrist-worn camera to recognize this group of gesture with high accuracy [32] and our motivation for choosing the finger pinch group stems from the hand’s natural ability to perform finger to thumb opposition [15]. Through examining these basic free hand gestures, we should be able to extract any statistically significant differences between them (if differences exist).

Specifically, we decided to use 3 finger lifts, the index lift (IL), middle lift (ML), and ring/pinky lift (RL). We study 4 different fingertip pinch gestures: index pinch (IP), middle pinch (MP), ring pinch (RP), and pinky pinch (PP). 11 participants (8 male 3 female ages 19-30) were asked to train the EMGRIE system to accommodate

their specific interpretation of each gesture. Once each gesture was trained and users felt mastery of gesture performance, the user completed 10 sequences of 7 random gesture appearances. Gestures were displayed one at a time via text description commands (i.e. lift index) and time to classification was recorded. Each sequence covered each gesture and was displayed in random order to mitigate learning and fatigue effects. The user rests both hands by their side or on their lap while orchestrating a series of such specified microgestures displayed on-screen 2 feet in front of them. Errors were flagged as either system or user errors and erroneous items were rescheduled in the sequence. To simulate an eyes-free situation, participants were asked to maintain their eyesight on the screen detailing the current microgesture to perform. There was a user-determined length break between each sequence and 3 seconds between each gesture performance. Each user test took on average 50 minutes to complete.

Our experiment design thus examines within-subject gesture factor and iteration factor and their relation to the continuous response factor task time. The experiment follows a randomized two-way repeated-measures design: each user performs each gesture 10 times ordered randomly. The results of this experiment upon the two-way repeated measure ANOVA should show any effect of short term training (the within-subject iteration factor) or gesture differences. Since task times are skewed in the positive direction, we removed outliers over a 95% confidence interval. In order to handle missing data, the model used was a Mixed Factor model. After finding that the interaction term between iteration and gesture was not significant, we converted to a repeated measures model that does not study the interaction term and only examine simple main effects and pairwise significance.

5.3 Results

Two-way repeated-measures ANOVA showed significant effect in the particular gesture factor ($p < .001$). We did not find a significant effect in the iteration factor. We continued our analysis through Tukeys pairwise multiple comparison post-hoc study. We found significant variations between pinky pinch and index pinch, and

between the ring lift and pinky pinch microgestures ($p < .02$). Figure 5-1 shows the task time spread and median for each gesture. We note that variance is large over each microgesture, but also see that task times can be as low as .5 seconds for each gesture. We use similar statistical tools to examine trends between microgesture and user error (summing over all users and iterations) assuming independent measures (task time and user error). Figure 5-3 shows the user error rate of each microgesture. Upon adding the microgesture factor to a repeated measures logistic regression model, model improvement showed microgesture having a significant effect on user error rate ($p < .0001$). Based on multiple comparisons (Tukey test) over microgestures, we found the index lift to have significant differences between the middle and ring pinches ($p < .02$).

Figure 5-2 shows the error rate of each gesture for each user. We would like to note that users have varying degrees of dexterity and hand pose experience - user 2 was an expert user that used EMGRIE for 5 hours each week for 15 weeks. User 10 was a pianist, and user 1 was a guitarist. Despite these differences, users still showed differences in error rates and task time and no trend was discernible.

5.4 User test conclusions

Before experimentation, it was not clear that gesture at this scale (simple free-hand pose) will show statistical differences in performance times. Fortunately, these results tell much about the relationship between gesture and task time. Clearly, microgesture has significant effect on performance time and user error, especially when gestures are physically different. One would expect that iteration of gesture performances should have an effect on task time, but it seems that 10 iterations over a 20 minute session is either a too little or too large sample size to show trend. Hence, we hypothesize that learning microgesture is similar to learning to keyboard - the required detailed control over ones hand requires large amounts of training to produce effects on task time.

In terms of the effect microgesture has on task time, we note that the pinky pinch

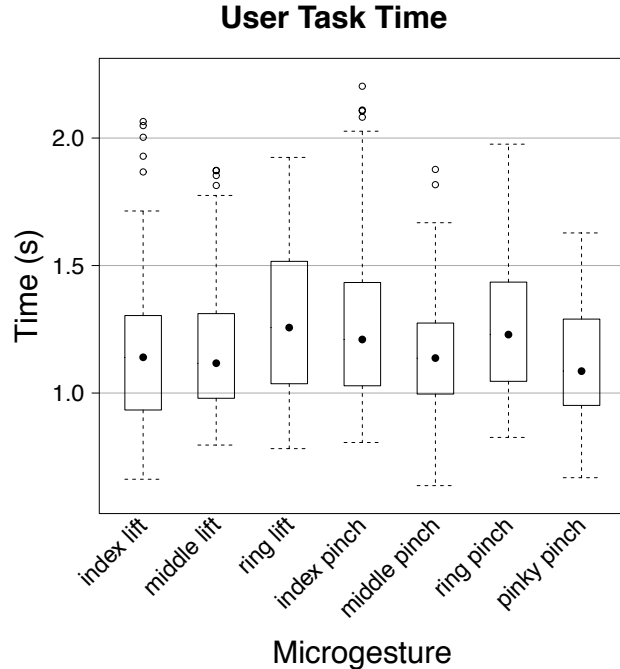


Figure 5-1: This box plot shows the task time spread for gestures over all users and all performance iterations.

and index pinch are physically different, and the ring lift and pinky pinch are physically different. It follows that we formulate a hypothesis as to why the pinky pinch microgesture was the fastest gesture: we believe that the pinky pinch holds a bias over the other microgestures during learning. The other microgestures may hold a continuous psychological representation or gradient of stimulus between themselves, where the pinky pinch (being the arguably most unnatural gesture concerning pinches [15]) may show a disconnect. Users may view the pinky pinch to be more unusual compared to the other microgestures and hence give the pinky pinch special attention due to the contrast bias [24] known in psychology. Further experimentation is needed to explore this resulting hypothesis. If true, then we would be able to infer an interesting generalization: while microgesture is more efficient than current ubiquitous gestural user interfaces, slightly unnatural microgestures are more efficient than natural more common microgestures due to natural microgestures being less distinct (special) and warranting more cognitive processing to perform.

Based on the scatter-plot in Figure 5-3, we find smaller user error rates concern-

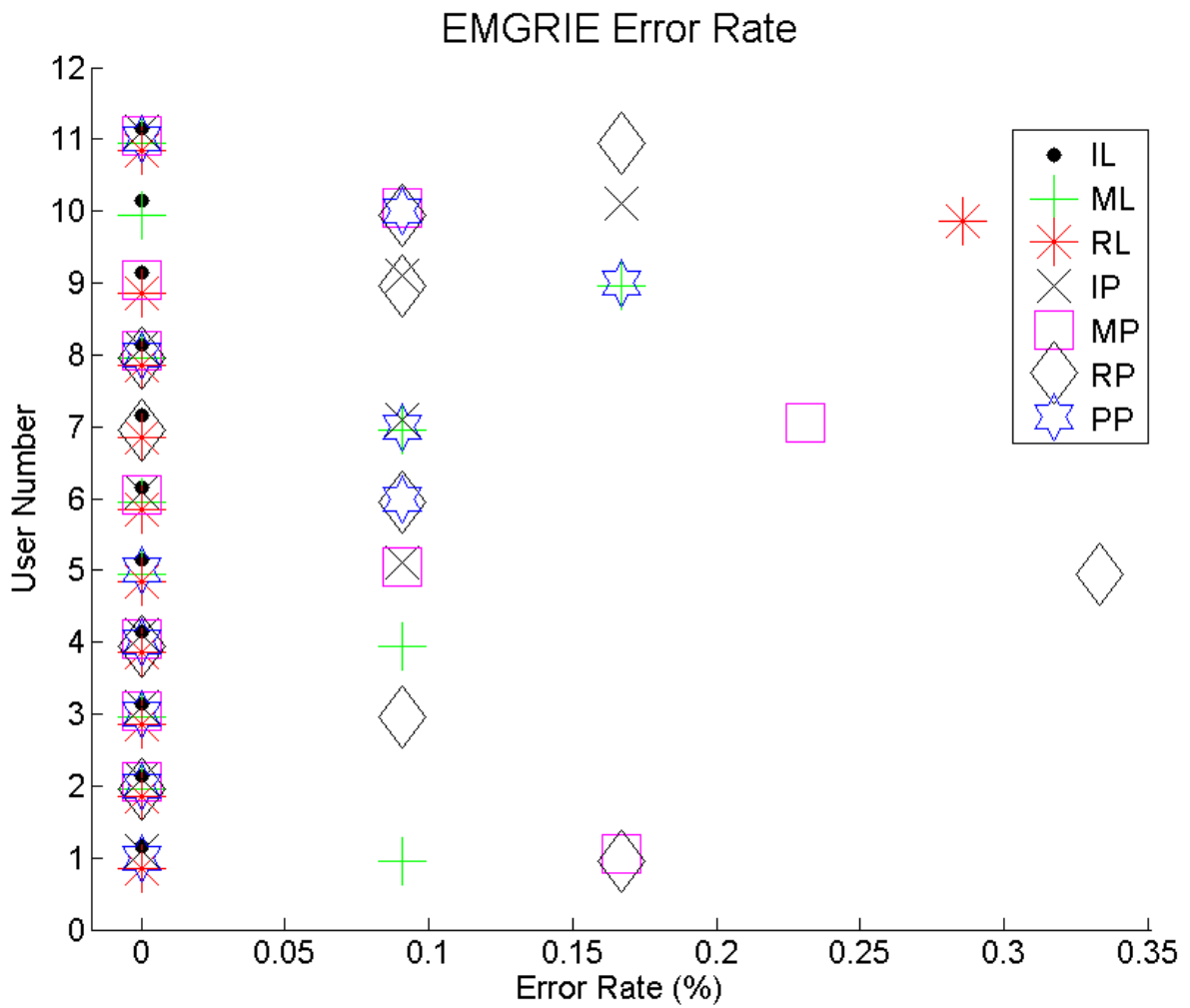


Figure 5-2: Scatter-plot showing the gesture error rates for each user over each gesture (IL - index lift, ML - middle lift, RL - ring lift, IP - index pinch, MP - middle pinch, RP - ring pinch, PP - pinky pinch). Since task times were dependent on errors, the number of trials was varied across users. On average, each gesture was conducted about 12 times.

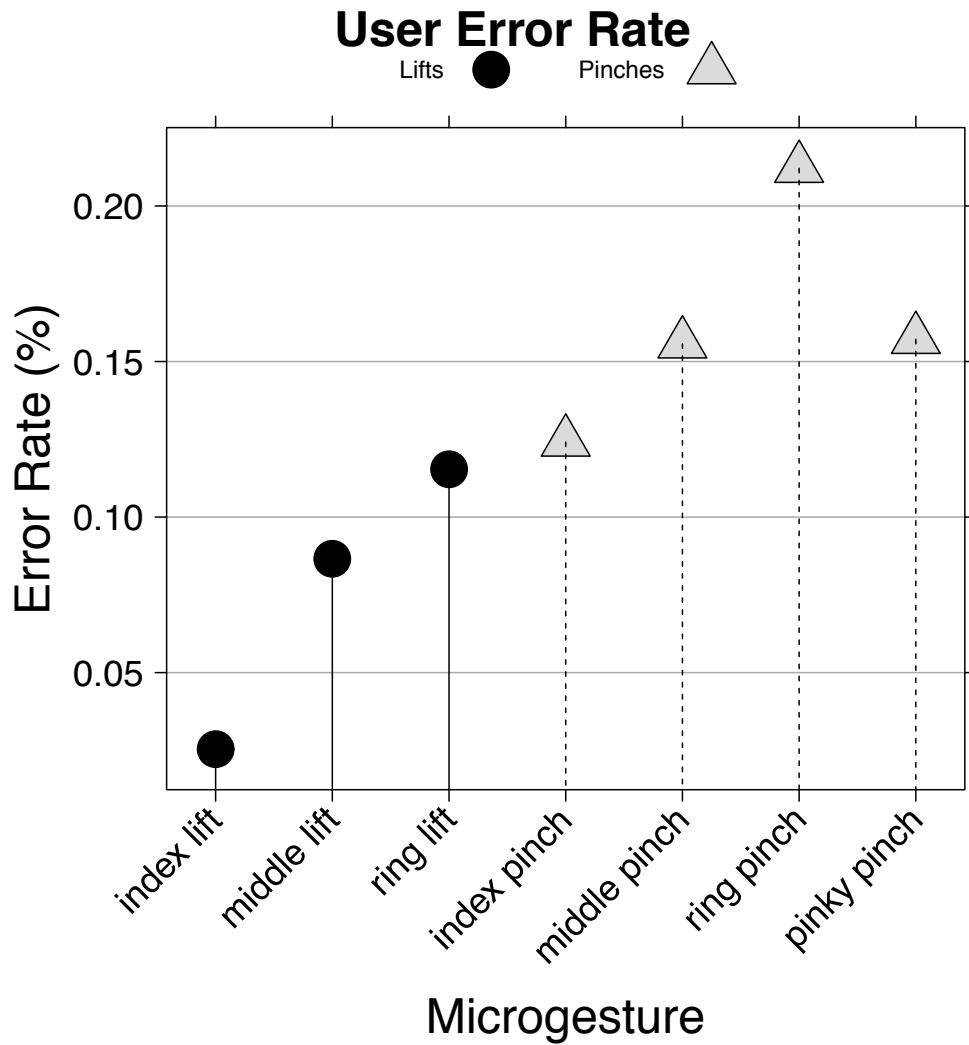


Figure 5-3: A table showing the user error rates (%) for each microgesture summed over each user. Clearly, the lift gesture group has lower error rates when compared with the pinch group. Since task times were dependent on errors, the number of trials was varied across users. On average, each gesture was conducted about 12 times.

ing the lift microgesture group compared to the pinch microgesture group. The lift microgesture group requires less muscle groups to perform and are generally more simple than the pinch group. We hypothesize that while the lift microgestures are not the most efficient microgesture group on average, they are simple for users to understand and are easier to perform correctly. We also find an interesting trend between fingers concerning user error: error rate increases near linearly from index lift through ring lift. This may be an effect of ring and middle fingers not having as much natural use as the fore finger. The pinky pinch user error rate continues to follow our theory concerning attention and contrast bias: it does not follow the trend of increasing error rate and we see a marked improvement from the ring pinch. Interestingly, we do not see a similar trend in task time that we see in user accuracy. More exploration is needed to examine the relationship between user error and task time in the microgestural application space.

Based on these findings, we believe that since the middle and ring pinches are not significant concerning user accuracy and efficiency, they may not be the best choice to be included in a universal microgestural language. The index finger and the pinky finger appear to be the most interesting potential bases of interaction. However, we would like to emphasize, from our own empirical observation, that users showed preferences for various microgestures. We suggest that microgestural application designers take into account the simplicity of the microgesture, the contrast bias, and the ability for the user to customize microgestures when building an application.

Chapter 6

Google Glass Text Entry Application

6.1 Introduction and

Google Glass Usability Constraints

Excitingly, we had the opportunity to utilize Google Glass for application tests. Upon first inspection of Google Glass, it became apparent that modes of interaction with the device are limited. To interact, users either utilize the track-pad on the side of the device or limited voice recognition capabilities. Hence, the lack of interaction modalities diminish the total capability of the device.

In this section, we attempt to show an example of increased capability of Google Glass through use of EMGRIE. Specifically, we decided to apply EMGRIE through gestural text entry and study user reaction to the text entry interface. For instance, based on the micro-gesture that EMGRIE recognizes, a certain character appears on the Google Glass display.

We do not foresee text entry to be the main use case for micro-gestural input on Google Glass. Such wearable displays will probably exploit high-level gesture to choose between context-emerged alternatives. As we will see from the proceeding experimental results, text entry requires acute attention to visual feedback (characters

on the display) and attention to minute gestural detail. The reader will see from interface design iterations that discrete micro-gestures become physically and sensibly similar when increasing the number of possible recognizable gestures, thus increasing the amount of concentration required to correctly input a gesture. Based on necessary close attention to input, we hypothesize that users will require large amounts of training to master gestural text input.

For the time being, we wished to use text entry as a mode to general exploration into task times and error rates to determine, over a large gesture space, which gestures are better suited for generic tasks and which gestures are more easily distinguishable to the sensor and the user. In the following sections we describe interface design choices, experimental design choices, and subsequent conclusions.

6.2 Text Entry Interface Design

6.2.1 First Iteration

In the first text entry interface design, we explored how a set of 127 characters could be referenced from only a set of 7 simple gestures. In the early stages of the EMGRIE system, seven gestures were available: index lift, middle lift, ring lift, index pinch, middle pinch, ring pinch, and pinky pinch (please reference chapter 3 for gesture descriptions). One possible method is to use gesture sequences and modifier gestures in tandem to access all possible characters. For instance, a sequence of 3 gestures from a possible set of 3 gestures can map to 27 characters ($3 * 3 * 3$). The other 4 gestures can be used as modifiers that switch input modes (uppercase characters, numbers, etc). Table 6.1 describes the gesture-to-character mapping.

We thought it would be valid to explore 3 micro-gestures that we hypothesized to yield fast transition times. In this case, we hypothesized that the index, middle, and ring lifts would yield fast transition times and decided to use those 3 gestures to construct sequences.

This application design covers all ASCII characters, yet requires the user to input

Gesture Sequence	Character Mode	Num Mode	Shift Mode
Index Lift, Index Lift, Index Lift (2,2,2)	space	space	space
Index Lift, Index Lift, Middle Lift (2,2,3)	's'	' 1'	'S'
Index Lift, Index Lift, Ring Lift (2,2,4)	'e'	','	'E'
(2,3,2)	't'	'2'	'T'
(2,3,3)	'o'	'3'	'O'
(2,3,4)	'i'	'4'	'I'
(2,4,2)	'r'	'5'	'R'
(2,4,3)	'a'	'6'	'A'
(2,4,4)	'f'	'7'	'F'
(3,2,2)	'c'	'8'	'C'
(3,2,3)	'v'	'9'	'V'
(3,2,4)	'w'	'0'	'W'
(3,3,2)	'j'	'!	'J'
(3,3,3)	'k'	'@'	'K'
(3,3,4)	'u'	'#'	'U'
(3,4,2)	'y'	'\$'	'Y'
(3,4,3)	'd'	'%'	'D'
(3,4,4)	'n'	^	'N'
(4,2,2)	'p'	'&'	'P'
(4,2,3)	'q'	*	'Q'
(4,2,4)	'b'	'('	'B'
(4,3,2)	'l')'	'L'
(4,3,3)	'x'	'['	'X'
(4,3,4)	'g'	']'	'G'
(4,4,2)	'h'	'{'	'H'
(4,4,3)	'm'	'}'	'M'
(4,4,4)	'z'	','	'Z'
Index Pinch	DEL	DEL	DEL
Middle Pinch	Num Mode On/Off	Num Mode On/Off	Num Mode On/Off
Ring Pinch	Shift Mode On/Off	Shift Mode On/Off	Character Mode On/Off
Pinky Pink	Punct. Mode On/Off	Punct. Mode On/Off	Punct. Mode On/Off

Table 6.1: Text Entry Application I

a minimum of 3 gestures per character. The experiment shown in chapter 5 explores described 7 gestures through task times and such times can extend to a sequence of 3 gestures. Subsequently, a small informal pilot experiment with an expert user (expert user being an individual well-practiced in performing gestures quickly, about 2-4 months) can show if results can actually predict gesture sequence. In this short experiment, the expert user conducts a series of gesture sequences. Gesture 2 stands for Index Lift, Gesture 3 for Middle Lift, and Gesture 4 for Ring Lift. Timing and error data are recorded. Figure 6-1 shows the average times for each gesture sequence. Each sequence was recorded 3 times and the whole study was conducted over 20 minutes.

One would expect sequences that require 2 gestures would take twice as long as sequences that require 1 gesture. However, this is not the case: the timing data does not scale linearly. We offer various hypotheses as to why there is a variable offset. Given that the experiment displays to the user 2 or 3 gestures at a time, the user can determine which gesture to perform next while performing the first gesture. However, the user still needs to spend time concentrating on the first gesture. This psychological computation yields an initial time baseline (the time the user spends thinking as to which first gesture to complete) which we would like to examine. Averaging the time required for sequences of various lengths, we see an average time increase in .24 seconds between sequences of 1 gesture and sequences of 2 gestures and an average time increase of .52 seconds between sequences of 2 gestures and sequences of 3 gestures. We hypothesize an increase in gesture sequence complication accounts for this increase in time difference between sequence lengths.

Given these time differences, we can estimate that a single movement then takes anywhere between .24 and .52 seconds, depending on the gesture and time interference from the user deciding which gesture to complete. Subtracting this value from the average time to complete one gesture, we are left with the average time the user took to decide which gesture to perform and what that gesture performance entailed. Unfortunately, we believe that .24 second and .52 second differences is not accurate enough to determine if the amount of time users take to perform gestures outweighs the amount of time users take to determine which gesture to perform next.

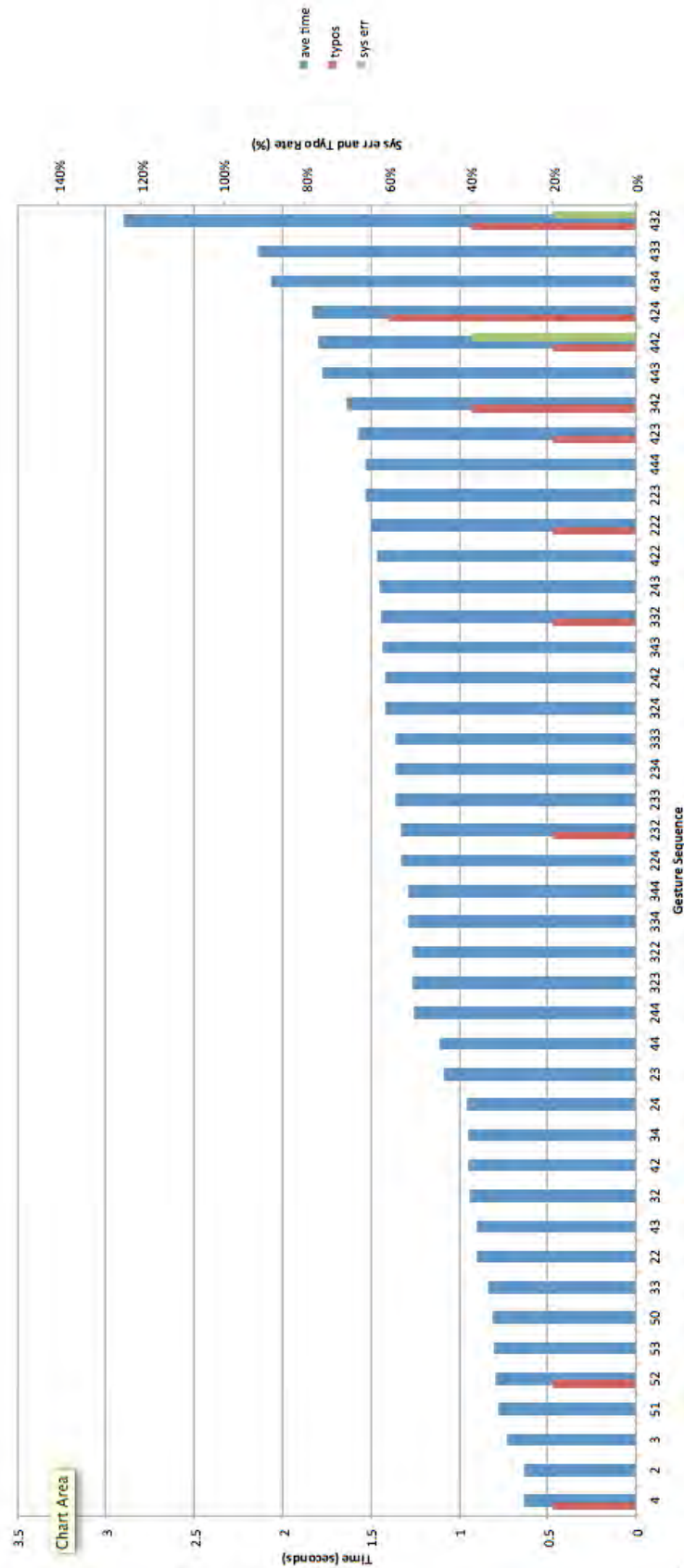


Figure 6-1: Timing and error data for gesture sequences for an expert user.

The variance of time over similar length gesture sequences is too large, which leads us to the conclusion that user time-to-action plays a larger part than the actual movement of the gesture. We would expect that after veteran use of EMGRIE, experiments would show time-to-decision to play less of a role in determining time-to-recognition.

6.2.2 Second Iteration

In the second design iteration, we attempted to remove the lengthy process of conducting 3 micro-gestures per character input by increasing the number of recognizable gestures. The first iteration explored 7 micro-gestures, but the second iteration explored 19 different micro-gestures including the original 7. The second iteration was designed under the constraint that one gesture should map to one character. In order to cover all characters, several modifier gestures are used. 13 gestures of the 19 are reserved for character input, while the other 6 gestures are reserved for modifier gestures and cursor manipulation. Table 6.2 shows in detail a sample gesture to character mapping allowing access to most all ASCII characters.

Given that there is a possibility of 13 gestures available, a very powerful mode of interaction involves extending EMGRIE to the second hand (the left hand). In this case, 'chording' between two hands is easily done. Text entry would not need a larger gesture set: 1 gesture in tandem with 1 gesture on the other hand yields 169 (13×13) different gesture chords. Given that this would require two cameras and significant amounts of software construction, we leave this aspect of interaction for future work.

Since we have more gestures available, we were able to make vital improvements to the glass interface design. In our first iteration, we did not handle screen space well. In this design, we allocate gestures to include 'next screen' and 'previous screen' actions. This allows the user to input large amounts text while not compromising text size - text must be fairly large in order to be seen on the glass display. We were also able to add actions allowing for cursor highlighting, copy, and paste. Our gesture-to-character mapping was done similarly to rows on the keyboard: r, f, and v map to gestures involving the index finger, and e, d, and c map to gestures involving

the middle finger. It may not be the case that this design is the best design. For this discussion, we are more concerned about the user reaction to gestures in a text entry use case rather than user preference in gesture-to-character mapping. The user preference in gesture-to-character mapping depends upon user reaction to gestures: gestures that are more difficult should map to less used characters (such as 'z'). There has been much research into the design of finger to button layout [19], and some design principles apply here.

6.3 User test experiment

We wanted the subsequent experiments to explore the user reaction to the various 19 gestures and their task times. Since it is difficult to design an experiment concerning 19 gestures that can we can easily analyze for usability, we decided to view simple main effects through comparing means and variance across gesture groups. In accordance, first-time users had difficulty learning 19 gestures in one sitting. Hence, gestures were split into 3 groups: the finger lift group, the finger to thumb pinch group, and the finger to side touch group. Learning to use EMGRIE is similar to learning a new instrument and is best done a few gestures at a time.

As the results will show, task times are skewed in the positive direction. Users also had a large range of previous experiences in dexterity: some considered themselves very dexterous and had many years playing the piano, where others did not consider themselves to be dexterous at all. While this range of previous experience allowed for a large variance in data, we hypothesize that previous experience has little effect compared with natural dexterity talent. Nonetheless, our experiments were designed to study the effect that gesture has on task times and not empirical data concerning past experiences.

Table 6.3 lists the experimental gesture groups. The middle index together lift and all together lift gestures were also included in each group and are not listed. These gestures map to space and delete actions such that each group can be examined separately with needed text entry capability. We decided to group gestures together

Gesture	Reg Mode	Switch Mode	Reg Shift Mode	Switch Shift Mode	Reg Num Mode	Switch Num Mode	Reg Shift Num Mode	Switch Shift Num Mode	Edit Mode
Middle Index Together Lift	space	space	space	space	space	space	space	space	space
Open Palm	'\n'	'\n'	'\n'	'\n'	'\n'	'\n'	'\n'	'\n'	'clear'
All Extend	Switch Mode	Reg Mode	Switch Shift Mode	Reg Shift Mode	Switch Num Mode	Reg Num Mode	Switch Shift Num Mode	Reg Shift Num Mode	Next Screen
Fist	Reg Num Mode	Switch Num Mode	Reg Shift Num Mode	Switch Shift Num Mode	Edit Mode	Edit Mode	Reg Shift Mode	Switch Shift Mode	Reg Mode
Thumbs Up	Reg Shift Mode	Switch Shift Mode	Reg Mode	Switch Mode	Reg Shift Num Mode	Switch Shift Num Mode	Reg Num Mode	Switch Num Mode	
Together All Lift	DEL	DEL	DEL	DEL	DEL	DEL	DEL	DEL	Prev Screen
Index Lift	r	u	R	U	1	!	[{	
Middle Lift	e	i	E	I	2	@]	}	
Ring Lift	w	o	W	O	3	#	\		
Pinky Lift	q	p	Q	P	4	\$;	:	
Index Pinch	f	h	F	H	5	%	'	"	Cursor Right
Middle Pinch	d	j	D	J	6	^	,	<	Cursor Left
Ring Pinch	s	k	S	K	7	&	.	>	
Pinky Pinch	a	l	A	L	8	*	/	?	
Index Touch	v	y	V	Y	9	(Start Selection
Middle Touch	c	n	C	N	0)			End Selection
Ring Touch	x	m	X	M	-	-			Copy
Pinky Touch	z	t	Z	T	=	+			Paste
All Touch	b	g	B	G	'	~			DEL

Table 6.2: Text Entry Application II

Group 1: Lift Group	Group 2: Pinch Group	Group 3: Side Touch Group
Index Lift	Index Pinch	Index Side Touch
Middle Lift	Middle Pinch	Middle Side Touch
Ring Lift	Ring Pinch	Ring Side Touch
Pinch Lift	Pinky Pinch	Pinky Side Touch
Extend All	Fist	All Side Touch
	Open Palm	Thumbs Up

Table 6.3: Experimental Groups

based on the natural categories described in Chapter 3. Gestures within the same category are similar enough such that they are easy to remember and to learn at once - yet different enough to be distinguishable to the sensor as well as the user. In order to cover all gestures, we distributed the rest of the gestures among the groups arbitrarily. Before each group, a gesture training phase was completed. For each gesture, the user formed that corresponding position with his/her hand and data was recorded for 3 - 8 seconds. The user rested their hand and wrist by their side in a comfortable position. Before recording, the user often practiced doing each gesture several times to test the consistency of the gesture. After each gesture recording, a classification model was loaded and the gesture was tested for reliability. About 500-800 data points were collected for each gesture. After all gestures could be reliably detected, the testing phase started. The following sections describe the testing conditions of each experiment.

6.3.1 Randomized Display Experiment

In this experiment, we hope to show which gestures are faster on average compared to other gestures. The test consisted of 5 repeated sections of conducting a random gesture in each group one at a time. To begin a section, the system counted down while the user maintained the base gesture. If the user left the base gesture before the count down finished, the count down started over. After the count down finished, the system displayed a random gesture graphic and recorded the next valid gesture recognized and the time to recognize that gesture. After each gesture performance,

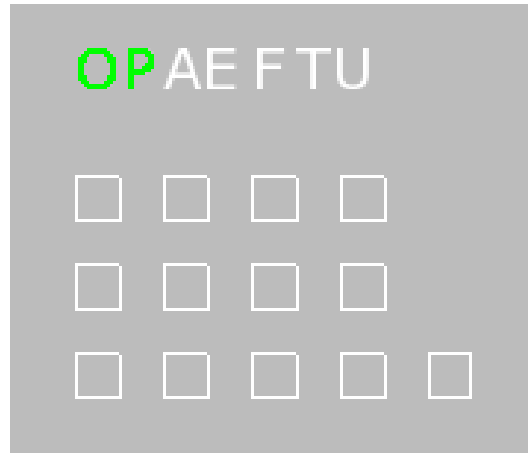


Figure 6-2: Displayed gesture graphic indicating "Open Palm" to the user.

another countdown started automatically until all gestures in the group were completed. Gestures that were completed incorrectly were appended to the end of the test. The randomized gesture prompt was displayed to the user through graphical form (Figures 6-2, 6-3). The first row in the graphic details the open palm (OP), extend all (AE), fist (F), and thumbs up (TU) gestures by displaying the corresponding letters green (Figure 6-2). The second row signifies the pinky lift, ring lift, middle lift, and index lift gestures in order from left to right by filling in the corresponding box (Figure 6-3). The third row details the pinches and the fourth row details the side touches in similar likeness. To signify the all touch side gesture, the bottom row rightmost box lights up. To examine errors, we record each user's goal gesture / recognized gesture pair count. For each user, different gestures are physically and mentally closer to different gestures - error pairs should show this trend. Though the goal / recognized matrix, we can predict each user's gesture preferences.

Results

Figure 6-4 depict the correct microgesture input time from the 'base gesture' to the prompted gesture. On the Y axis, the finger lift gestures, finger pinch gestures, and index touch gestures are displayed bottom to top respectively. Time is on the X axis. Data is split into groups per user in order to visualize the response each user has for

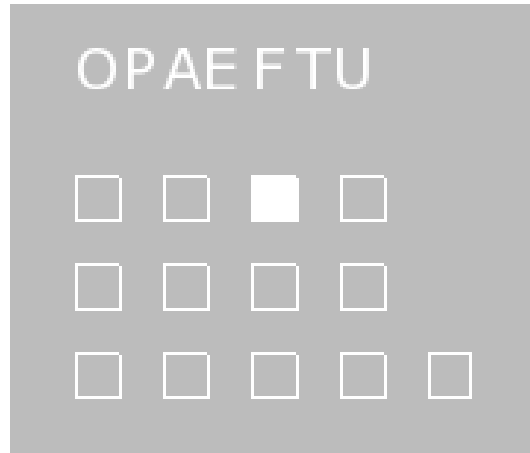


Figure 6-3: Displayed gesture graphic indicating "Middle Lift" to the user.

Time (s)	Lifts	Pinches	Side Touches
Mean	1.08	1.18	1.8
Variance	.095	.037	.74

Table 6.4: Variance and mean over all users (in seconds).

each gesture. On average, the pinky lift was the fastest gesture (.91 s), as were the lifts the fastest group compared to the pinches and side touches. Figure 6.4 shows the variance and mean of gesture groups over all users, and 6-5 plots the median and spread in a box plot form. Figure 6-6 and 6-7 displays user 2 and 7 's goal gesture / incorrect recognized gesture matrix respectively (these users showed similar trends for errors, as described below).

Experimental Conclusions

Based on the low variance and mean time of the lift and pinch gestures, it appears the that the lifts and pinches are faster gestures over all users. We can predict that these gestures are more familiar to users than the other gestures. It seems that user 7 provided outlier data for the side touch gestures - user 7 had difficulty interpreting the gesture graphic (figure 6-2) for these gestures. Several users expressed that the side touch gesture indicators were 'flipped' compared to the hand orientation (the index touch indicator is on the right side of the graphic where the index finger is on

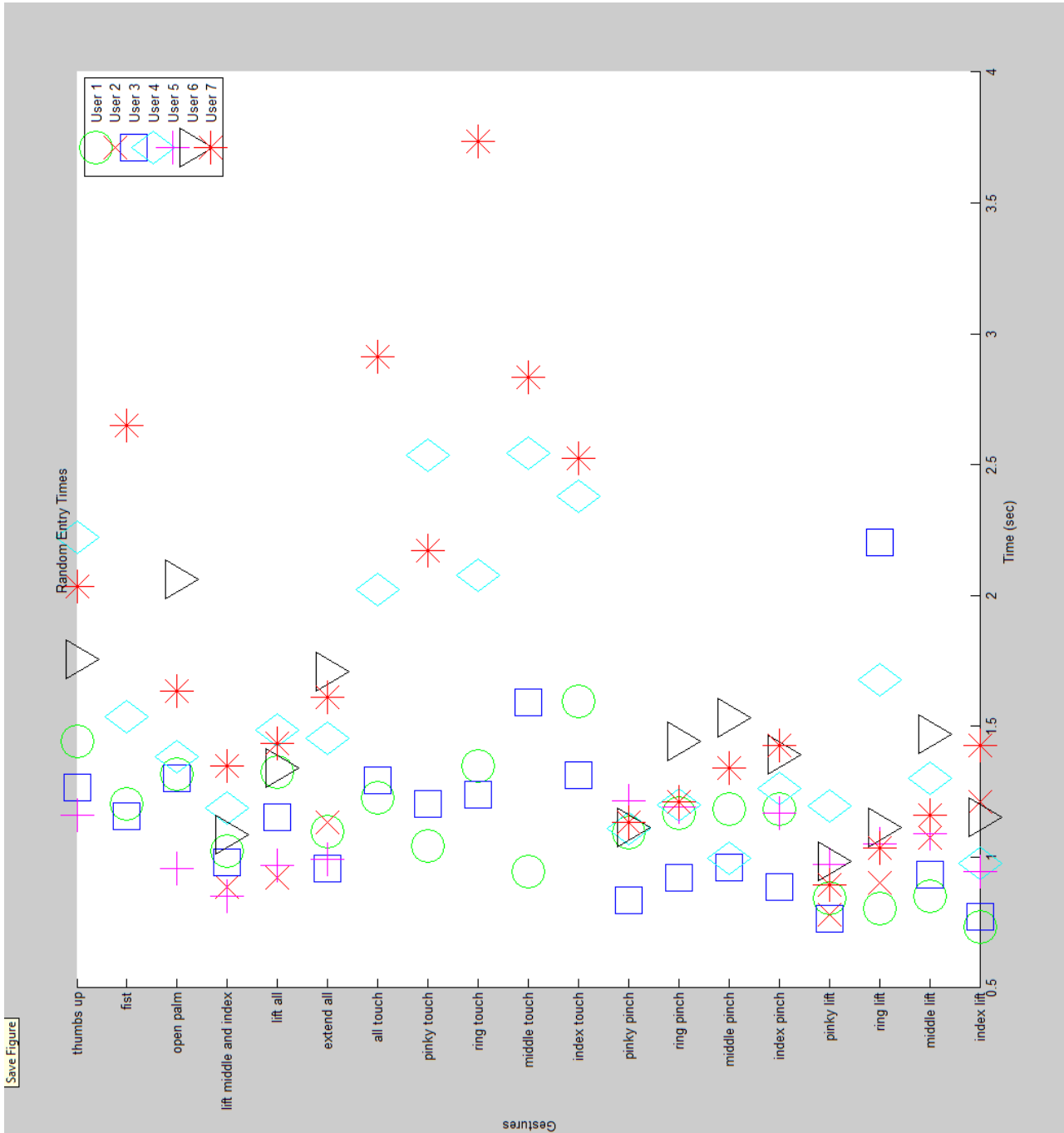


Figure 6-4: Scatter plot showing task times times for randomly displayed gestures. Each gesture was performed 5 times.

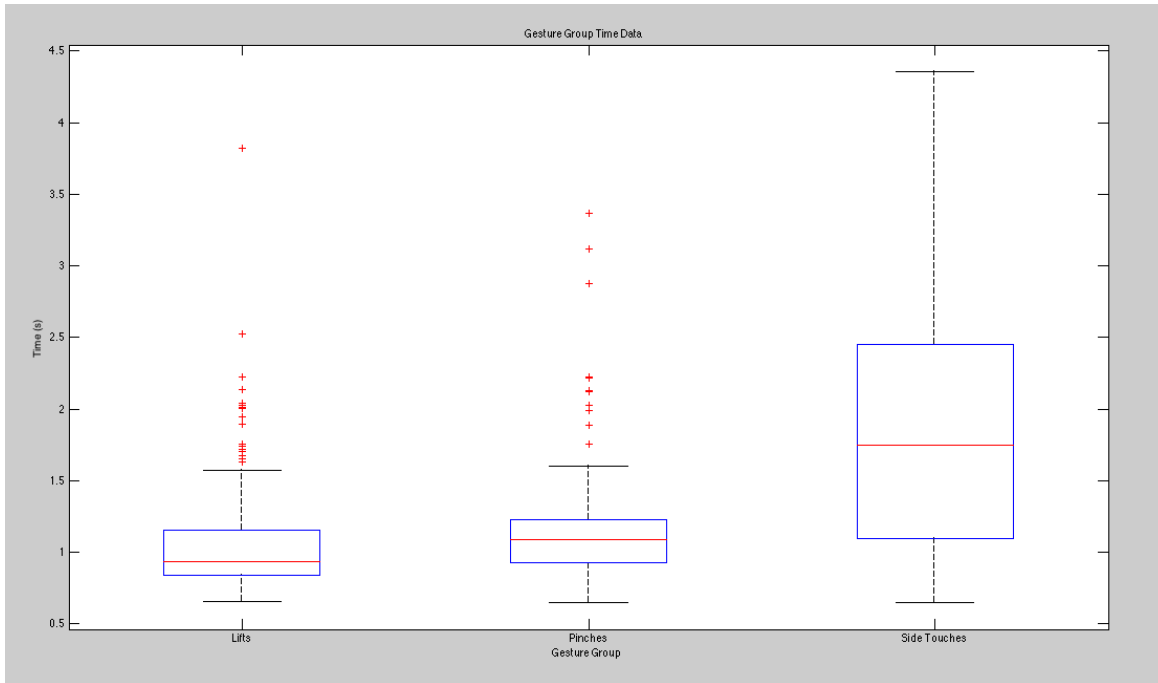


Figure 6-5: A Box plot comparing gesture groups. The red line in each box indicates the median, where the edges of the box show the quartile boundaries. Outliers are also displayed.

the left side of the graphic). This may be an explanation as to why the finger side touches have larger variance.

Most users made few errors over the 5 tests (each gesture was attempted until 5 successful performances per group). User 7 made the largest number of errors over all users. There was common problem (shown in user 7 and user 2) with identifying between the middle index lift together and middle lift gestures. When the middle finger lifts, it often occludes the index from the perspective of the camera. Given that the system does not take into account the width of each finger blob, the system cannot determine if the index finger is also lifted and hence cannot often determine the difference between gestures. To fix this problem, the user had to lift the middle finger such that it does not occlude the index finger. This problem captures the shortcomings of a bag-of-words machine learning model but also captures an interesting interactive/usability machine learning concept. The user can change the input to accommodate the system at the expense of usability (it is not desirable for the

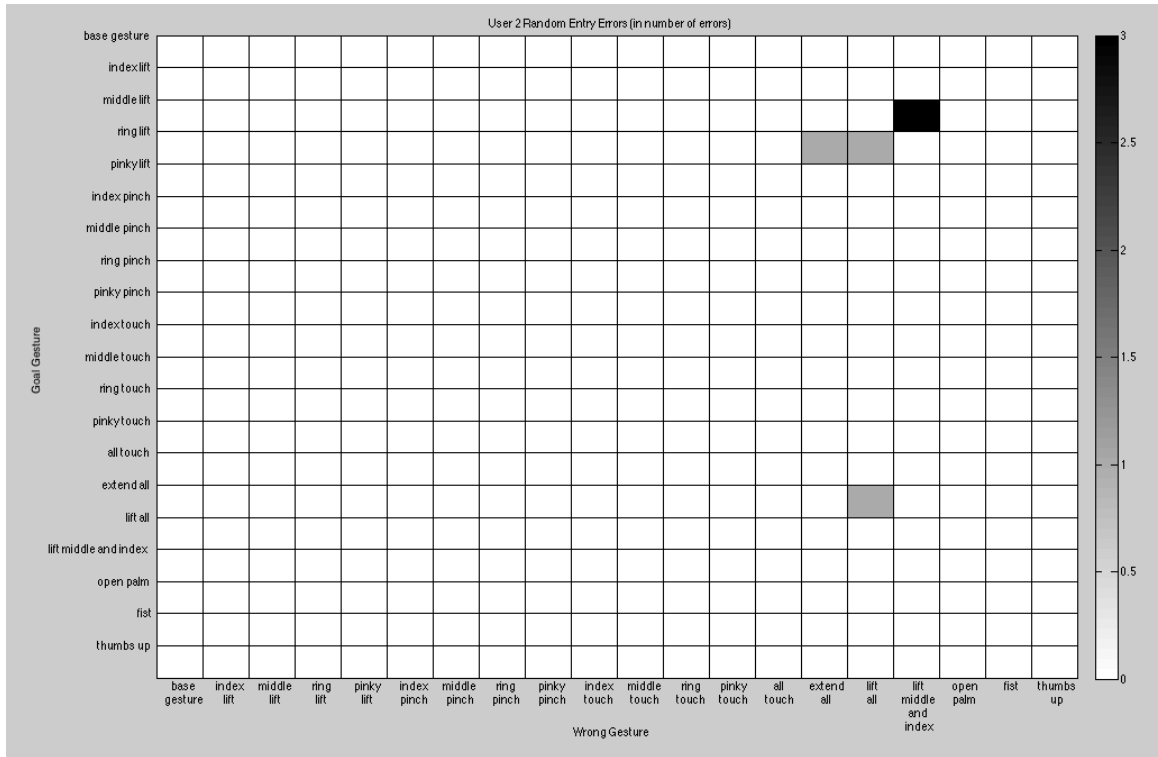


Figure 6-6: User 2 random entry error matrix

user to modify gesture input away from what is natural to him/her). We can see that when user 7 attempted the index pinch gesture, the system was mistaken for the middle pinch gesture. Similarly, we can see that the fist gesture was mistaken by the lift all gesture. Viewed from the perspective of the camera, corresponding hand blobs appear similar in such away that the system cannot differentiate between the mentioned gestures. For some users, these gestures are very similar and such errors are expected.

6.3.2 Randomized Transition Experiment

In this experiment, we hope to show which gesture transitions are faster on average compared to other gesture transitions. Rather than display one gesture graphic at a time, we displayed several. The user was asked to focus on the leftmost graphic gesture and anticipate the following gesture graphics to the right. The user conducted the gesture that was prompted by the leftmost gesture. After completing this gesture

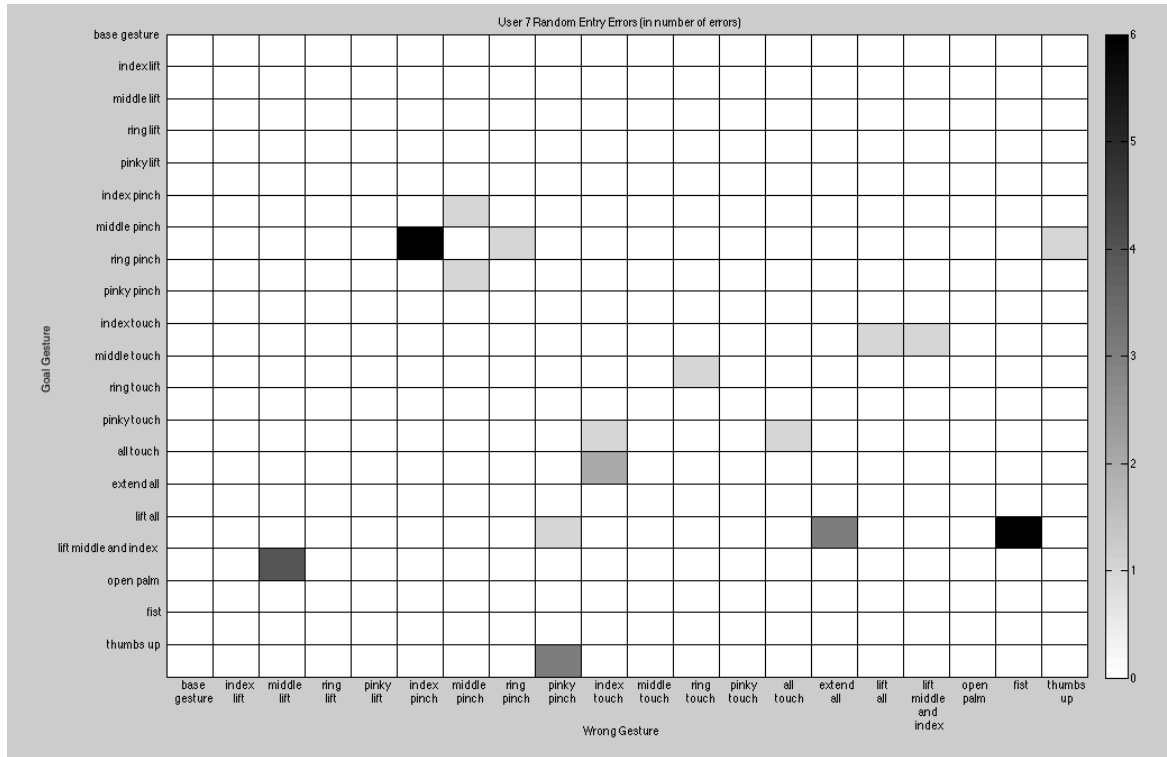


Figure 6-7: User 7 random entry error matrix

correctly, the list of graphics shifted to the left. We recorded errors as they were made yet did not give an indicator to the user that an error was recorded and did not ask the user to re-perform the transition. We examine average error gesture rates and average correct gesture input rate. Between each gesture, the user was asked to return to the base gesture.

Results

Figure 6-9 plots users on Y time and average speed on X. It shows both correct rates and error rates. For each user, there is a data point that describes a separate group gesture as explained in Table 6.3. Figures 6-10 and 6-11 show box plots describing the amount of time that passed between each gesture in gesture / second data points. Figure 6-12 show the average times for each transition between gestures for user 3. Figures 6-13 and 6-14 show the total number of transition errors for user 3 and user 5 respectively.

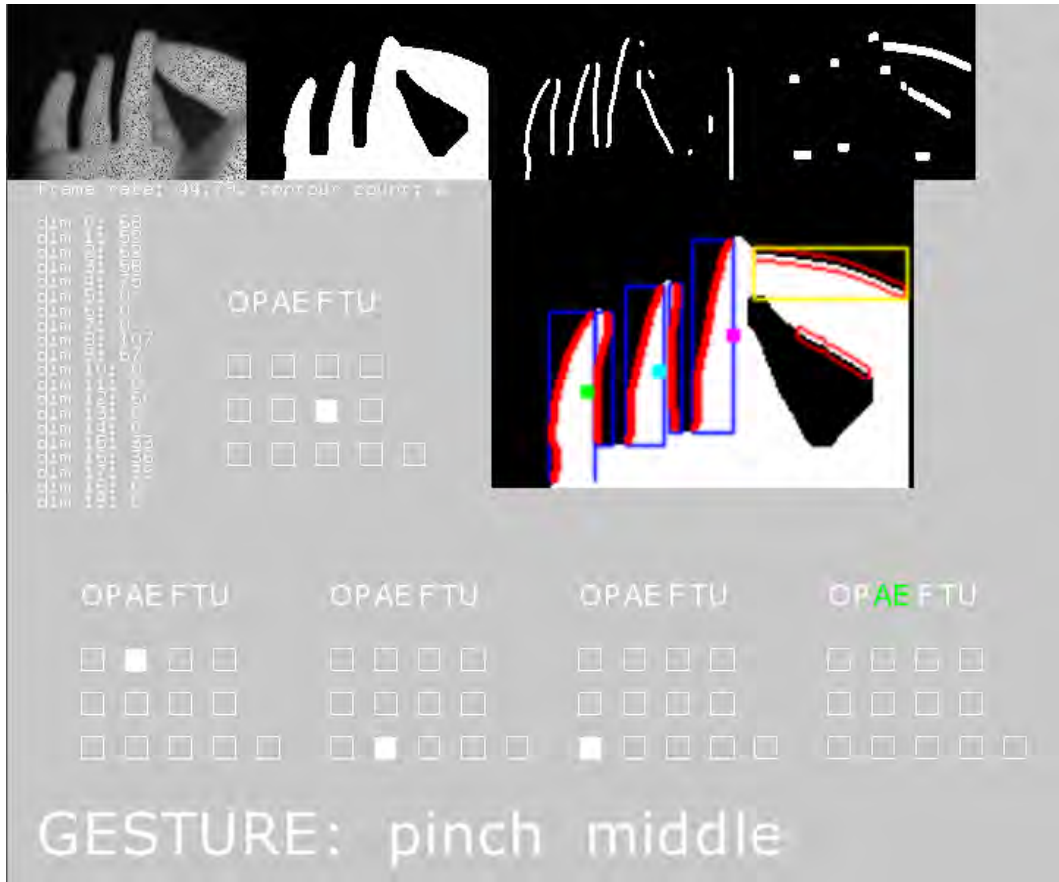


Figure 6-8: The user viewed this screen while conducting experiments. The current gesture graphic was displayed to the left of the camera video and gesture prompting was displayed underneath. The current gesture in text form was displayed at the bottom. At the top, we showed the current sequences of vision computational preprocessing.

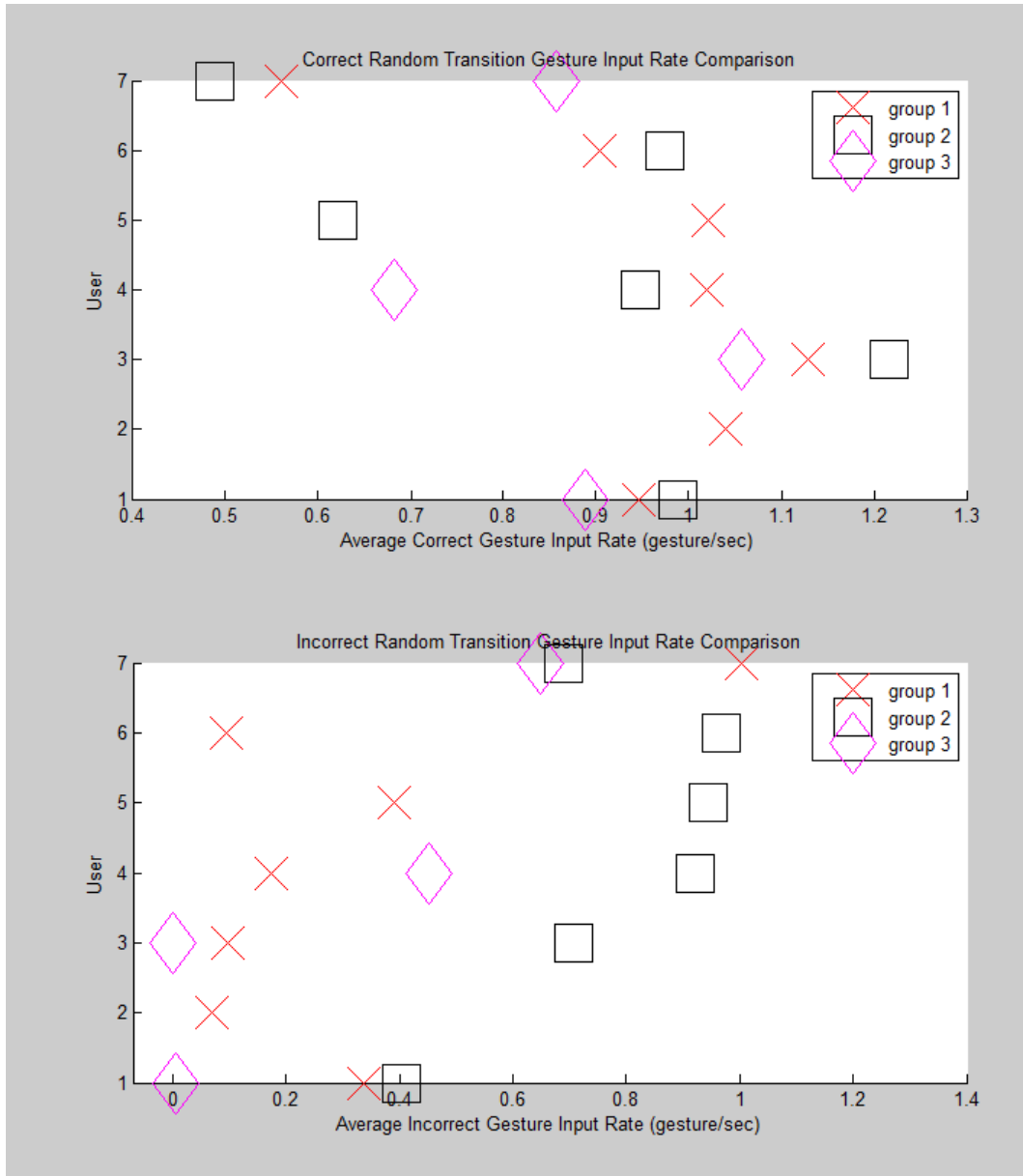


Figure 6-9: Random transition average group gesture rates and error rates. We see that the pinch gestures were more difficult to perform in transition between other gestures.

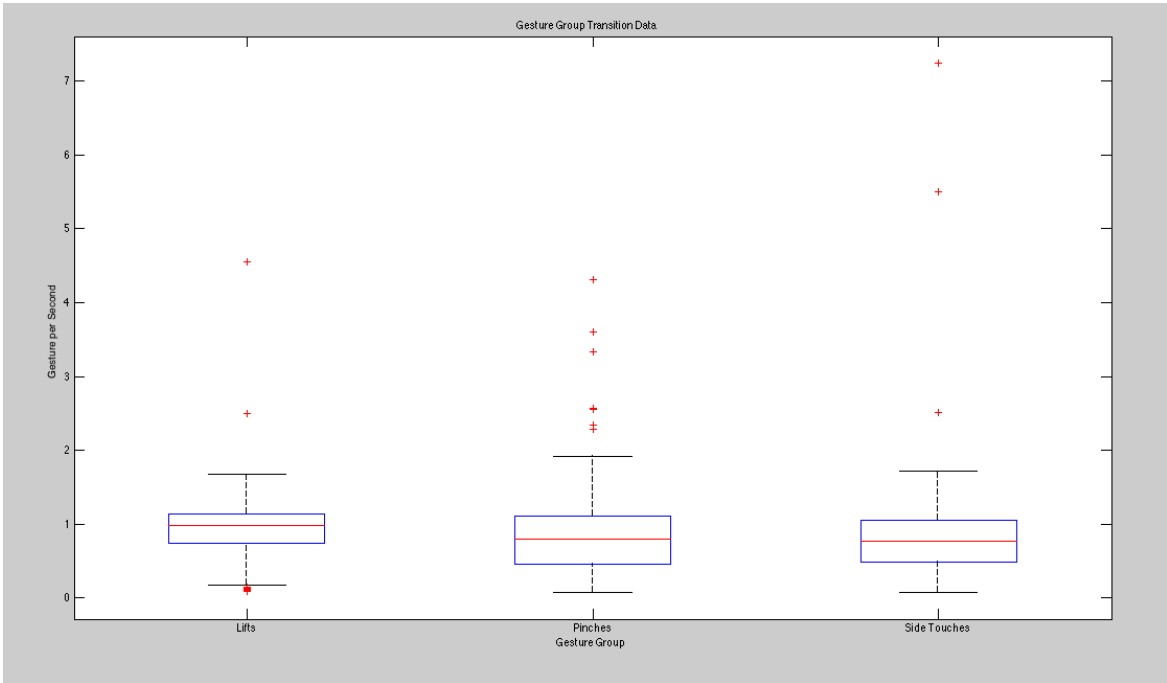


Figure 6-10: A box plot describing random transition time for each group in gesture per second.

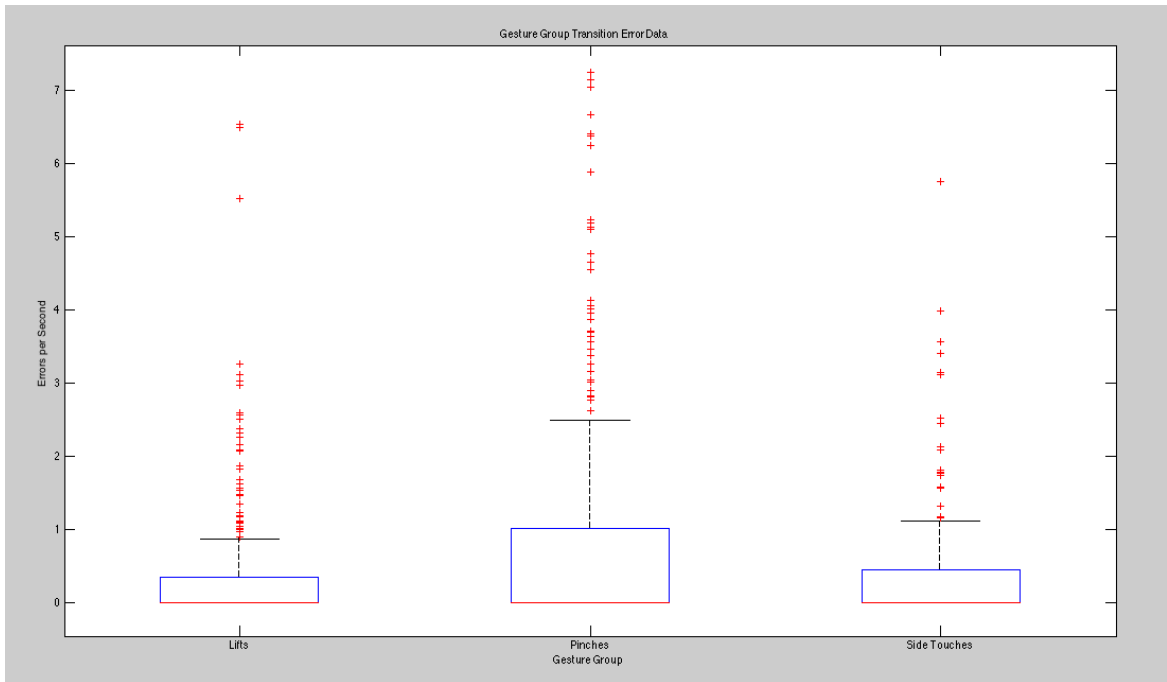


Figure 6-11: A box plot describing random transition errors for each group in incorrect gesture per second.

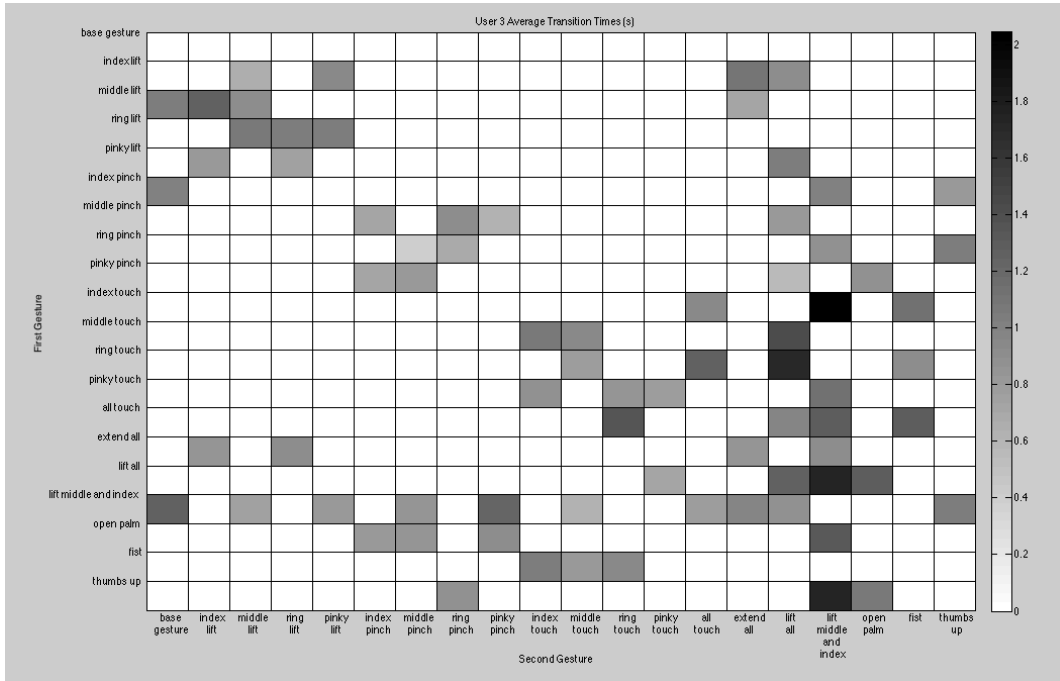


Figure 6-12: User 3 average random transition times

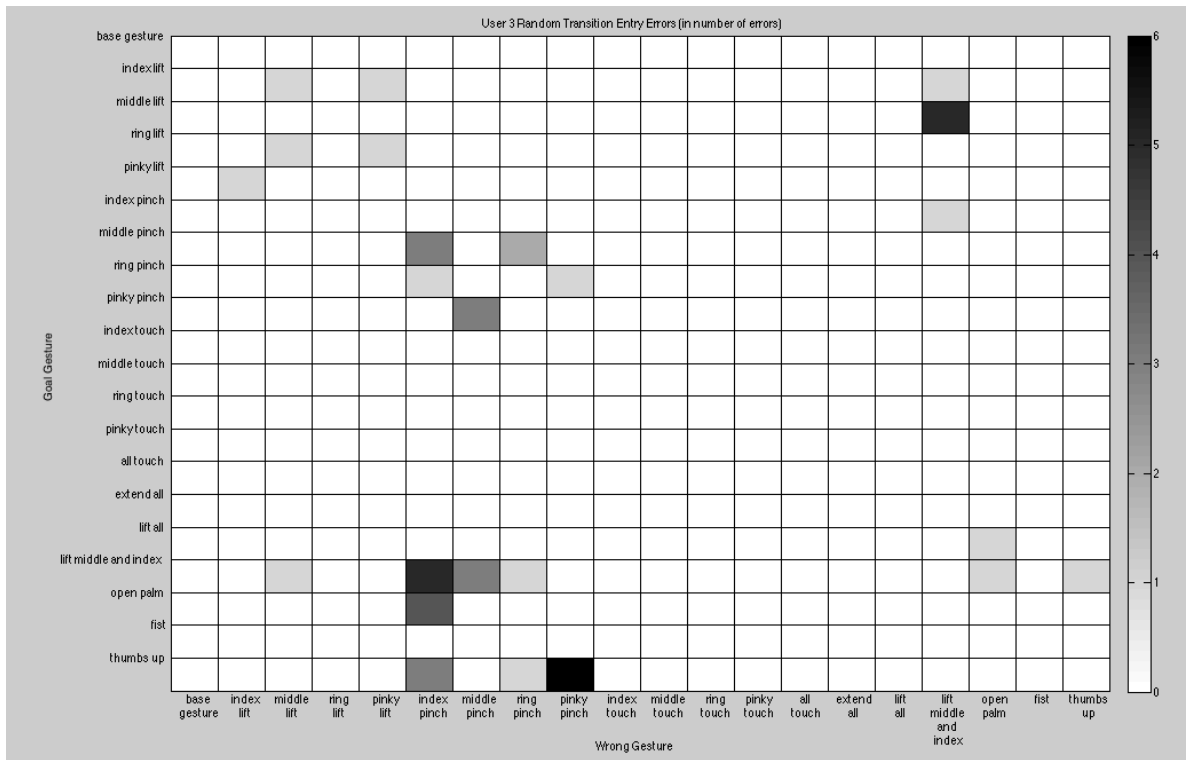


Figure 6-13: User 3 random transition errors

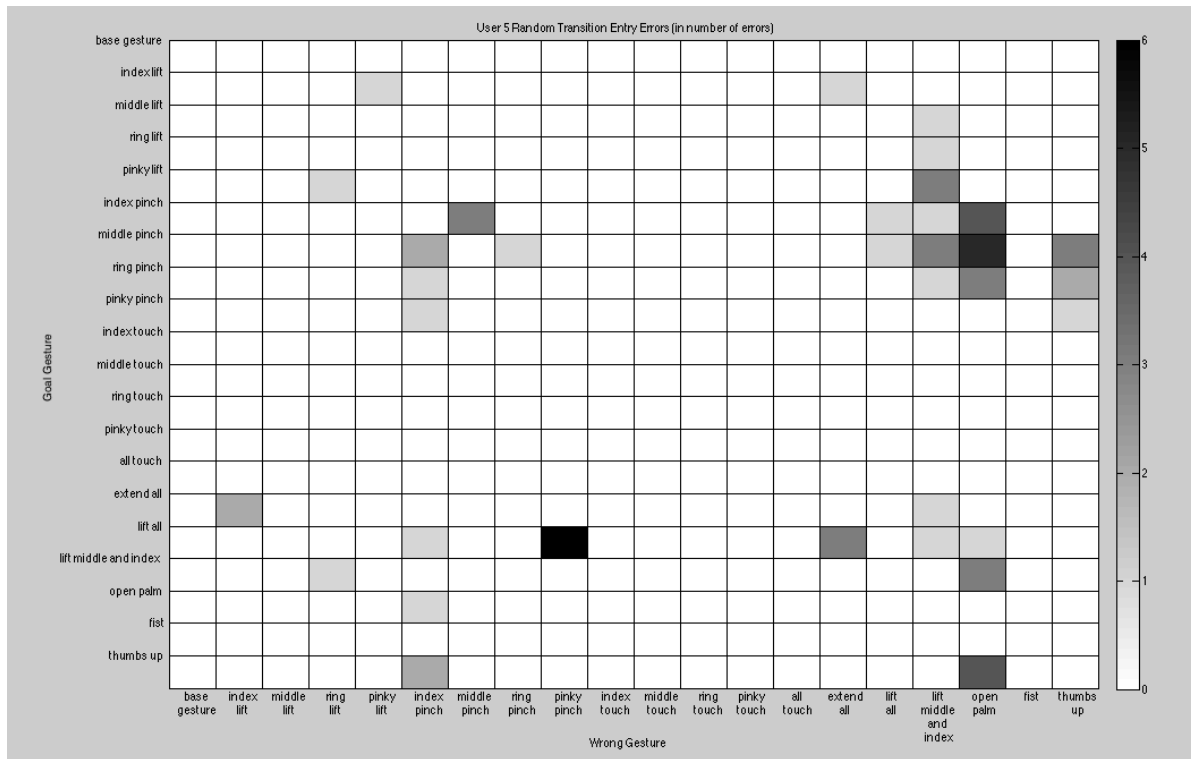


Figure 6-14: User 5 random transition errors

Experimental Conclusions

Based on the results given by Figure 6-9 and 6-11, it is clear that pinch gesture group gives a much larger average error rate than the lift group or external touch group. This may be due to the fact that the pinch gesture requires movement of a large number of hand components and is difficult to perform well on a consistent basis. The finger lift group (group 1) shows the most consistency in gesture speed and low error rates.

Figure 6-12 shows that user 3 was slower in performing transitions going from the lift all and index middle lift gestures to the pinch gestures. Given the larger amount of time required, it seems that user 3 would not prefer applications that use such transitions often.

The transition error matrices show which areas the users struggled with the most. Similar to most users, user 5 and user 4 (figures 6-13 and 6-14) had trouble performing the middle lift gesture and the middle index lift gesture such that the system could

differentiate between them. Some other areas which yielded high error rates were the pinky pinch and thumbs up gestures. This issue was discussed in previous sections - the shape the hand makes from the perspective of the camera is similar for both of these gestures and it is difficult to extract differentiating features. We can use these conclusions to determine which gesture designs are poor. In all it seems that gesture personalization as well as gesture separation will be very key when designing gestural applications.

6.3.3 Character Display Text Entry Experiment

In this experiment, we hope to compare the average correct gesture rates and average incorrect gesture rates between the three experimental gesture groups. Error rates were recorded similar to what was described in [21] - uncorrected errors and corrected errors were both accounted for. The experimental texts for each group are in table 6.5. As before, rates were recorded on a gesture per second basis. In order to compare errors, we examine the rate of incorrect gestures as well as the number of goal gesture / recorded gesture pairs. This experiment took about 4 minutes for each group to conduct.

Results

Figure 6-15 shows text entry input rates for the three different groups. Figure 6-18 shows user 1's goal / recognized error trend. Table 6.6 shows the average group correct / error gesture rates and variances across all users.

Group	text string
1	re re wq rewq rewq qwer we ew qr rq
2	fd sa as df df sa sd fa fdsa asdf
3	vc xzb bzxcv bzxcv bz bz cv cv xc xb

Table 6.5: Text entry strings for each gesture group. Since character mappings were chosen in accordance to gesture, proper sentences were not used.

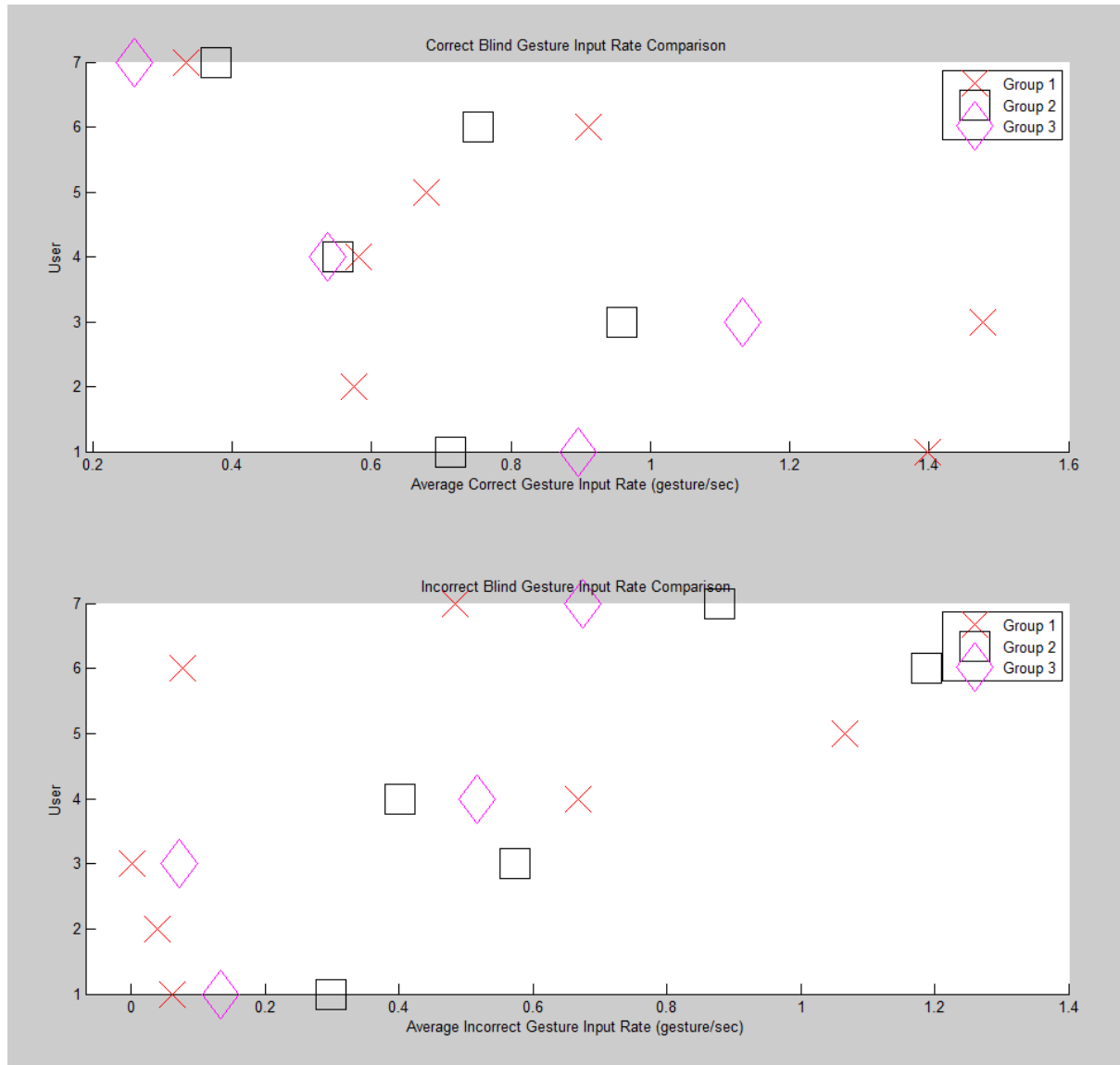


Figure 6-15: Text entry input rates for each user and gesture group. The data is incomplete, since some users had other commitments.

Text Entry Rates (gesture/second)	Group 1	Group 2	Group 3
Mean Correct Rate	.85	.67	.71
Variance of Correct Rate	.19	.045	.15
Mean Error Rate	.34	.67	.35
Variance of Error Rate	.17	.13	.086

Table 6.6: Variance and mean for correct / error gesture rates (gesture / second) over all users for each gesture group.

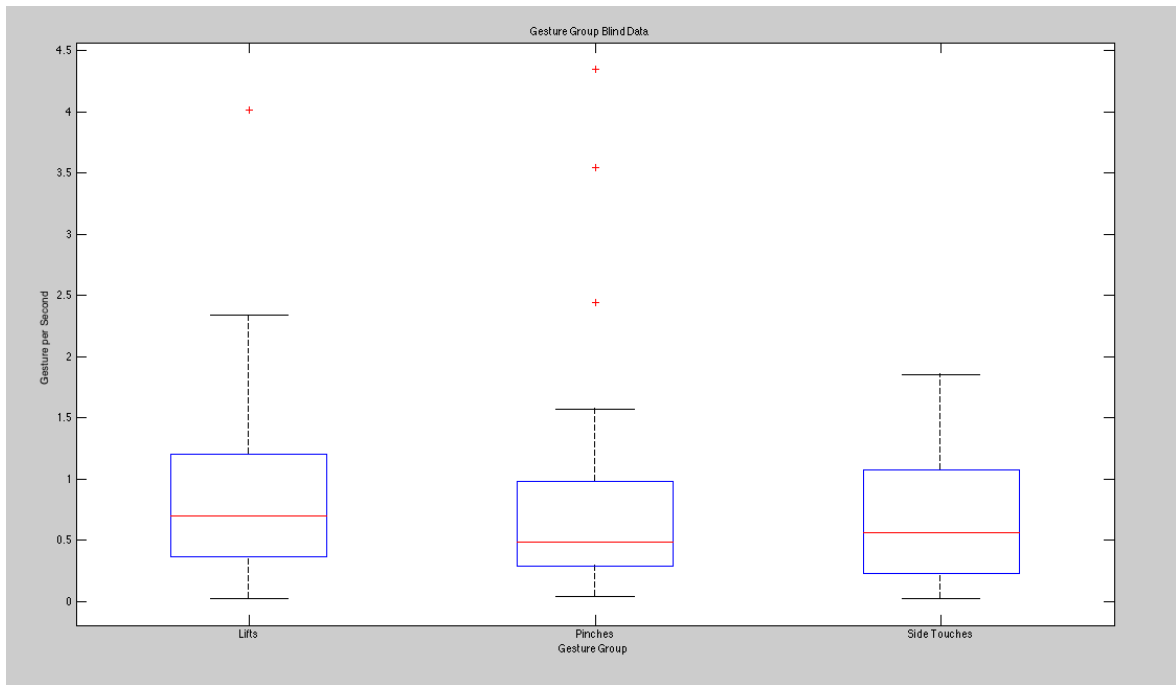


Figure 6-16: Text entry input rates (gesture/second) for each group over all users

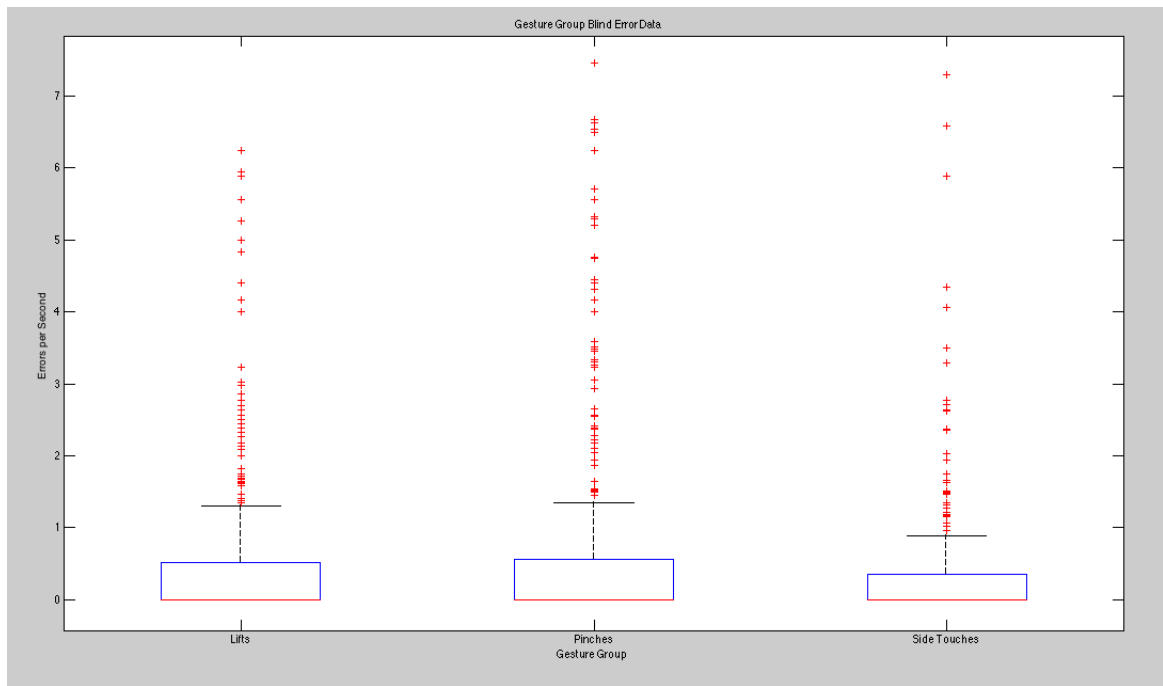


Figure 6-17: Text entry error rates (error/second) for each group over all users

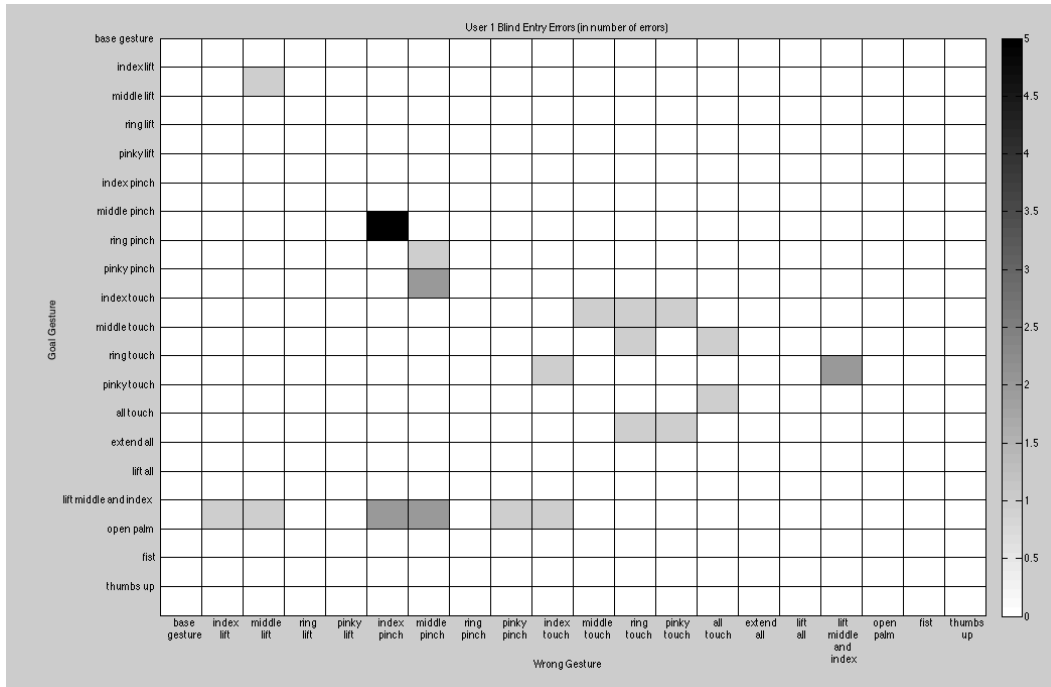


Figure 6-18: User 1 text entry error trends

Experimental Conclusions

As we can see, group 1 had on average the smallest error rate and largest correct gesture error rate. All gesture groups yielded slightly more sporadic input rates across users - some users were able to grasp the sensitivity and constraints of the system and modify interactions, while other users had more difficulty. We can see this from the larger variance values shown in figure 6.6. Users had difficulty remembering to return to the base gesture between pinch gestures - it was very natural to leave the thumb over the fingers while tapping on various fingers. This caused the system to often lose finger tracking as many fingers became occluded. Specifically, user 7 appeared to have extreme difficulty. Nonetheless, user 7 showed that the lifts produced the least number of errors. User 1 showed difficulty in differentiating between the index pinch and the middle pinch. Often users had difficulty separating their fingers during pinch gestures and the system thus could not separate which finger was being touched. This problem is expected: many users will not be able to adhere to the system constraints.

Comparing the results from the previous two experiments, we can see that the

Glass Entry Rates (gesture/second)	Group 1	Group 2	Group 3
Mean Correct Rate	.71	.69	.62
Variance of Correct Rate	.11	.03	.15
Mean Error Rate	.60	.93	.41
Variance of Error Rate	.15	.11	.069

Table 6.7: Glass entry variance and mean for correct / error gesture rates (gesture/second) over all users

variance of speed and error rate has gone up for gesture groups 1 and 2. This is expected, as the difficulty in completing a sequence of gestures correctly is more difficult than examining one gesture input sequentially.

6.3.4 Google Glass Display Text Entry Experiment

In this experiment, we hope to conduct the same experiment as the text entry experiment except via character prompting on Google Glass and observe performance changes. The display did not show the depth camera perspective. Often, a separate service 'PowerManager' on the Google Glass operating system interfered with the test. The service halted the update thread on the android application, causing users to often repeat performing a gesture given there was no haptic feedback nor visual feedback. This affected the results of the experiment such that the tests were often longer compared to the text entry experiment and users required to input more gestures. However, we can still extract differences by viewing the rates in which gestures were recognized while obtaining visual feedback from the Glass display.

Results

Figure 6-19 shows the average incorrect / correct gesture input rates for each gesture group for each user while viewing text entry on Google Glass. Table 6.7 shows the average incorrect/ correct gesture input rates and variances values across users for each gesture group. Figure 6-22 shows the goal gesture / recognized gesture error matrix for user 1 in contrast to figure 6-18.

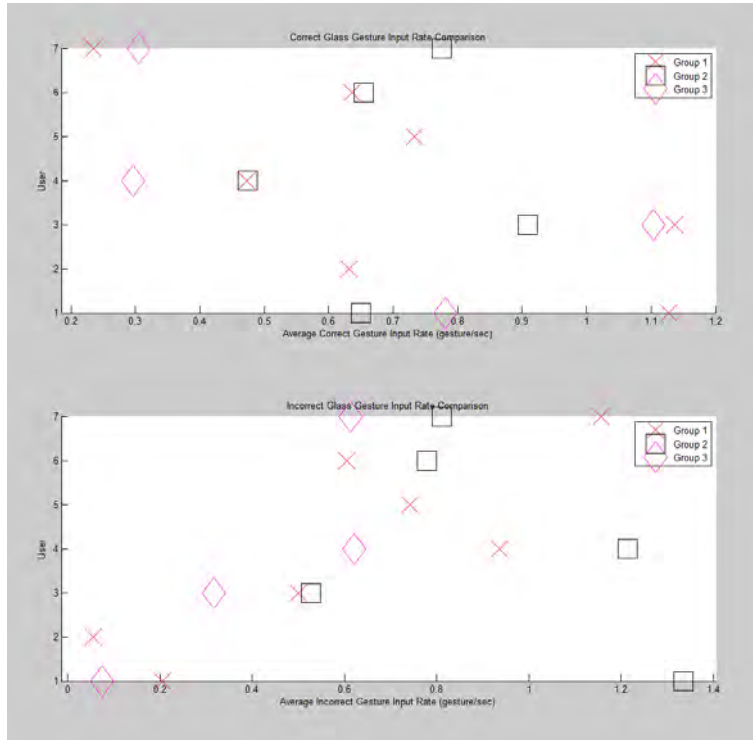


Figure 6-19: Google Glass text entry input rates

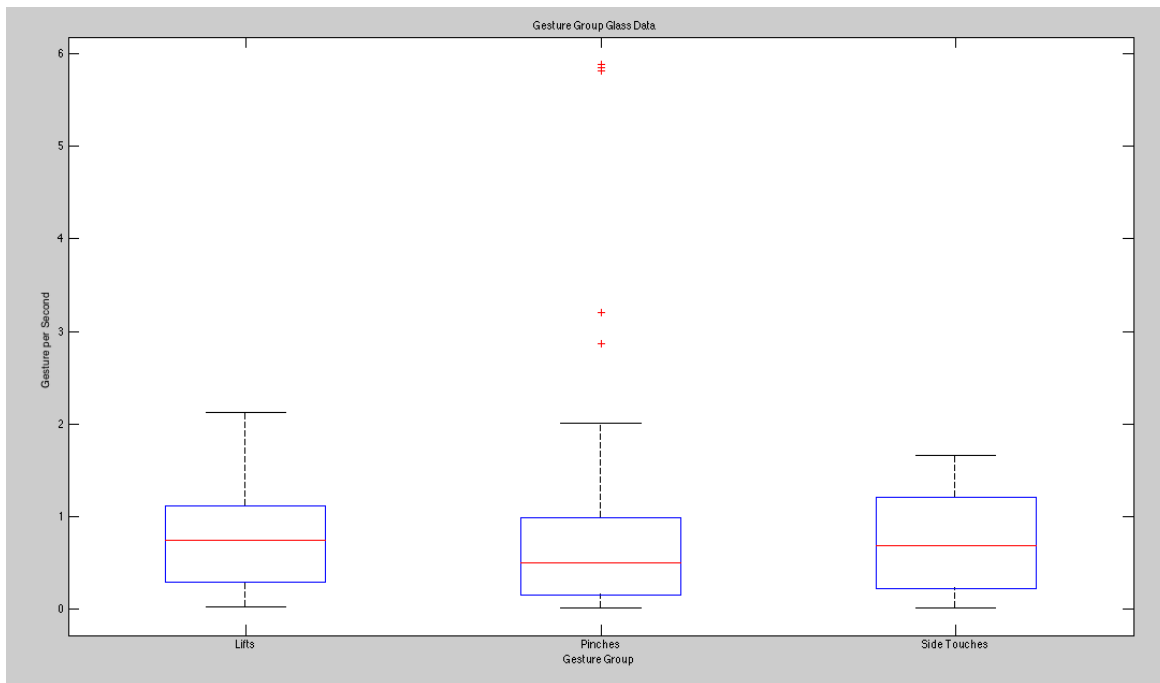


Figure 6-20: Box plot describing the correct gesture rate for each gesture group over all users

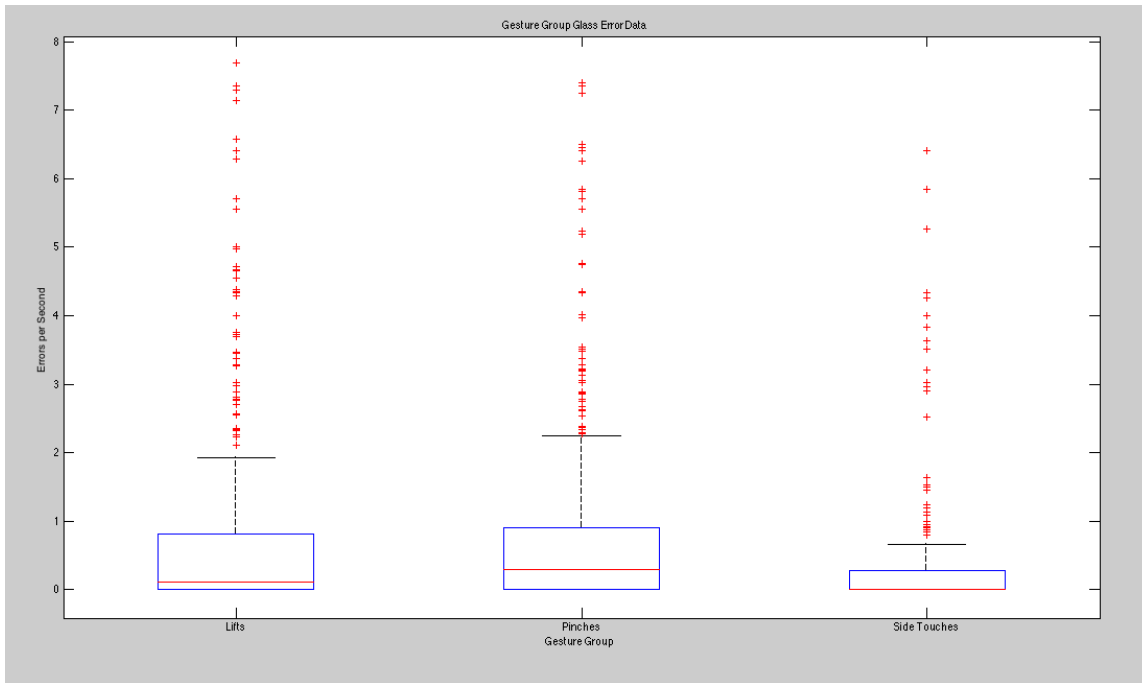


Figure 6-21: Box plot describing the incorrect gesture rate for each gesture group over all users

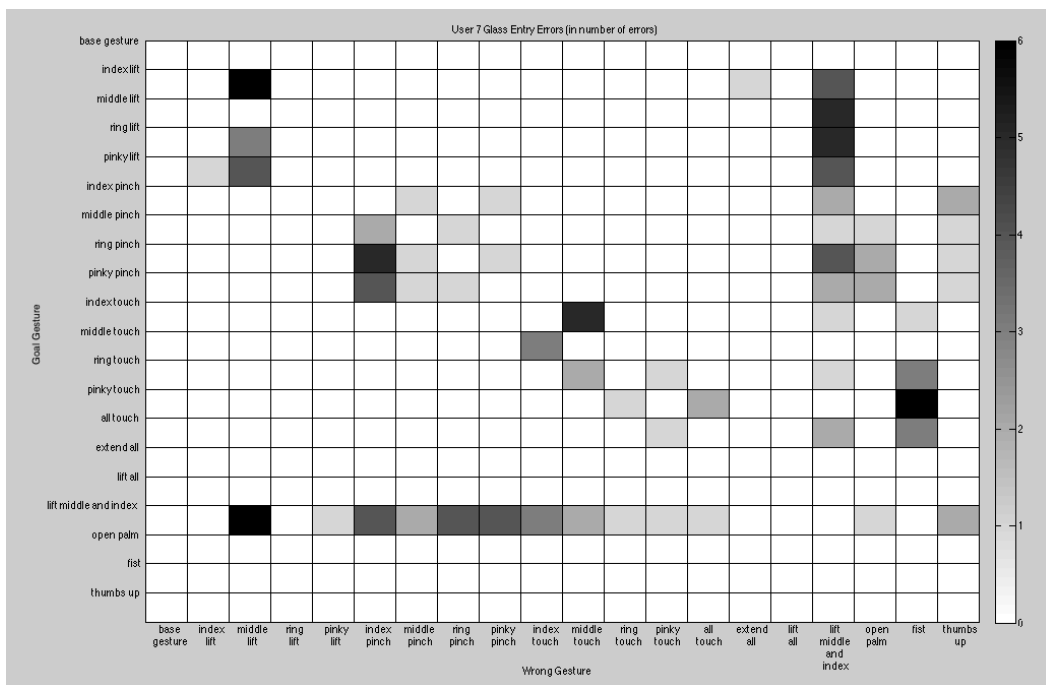


Figure 6-22: Glass entry user 1 text entry error trends

Experimental Conclusions

Based on the results in table 6.7, it is clear that the average correct input rate decreased for groups 1 and 3. The average incorrect input rate for all groups is much higher (by about .3 seconds). Given the increase in error rate as well as the increase in number of errors given by figure 6-22 we can determine that it was more difficult for users to use Glass as opposed to desktop display visual feedback. The variances appear to be similar in both experiments. Interestingly, we can see from table 6.7 and figure 6-19 most users do slightly better in group 2 when receiving visual feedback on glass. The decrease in correct gesture rate between experiments for group 3 is also less than group 1. The increase in error rates for groups 2 and 3 is also less than group 1 between experiments. Based on these distinctions, we can make hypotheses as to why Google Glass may be difficult to use with EMGRIE. There is less visual feedback of what the depth camera sees while wearing Google Glass. Gesture group 2 and 3 gives the user more physical feedback when performing - the thumb touching the finger tells the user that the hand is in the proper position and the finger touching the side tells the user that he/she is performing the gesture correctly as well. This increase in sensory perception may help the user perform the gesture as opposed to very small sensory changes when users perform the first gesture group. In contrast, some users were able to perform all gestures consistently well - more exploration is needed to determine why these users were successful.

6.3.5 Expert User Learnability Experiment

In this experiment, we hope to model the learning rate of a user over many sessions via the character display text entry experiment. We chose to use an expert user for this experiment, where this user had familiarity with EMGRIE (5 hours a week for several months). Originally, users had difficulty learning more than 7 gestures in one session. Given the user was an 'expert', no gesture groups were needed and the full 19 gestures were tested during the experiment. Each session took about 2 minutes to complete as was done every three days for about one month.

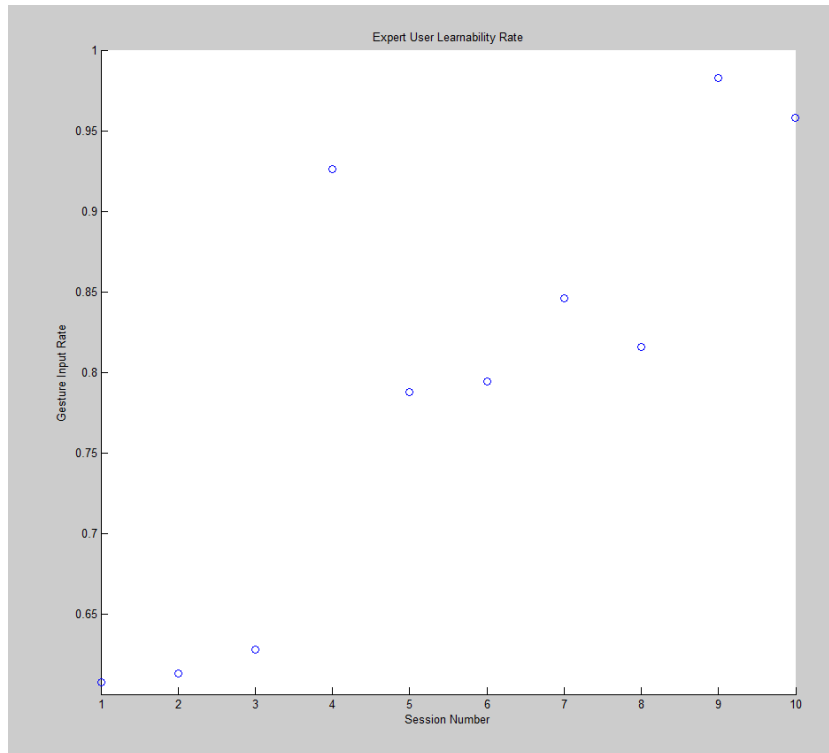


Figure 6-23: Expert user correct gesture input rates over sessions (one session every few days for one month).

Results

Figure 6-23 shows the gesture rates given by an expert user over several sessions. The expert user had access to practice gestures for several months before starting the experiment. However, the user did not practice gesture input under experimental conditions until the test began.

Experimental Conclusions

As we can see from the data points, the user continues to improve throughout the sessions as expected. We have yet to see where the limit lies in terms of correct gesture input speed. While the maximum is currently about 1 correct gesture/second, the limitation appears to be the amount of concentration and thinking required to perform each gesture required. Since this can be improved with time, it is unclear as to how fast a user's correct gesture input rate can be. Given that the user appeared to

make more mistakes for attempting to perform faster, we hypothesized that continued training would not improve input rate further.

6.4 Text Entry Application Design Discussion

There are vast possibilities to explore when discussing gestural text entry application design. For instance, we primarily used error rates and task times to evaluate interactions. While this gives a general conclusion as to which gestures are better suited for various tasks there is much potential for a deeper exploration into memorability, comfort, ease of execution, and variation between hand positions during gesture articulation. Specifically, we described several gesture designs (gesture groups 1, 2, 3) and gesture to character mappings (application design iteration 1 and 2) but there are many more possible application designs. In order to properly determine which gestures map well to characters, techniques described in various references [31] should be used.

However, based on the results from the described experiments, we can show which gestures may help users the best in terms of inputting text onto a Google Glass interface (or any other device). It seems that the lift group produced the fastest and least error prone results outside of the Google Glass interface. The pinch group produced the least increase in error rate when used with Google Glass, while most users had varying experiences with using EMGRIE with Glass. Based on the expert user experiment, we can see that practicing would improve the power of EMGRIE.

We wanted to test to see the initial interactions and responses from users when faced with inputting text for the first time using gestures. The conclusions show that text entry gestural use case is extremely difficult and requires natural talent / training. We would like to note that standard text entry (keyboarding, etc) ability, while common, is very difficult and often requires extensive amounts of training. We believe that while gestural text entry is powerful, users would most likely exploit EMGRIE for simpler tasks in the short term.

Chapter 7

Tidmarsh Focus Selection Application

7.1 Unity and Data Visualization Application

Design Introduction

In this section, we explore an extremely different mode of interactive computation with Google Glass in conjunction with Unity (a popular game engine). We were able to utilize Unity in creating two different data visualization / focus selection Android applications to test gestural application usability. Given the Unity Android applications, we could side-load them onto Google Glass for user testing.

Given the fact that focus selection applications do not require as large a command set as text entry, we believe that such Google Glass gestural applications would have improved usability over text entry applications and hence be more useful to users. Smaller command sets increase the minimum physical difference between various gestures which decreases potential user and system recognition confusion between gestures.

We also changed the experimental design when evaluating the applications. In this case, we ask empirical questions concerning gesture to command association strength and the effort involved to perform that gesture. Through such experimental results,

we can determine which gesture to command correspondences were of poor/good design and which gestures are easy/difficult to perform.

7.2 Interface Design I

For our first application design, we were tasked with designing a data visualization application for use with Tidmarsh (tidmarsh.media.mit.edu), a 600 acre span of forest and marshland in southern Massachusetts being outfit with hundreds of multimodal sensors that can be browsed in a game-like augmented reality application. Hypothetical users, physically traversing the marsh, view sensor data on Google Glass detailing data from the current area of the marsh the user is located in. In order to select the area of the marsh the user is interested in, he/she issues the corresponding gestures to move the sensor focus to various discrete pre-determined locations. Experts would choose these locations based on the geographical layout of the marsh and the commonalities of certain areas. Similarly, the user issues gestures to select the form of data he/she is interested in. Table 7.1 shows the gesture to action mapping. There are two modes - data selection mode and camera selection mode. Figure 7-1 shows the screen in camera selection mode. The box in the upper left corner details which camera position the user is currently in by highlighting the corresponding number green. In the current Tidmarsh application, camera positions were arbitrarily chosen for demonstration purposes, and there are two altitude levels of camera positions. The higher level has 6 camera positions in a grid-like fashion and the lower level has 2 camera positions beneath the grid.

Gestures were chosen based on the previous experiment's results: gestures that were easy to perform and statistically different from each other were hypothesized to yield the best user experience. For instance, we noticed that the pinky pinch and fist gestures were often confused by the system and hence the pinky pinch was replaced with the ring pinch. Since only a few actions were needed, we were able to cover all actions without choosing gestures physically and statistically close. We also wanted to choose gestures that related to the action they were associated with

Gesture	Camera Mode	Data Selection Mode
Lift All	move camera West	move cursor left
Extend All	move camera East	move cursor right
Index Pinch	move camera North	move cursor up
Ring Pinch	move camera South	move cursor down
Index Lift	move camera to higher level	toggle data selection
Index Extend	move camera to lower level	toggle data selection
Fist	toggle mode	toggle mode

Table 7.1: The gesture to action mapping for the discrete Tidmarsh application

on the application level. To move the camera left, we hypothesized that an intuitive sweeping motion to the left with all of the fingers (the lift all gesture, when the palm normal is pointing left relative to the user) would yield strong user experience and usability data. Similarly, we hypothesized that the index pinch, ring pinch, and extend all gestures represented the other cardinal directions in a similar fashion. The index extend and index lift gestures are reminiscent of pointing, by which we hypothesized that this gesture should relate to the camera movement of forward and back (or up/down in altitude levels). A gesture (fist) unrelated to the other gestures was chosen to switch camera/data selection modes, and all gestures mentioned are easily detectable by the EMGRIE system.

In terms of gesture spotting for both application design iterations, we decided to leave the "start and stop listening for gestures" action to be specified as the user prefers. The gesture to start recognizing gestures must be non-natural yet still easy to perform.

7.3 Design I Experiment

To evaluate the application, we examined the task times of two tasks: task 1 involved moving the camera from position 1 to position 6, and task 2 involved moving the camera from position 6 to position 1. The user was free to move the camera through any sequence of actions. After learning the gestures and application, the user then participated in a survey where the user was asked to rate the level of association



Figure 7-1: Tidmarsh application interface for discrete camera positions. The highlighted green number in the upper left indicates the current focus position. The views are the area from the birds-eye perspective.

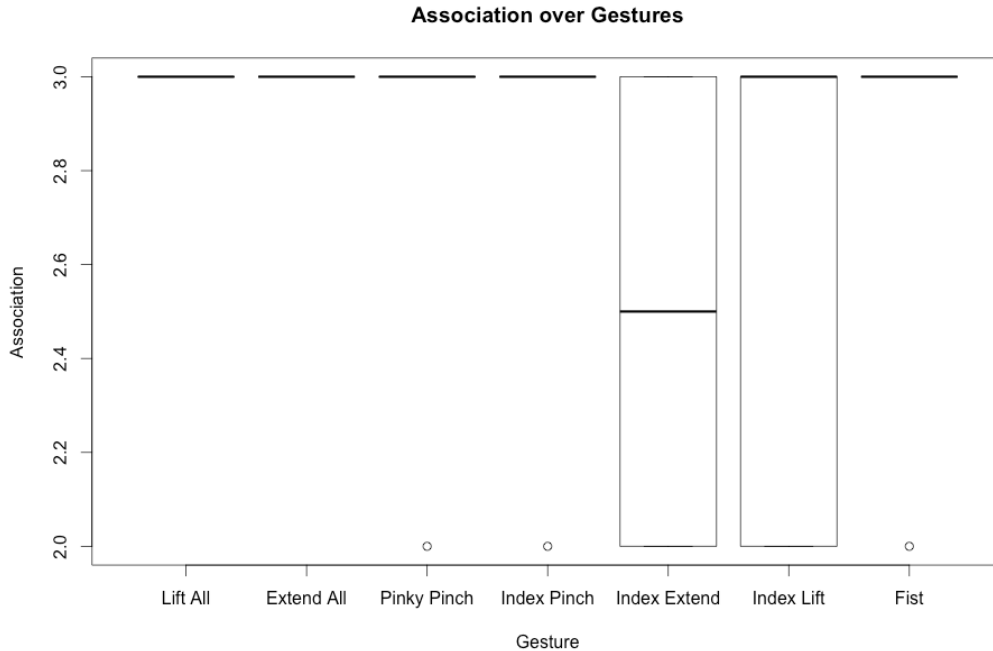


Figure 7-2: The box plot for each gesture and the corresponding association for 6 users in a user survey. A 3 indicates a level of strong association, and 2 indicates a medium level of association.

between the action and corresponding gesture as weak, medium, strong. The user was then asked to rate the level of effort required to perform such gesture on the Borg scale [1, 10] where 1 is no effort at all and 10 is maximal effort. 3 users were asked to complete task 1 and 3 users were asked to complete task 2 as quickly as possible. In all, 6 users took the survey giving their opinions on the association level and effort level.

In the study by Stern [31], each gesture was tested for each association. While we think that this method is correct, we believe that it is unnecessary.

7.3.1 Results

The following box plots (Figures 7-2, 7-3, 7-4) show the results for surveyed association level, surveyed effort level, and task times respectively. ANOVA results showed no effect.

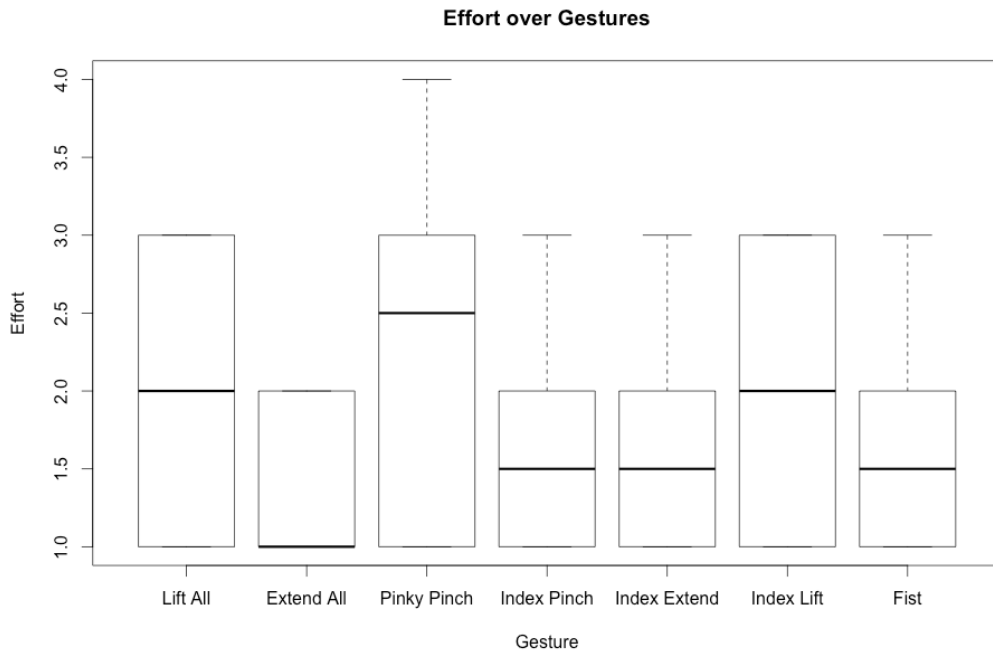


Figure 7-3: The box plot for each gesture and the corresponding effort level on the Borg scale for 6 users in a survey.

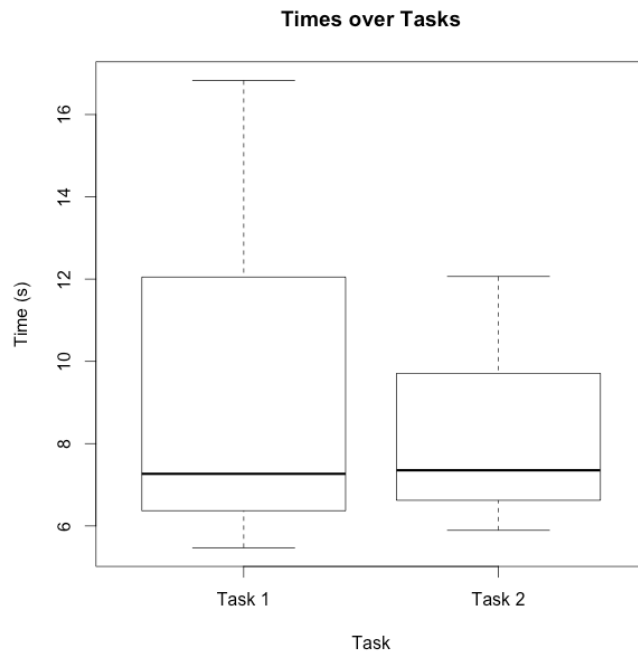


Figure 7-4: The box plot for each task and the corresponding discrete task times.

7.3.2 Result Discussion

Based on Figure 7-2, we can see that our hypotheses that our intuitive gesture-to-effect design yields good user experience was "on track". The association label for each gesture to command was mostly 'strong', which we can take to be a good sign that such gestures are good application design for those actions. However, to prove beneficial user experience, we must study many other facets of usability.

Figure 7-3 shows that the median level of effort required for most gestures varies between 1.5-2. On the Borg scale, is this minimal effort - which follows our hypothesis that microgestures are very easy to perform.

Task times are always skewed in the positive direction, hence, it is often best to use the median time to compare. The tasks studied utilized very different gestures to complete the task. Given the similarities between the task times, we can conclude that in the application space, microgestures have little effect on the task time for this particular task. From our ANOVA analysis, we find that microgestures, in this space of application, all have similar effects and yield similar results examining task time, effort, and fit-to-function.

7.4 Interface Design II

In our second application design, we took what we found in our first iteration and built similar gestural mappings for camera manipulation in a continuous manner. For instance, the user uses the continuous gesture modality for index extend / index lift gestures to move the camera forward and back. Table 7.2 shows the gesture to action mappings. Given that there are 6 different dimensions in which the camera can move (forward/back, left/right, up/down, tilt, roll, yaw), we decided to simplify the application such that only 4 dimensions are utilized. Figure 7-5 shows what the application screen may look like to the user. The gestures in Table 7.2 were chosen based on the same criterion used in the first application design iteration. We included the Index Middle Together Lift/Extend continuous gesture and the Pinky Ring Middle Together lift/Extend continuous gesture to handle stafe (side to side

Continuous Gesture	Action
Middle Pinch Slide	move camera higher / lower
Index Lift/Extend	move camera forward / back
Index Middle Together Lift/Extend	turn camera left/right
Pinky Ring Middle Together Lift/Extend	strafe camera left/right

Table 7.2: The gesture to action mapping for the continuous Tidmarsh application



Figure 7-5: Tidmarsh application interface for cont camera positions

motion).

7.5 Experiment Design II

To evaluate the application, we examined the task times of two tasks: task 1 involved moving the camera from one way-point (seen in Figure 7-5 as a floating orb) to another way-point in the unity game space. The second task involved a similar motion but required different gestures to complete the task. The user was free to move the camera through any sequence of actions. After learning the gestures and application, the user

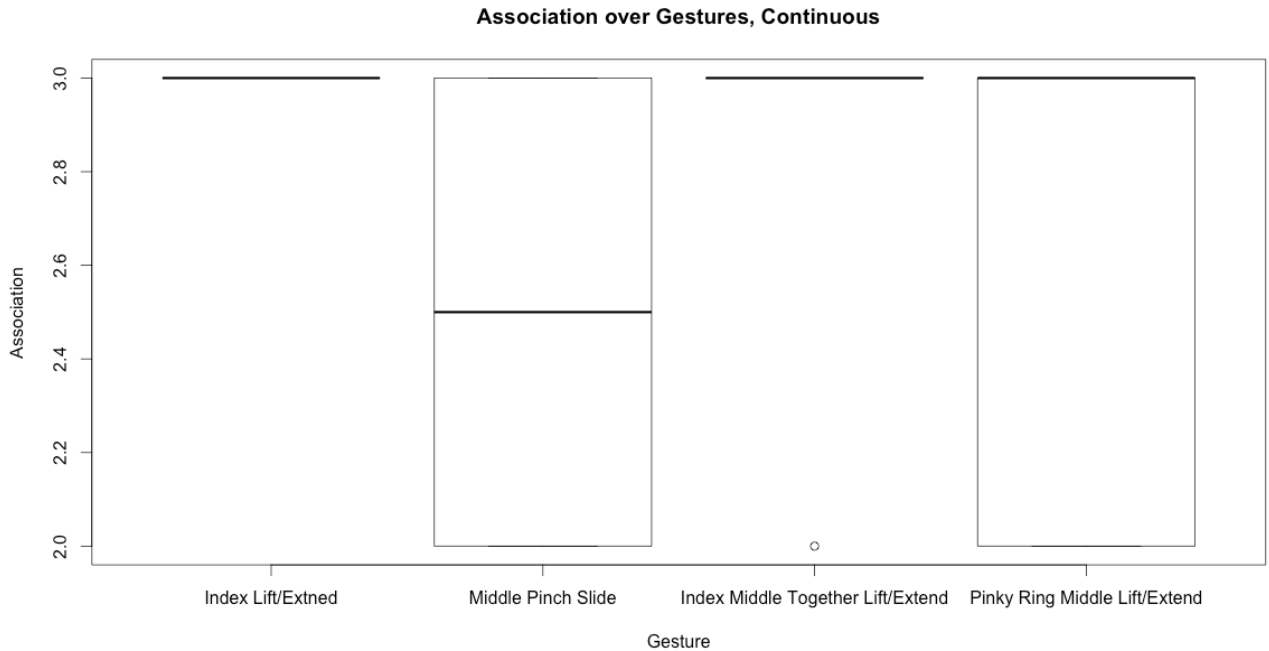


Figure 7-6: The box plot for each gesture and the corresponding association. A 3 indicates a level of strong association, and 2 indicates a medium level of association.

then participated in a survey where the user was asked to rate the level of association between the action and corresponding gesture as weak, medium, strong. The user was then asked to rate the level of effort required to perform such gesture on the Borg scale [1, 10] where 1 is no effort at all and 10 is maximal effort. 3 users were asked to complete task 1 and 3 users were asked to complete task 2 as quickly as possible. In all, 6 users took the survey giving their opinions on the association level and effort level.

7.5.1 Results

The following box plots (Figures 7-6, 7-7, 7-8) show the results for association level, effort level, and task times respectively. ANOVA results showed no effect (as was expected). Task times at this scale (20-40 seconds) produced by a series of gestures have large variation.

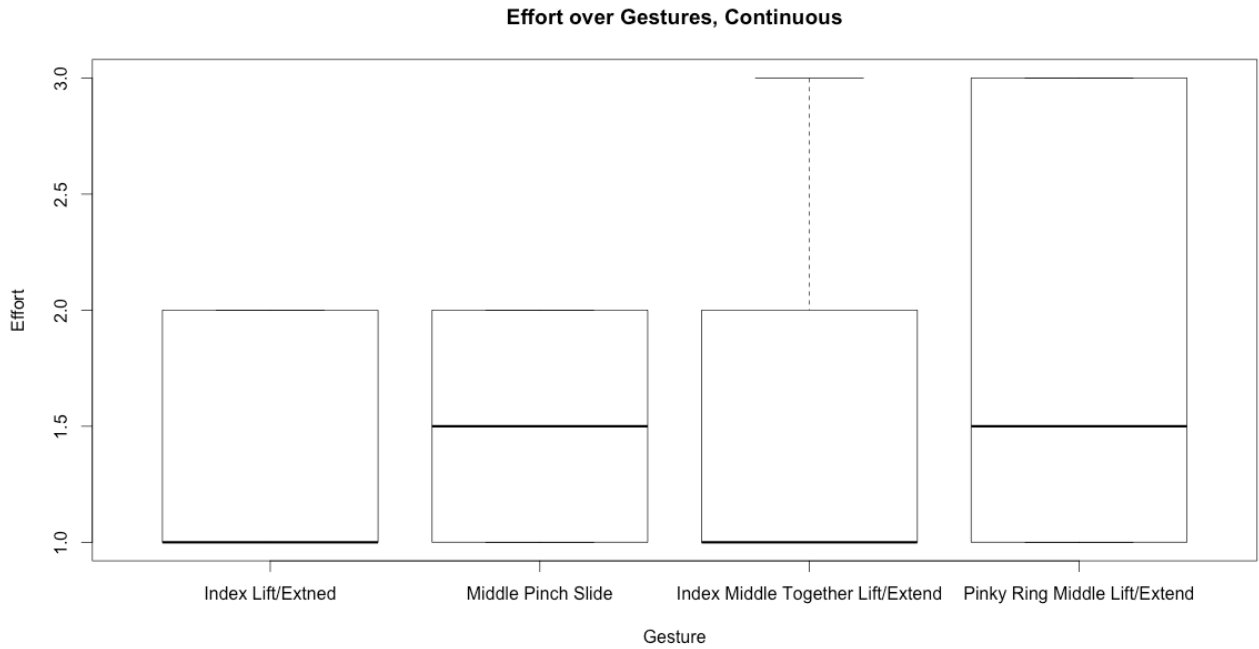


Figure 7-7: The box plot for each gesture and the corresponding effort level on the Borg scale.

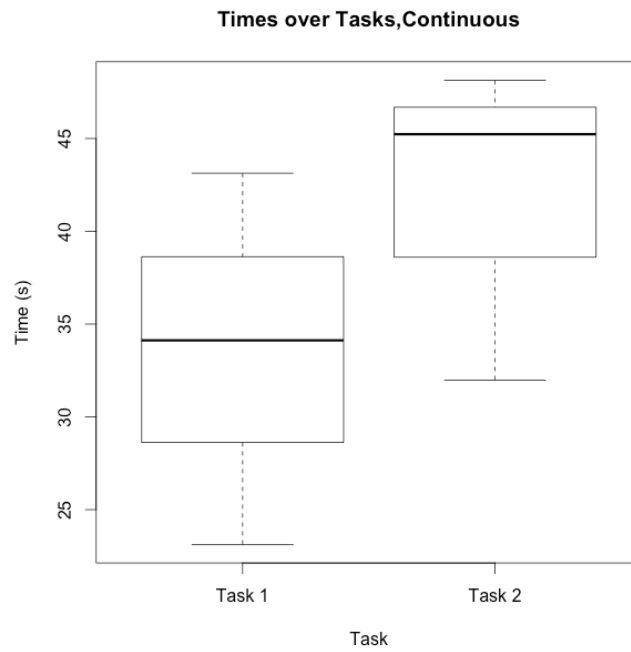


Figure 7-8: The box plot for each task and the corresponding task times.

7.5.2 Result Discussion

Based on the figures shown, we can see that there is little significant change between the last experiment on discrete camera positions and the continuous camera position application. The tasks we asked the users to do should not be compared, as the rate of which the camera moved was held as a maximum value in this design. Hence, it took significantly longer for the user to move between focus locations compared to the discrete location experiment. Both camera movement tasks require about the similar amount of time from the user. Based on the association and effort plots, the gesture choice stated for continuous camera movement appears to yield good usability and intuition. Similar to the previous application design, this application design yielded microgestures that had no linear effect on fit-to-function, task time, and effort (results were consistent for each microgesture).

7.6 Focus Selection Design Discussion

In summary, we believe that microgesture works well with simple applications of this sort (e.g., moving a focus of attention). Focus selection applications require few actions and need to be easily accessible by the user. Since hand configurations are very easy to perform quickly and easily accessible (compared to other hand-involved head mounted display user interfaces), the applications described in this chapter are very good solutions for accessing data on the go.

It is important to note the methodology in how we designed both applications - based on the movements of the hand, we attempted to match the gesture with a logical camera movement. The results show that our matching was intuitive to the user and easy for the user to perform. We believe other applications should be designed in the same manner.

Chapter 8

Future Work

Throughout the course of developing microgestural applications, we often found that there were many possible directions in which to take the research for which we did not have the resources. For instance, we mainly focused on task times, what they tell about certain microgestures, and how they relate to each other. We would like to extend this research to look more closely at memorability and learnability. Experiments examining memorability and learnability require extended-period experiments beyond our level of expertise and time availability. Such aspects may yield more information on which microgestures are more quickly learned on average and hence provide better user experience or insight towards a universal microgestural language. Other empirical evaluations we only touched on concerned fit-to-function and ease of execution aspects. Such aspects are important, easily evaluated, and should not be ignored in future experimentation [25]. We would also like to continue to examine the effect contrast bias has on microgesture. If it becomes clear that microgestures related to everyday natural circumstances require substantially more cognitive processing to perform compared to more unusual (but simple and easy) microgestures, then developers could use such a principle to greatly increase user experience in the realm of microgesture application.

Based on the constraints of a vision-based system that employs a bag-of-words machine learning model, it is clear that a stronger system could be built that depends less on the shape and natural gesture positioning of the user's hand. Some users had

difficulty using the system given the dependencies of the required separation of the fingers and horizontal positioning of the thumb. To prevent these problems, feature extraction techniques must be improved such that processing employs non-trivial thresholding, improved seam-carving [5], and less relational blob dependencies. To further improve the system, we can leverage other modalities as well. The microphone helped improve certain cases where users allowed their thumb to hover over the fingers, but the microphone was not fail-safe. By adding more microphones to more areas of the wrist and recording the responses from each of the microphones for each finger, finger pinch gesture recognition should improve [4]. Unfortunately, this does not entirely solve the problem of when the user fails to emit vibration in the hand when performing the gesture - this problem may only be preventable through training and/or ongoing vision based gesture recognition.

We would also like to spend more time examining other more difficult-to-perform gestures. While text entry requires many different gestures, there are still many more gestures that the hand can perform and deserve to be examined as possible gesture choices in applications. For instance, many users find it difficult to perform adjacent finger contact gestures (such as the Vulcan sign from star trek) but we believe that to more advanced users the power of the hand may be under-utilized. In conjunction with more difficult microgestures, we very nearly decided to build a second EMGRIE system and study user reaction to performing two microgestures at once (while wearing two depth cameras on each hand). This would decrease the level of complication required to cover all actions in an application, and we emphasize that future studies should include this power.

Finally, we want to explore more possible applications. The possibilities to test microgesture applications are seemingly endless - specifically in the realm of Internet of Things. Given correct gesture spotting implementations, microgestures can possibly interact with everything - doors, light switches, TV remotes, anything that may possibly be digitized. It would be very interesting to see such applications in action.

Chapter 9

Conclusion and Discussion

It is difficult to design a microgestural application. Limitations of the human body, the nervous system, and sensor framework yield simultaneous constraints that must designers abide. For several years, researchers have delved into the aspects of similar multi-disciplinary work, and it is clear that many of such limitations are well understood. However, we believe that not enough emphasis has been placed on the power and usability of the microgesture. Microgesture (especially where the sensor is located on the wrist) allows the hands to be free of devices, requires minimal effort, and through good application design, allows for intuitive and efficient interaction.

We have seen through application case studies that many factors influence the efficiency (task time) of an application. For instance, we saw that single discrete microgesture performance, visual feedback, and task prompting affects the user's correct gesture input rate. It also appears that through the pilot learnability study conducted over several weeks with an expert user, learnability may have a larger effect on efficiency than the previously mentioned factors. Microgestures are powerful but sensitive - learning a new microgestural user interface may compare to learning a new instrument that requires expert dexterity and time to master. In accordance with efficiency, we saw that well-designed applications yield good empirical data concerning fit-to-function and effort through the Tidmarsh case study. We believe that focus selection, data visualization, and other on-the-go applications would benefit greatly from microgesture, given the benefits microgesture has over other forms of input.

We believe that future mobile system designers should explore the methods presented in this manuscript and find value in studying the potentials of microgestural applications. The previously mentioned application areas are only a few of a myriad of possibilities. Given the rise of wearable technology and ubiquitous use of sensors and applicable interfaces, users will need an ubiquitous mode of interaction that is efficient, unencumbering, and simple. We believe that a universal microgesture language may help develop strong usability in applications and can stem from the microgesture applications in this manuscript. The universal microgesture language should also take into account the possibility of contrast bias affecting microgestural user experience. However, we believe that gestural customization is also very important to yield strong usability in microgestural applications and mitigate longer training periods (users can choose microgestures they are best at, yielding lower error rates and less required training eg. see chapter 5). Hence, we believe applications should make suggestions as to which gestures should represent various actions, but leave the choice to the user.

Bibliography

- [1] Gorilla Arm. <http://www.kickerstudio.com/blog/2009/02/gorilla-arm/>.
- [2] Tidmarsh Project by Responsive Environments. <http://tidmarsh.media.mit.edu>.
- [3] Unity Game Engine Website. <http://unity3d.com>.
- [4] Brian Amento, Will Hill, and Loren Terveen. The Sound of One Hand: A Wrist-mounted Bio-acoustic Finger Gesture Interface. *CHI*, 2002.
- [5] Shai Avidan and Ariel Shamir. Seam Carving for Content-aware Image Resizing. *ACM Trans. Graph.*, 26(3), July 2007.
- [6] Rachel Bainbridge and Joseph Paradiso. Wireless Hand Gesture Capture Through Wearable Passive Tag Sensing. In *International Conference on Body Sensor Networks*, pages 200–204, 2011.
- [7] Steve Benford, Holger Schnadelbach, Boriana Koleva, Rob Anastasi, Chris Greenhalgh, Tom Rodden, Jonathan Green, Ahmed Ghali, and Tony Pridmore. Expected, Sensed, and Desired: A Framework for Designing Sensing-Based Interaction. *ACM Transactions on Computer-Human Interaction*, 12(1):3–30, March 2005.
- [8] Baptiste Caramiaux and Atau Tanaka. Machine Learning of Musical Gestures. *NIME '13*, 2013.
- [9] Liwei Chan, Rong-Hao Liang, Ming-Chang Tsai, Kai-Yin Cheng, and Chao-Huai Su. FingerPad: Private and Subtle Interaction Using Fingertips. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 255–260, New York, NY, USA, 2013. ACM.
- [10] S. Sidney Fels and Geoffrey E. Hinton. Glove-Talk: A Neural Network Interface Between a Data-Glove. *IEEE Transactions on Neural Networks*, 3(6), 1992.
- [11] Paul M. Fitts. The Information Capacity of the Human Motor System In Controlling the Amplitude of Movement. *Journal of Experimental Psychology*, 47(6):381–391, June 1954.
- [12] Nicholas Gillian. Gesture Recognition Toolkit. <http://www.nickgillian.com/software/grt/>, June 2013.

- [13] Chris Harrison, Desney Tan, and Dan Morris. Skinput: Appropriating the Body As an Input Surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 453–462, New York, NY, USA, 2010. ACM.
- [14] Bruce Howard and Susie Howard. Lightglove: Wrist-Worn Virtual Typing and Pointing. pages 172–173, 2011.
- [15] Christine L. MacKenzie; Thea Iberall. *The Grasping Hand*. North-Holland, 1994.
- [16] David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. Digits: Freehand 3D Interactions Anywhere Using a Wrist-worn Gloveless Sensor. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pages 167–176, New York, NY, USA, 2012. ACM.
- [17] Jungsoo Kim, Jiasheng He, Kent Lyons, and Thad Starner. The Gesture Watch: A Wireless Contact-free Gesture based Wrist Interface. In *Wearable Computers, 2007 11th IEEE International Symposium on*, pages 15–22, 2007.
- [18] S.C. Lee, Bohao Li, and T. Starner. AirTouch: Synchronizing In-air Hand Gesture and On-body Tactile Feedback to Augment Mobile Gesture Interaction. In *Wearable Computers (ISWC), 2011 15th Annual International Symposium on*, pages 3–10, 2011.
- [19] Stephen E. Liebowitz, Stan; Margolis. The Fable of the Keys. *Journal of Law and Economics*, 33(1):1–26, 1990.
- [20] Christian Loclair, Sean Gustafson, and Patrick Baudisch. PinchWatch: A Wearable Device for One-Handed Microinteractions. *MobileHCI*, 2010.
- [21] Kent Lyons, Daniel Plaisted, and Thad Starner. Expert Chording Text Entry on the Twiddler One-Handed Keyboard. *ISWC Proceedings of the Eighth International Symposium of Wearable Computers*, pages 94–101, 2004.
- [22] David Merrill and Joseph Paradiso. Personalization, Expressivity, and Learnability of an Implicit Mapping Strategy for Physical Interfaces. *CHI 2005 Conference on Human Factors in Computing Systems*, pages 2152–2161, April 2005.
- [23] Pranav Mistry and Patricia Maes. SixthSense - A Wearable Gestural Interface. Yokohama, Japan, 2009. SIGGRAPH Asia 2009.
- [24] Gordon B Moskowitz. *Social Cognition: Understanding Self and Others*. Guilford Press, 2005.
- [25] Michael Nielsen, Moritz Störring, Thomas B. Moeslund, and Erik Granum. A Procedure for Developing Intuitive and Ergonomic Gesture Interfaces for HCI. *International Gesture Workshop 2003*, pages 409–420, 2004.

- [26] Vitor Pamplona, Leandro Fernandes, João Prauchner, Luciana Nedel, and Manuel Oliveira. The Image-Based Data Glove. *Revista de Informática Teórica e Aplicada*, 15(3):75–94, 2008.
- [27] Jun Rekimoto. GestureWrist and GesturePad: Unobtrusive Wearable Interaction Devices. In *Proceedings of the 5th IEEE International Symposium on Wearable Computers*, ISWC '01, pages 21–, Washington, DC, USA, 2001. IEEE Computer Society.
- [28] Jeff Sauro and James R. Lewis. Average Task Times in Usability Tests: What to Report? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2347–2350, New York, NY, USA, 2010. ACM.
- [29] Tom Simonite. Depth-Sensing Cameras Head to Mobile Devices. <http://www.technologyreview.com/news/519546/depth-sensing-cameras-head-to-mobile-devices/>.
- [30] Thad Starner, Joshua Weaver, and Alex Pentland. Real-time American Sign Language Recognition Using Desk and Wearable Computer Based Video. *IEEE Transactions on pattern analysis and machine intelligence*, 20(12):1371–1375, 1998.
- [31] Helman I. Stern, Juan P. Wachs, and Yael Edan. Designing Hand Gesture Vocabularies for Natural Interaction by Combining Psycho-physiological and Recognition Factors. *International Journal of Semantic Computing*, 02(01):137–160, 2008.
- [32] A. Vardy, J. Robinson, and Li-Te Cheng. The WristCam as Input Device. In *Wearable Computers, 1999. Digest of Papers. The Third International Symposium on*, pages 199–202, 1999.
- [33] Katrin Wolf, Anja Naumann, Michael Rohs, and Jörg Müller. A Taxonomy of Microinteractions: Defining Microgestures Based on Ergonomic and Scenario-Dependent Requirements. *International Federation for Information Processing*, 2011.