Touch-Enhanced Gesture Control Scheme

A Thesis

Submitted to the Faculty

of

Drexel University

by

Chelsea M. Myers

in partial fulfillment of the

requirements for the degree

of

Master of Science in Digital Media

June 2016

Acknowledgments

I want to thank Dr. Jichen Zhu for advising this project and her excellent advice, insights, and guidance through it. My committee members, Troy Finamore and Dr. Andrea Forte, for their help and valuable knowledge. Thalmic Labs for their technical support in development for the Myo armband. And the 2016 Drexel Digital Media Graduates for their moral support.

Table of Contents

List of Tables

List of Figures

Abstract

Touch-Enhanced Gesture Control Scheme

Chelsea M. Myers
Advisor: Dr. Jichen Zhu, Ph.D.

We present an approach for improving gesture control by combining it with touch input to address a key shortcoming of gesture — live mic syndrome — by using touchscreen commands as a virtual clutch. The touch-enhanced gesture control scheme is designed and developed using a generic smartphone. For performance evaluation, this scheme was compared to the commercially available Myo armband device. Two tasks designed to measure selection accuracy and speed in a within-subject user study (n=30) reveal our touch-enhanced control scheme is faster and more accurate when executing selection commands. Additionally, qualitative results from a post-study questionnaire showed a majority of participants selected the touch-enhanced as easier to use over the Myo.

1. INTRODUCTION

Gesture control continues to grow with a wave of new gesture input devices like the Myo, Leap Motion, Gest, Ring Zero, and Reemo. However, further development of gesture control is currently hindered by a major drawback: Live Mic Syndrome [30]. This issue impacts a gesture system's ability to properly read the user's intent and decrease its execution accuracy and speed.

We designed and evaluated an approach to mitigating the drawbacks of gesture control and improving its ability to read intent by combining gesture and touch. In particular, we hypothesized that combining touch and gesture will increase users' precision and performance speed compared to wearable in-air gesture alone. Our goal is to not replace modern gesture control devices, but instead offer a cheaper and ubiquitous solution by utilizing an unmodified smartphone as a gesture control input. To test our hypothesis, we designed and developed a prototype touch-enhanced gesture control scheme that uses the motion tracking capabilities of a generic smartphone with its touchscreen to create a gesture input device. In the rest of the paper, we discuss related work, present our prototype, review our user study design to evaluate our approach, and analyze the gathered results.

Since the earliest research on gesture control systems [5], significant advancements have been made in both hardware and software. However, even with the many commercially available products for gesture control on the market today, gesture control still has its shortcomings that prevent it from wider adoptions. One major shortcoming of gesture control has been identified as *live mic syndrome* [27, 30], also referred to as "immersion syndrome".

Live mic syndrome, comes from the fact that the actions of a person using gesture control are always being recorded. While using a gesture control device, any user actions, whether intentional or not, can be misinterpreted as a command. For example, if the user scratches her head, the gesture device may interpret it as a command and hence produce a false-positive error. In other words, the core issue here is intent — how to differentiate intended actions from the other ones.

The Myo armband (Figure 1.1), the representative wearable gesture control system we choose for our project, suffers from live mic syndrome. It was designed to be worn on the forearm of the user and uses surface electromyographic (SEMG) signals in the user's arm to detect movement and hand poses. The Myo is an *in-air* gesture control device, meaning commands are made in-air without touching anything (like making a fist to select). Like some other gesture control devices, Myo attempts to solve live mic syndrome by having a *virtual clutch*.

A virtual clutch is a reserved command that toggles whether or not a gesture device is executing commands. The command must be outside of common gestures preformed naturally. Although it may reduce the amount of false-positive errors, virtual clutches come with their own problems. A *false-negative error* happens when a user gives a command, but the gesture device fails to read it [30]. This type of error can be caused by the actual device failing to read the command, or the software failing to interpret it. This impacts the reading and execution of the virtual clutch command. If gesture devices need users to repeat the command before recognition, this increases the time it tasks to toggle on and off tracking and the time it takes to complete tasks.

Currently, gesture control devices are still not as accurate in reading commands as touchscreen interaction. Therefore, combining the precision of touch and the directional manipulation of gesture offers a promising direction to improve the latter. Our proposed system does not use an in-air gesture as its virtual clutch and instead uses the touchscreen to toggle motion tracking. Our touch-enhanced control scheme aims

Credit: Thalmic Labs

Figure 1.1: The Myo Armband.

to make executing the virtual clutch easier than it's wearable in-air gesture counterpart. By achieving this, gesture control users would be able to start and stop control more easily and reduce false-positive errors.

In terms of false-negative errors, it is fairly common that a user may have to perform a command a few times before the device reads it as such. Redg Snodgrass, CEO and cofounder of Wearable World says, "Even though we are very much the same, we still have our own individual quirks. That makes it very hard to interpret what intent is...Gesture is all about intent." [23] Gesture control devices have to account for the variation among their users to be accessible. As Snodgrass said, individual quirks means that for some users, a command may happen naturally for them but be more difficult for others. By moving the virtual clutch from an in-air gesture to the touch screen in our system, we saw a reduction false-negative errors since user variation does not impact touch interaction as heavily.

1.1 Research Question

Does combining touch with gesture control improve gesture's main shortcoming; live mic syndrome?

1.2 Terms

Interaction Commands: A touch or gesture, or a combination of touches or gestures, to complete an action on a digital device.

Touch: An interaction with a device by physically touching a screen.

Gesture: An interaction with a device by using in-air bodily movements (e.g. Moving of arms).

In-Air Gesture Command: Commands made in-air without touching a device (e.g. making a fist to select).

2. Background

Newer interaction inputs are being incorporated into consumer products. Interaction methods like gesture, voice control and eye tracking are becoming more prevalent [2]. The Samsung Galaxy 4, a popular smartphone, has basic eye tracking and gesture tracking to control video and page scrolling. Voice control systems like Apple's Siri and Mircosoft's Cortana are becoming established and more accurate. Commercial and consumer ready eye-tracking systems are available as well as products like the $99 Eye Tribe. There is no longer a standard for what size or shape our digital devices may be. The way we interact with these devices and sensors will vary as well. Not every device and interaction method will be a match. Some devices in our future may be better suited for touch interaction, rather than gesture control, while others may need both.

For this study, the definition of gesture is in-air bodily movements controlling an interface or object. Other studies use the term gesture to refer to complex touch screen interactions. For this research we will not be using the term this way. Gesture control converts bodily movements into directional manipulation or commands. Gesture control can track the user's whole body or a specific body part. What part of the body a gesture device tracks varies from device to device.

Gesture control is not new. One of the earliest gesture control experiments was created in 1993 [5]. A gesture control project called Charade [5] tracked one user's hand by having them wear a data glove. Charade's commands revolved around controlling presentations. Baudel et al. was also aware of live mic syndrome, but referred to it as "immersion syndrome." Live mic caused the user to be "cut off from the possibility of communicating simultaneously with other devices or people." [5] Baudel's hypothesized solution to this was to increase gesture's hardware and software sophistication.

But this cannot be done without considering the impact on cost. John Underkoffler's gesture scheme are some of the most complex commercial systems. Underkoffler latest project was the Mezzanine (`http://www.oblong.com/mezzanine/benefits/`). Mezzanine is an expensive gesture controlled digital workspace. In September 2014, Underkoffler filed a patent [25] to protect a “gestural vocabulary” he developed. In this patent Underkoffler and Kevin Parent present a two handed gesture scheme with fine controls using finger poses. The scheme is “encoded” and needs a key to read. Most of the proposed gestures need one arm/hand to execute. The commands focus on manipulating media and organizing a digital workspace. Like the Mezzanine, most of Underkoffler's gesture schemes are not available to the public or come with high price tags.

To help lessen live mic syndrome, Baudel et al. implemented an “active zone” for Charade where the user could point to input a command. Even with this active zone, Baudel et al. found that the user's intent was still being misread and the system would sometimes execute a command when none where given. And with new gesture control technology that is more sophisticated than the data glove, we still see live mic as an issue. With Charade, Baudel et al. aimed to explore gesture control schemes and create a device that could interpret the commands as accurately as possible.

Our gesture control devices still suffer from live mic syndrome. Chudgar et al. [9] in 2014 built a gesture control scheme for the Samsung Galaxy Gear smartwatch. Chudgar et al. developed a system based on the device's accelerometer recognizing six gestures. Different commands could be mapped to five of them. Most of the designed gestures were directional; up, down, left, right. The last two gestures were a clockwise movement and drawing a “S” shape in the air.

The team admits issues with their system though. “While performing the 'Down' gesture, the hand first moves upwards to position itself and then downwards. The first upward positioning movement in some cases can be erroneously detected as 'Up'." [9] This issue is a symptom of live mic syndrome. A user wearing a smartwatch with her arm at rest must first lift her arm before moving it back down to signal the "Down" command. Since the smartwatch is constantly tracking movement, both the "Up" and "Down" commands execute. By exploring more options for reducing live mic syndrome, we can look for a better solution.

Affordable gesture control devices available to the public have begun to appear. Smartwatches have basic gesture control. Some type of gesture control is available on all the modern console (e.g. PlayStation 4, XBox One). Even some cars have gesture integration. The 2016 BMW 7 Series comes installed with a gesture tracking dashboard. The dashboard allows drivers to control volume, accept and deny calls, and view the car's perimeter with gesture commands [12]. On top of this, as seen in Figure 2.1, gesture control specific devices are available like the Myo and Leap Motion. Some of these devices have a specific interface to control. Others are an input device that can power a multitude of interfaces and objects.

The devices seen in Table 2.1 are of all different shapes and sizes. *Wearable* gesture control devices can be, as the name suggests, worn by the user. They can be worn on the forearm, wrist, fingers; potentially anywhere to track movement and commands. *Vision-based* devices like the Kinect and Leap Motion are stationary cameras that track the user when in view. For this study, we will be focusing wearable gesture control that tracks the movement of the user's dominant arm. Our touch-enhanced control scheme, like wearable gesture control, is mobile and can be carried with the user. We argue that since our control scheme uses a device people probably already have on hand, it is more accessible.

Gesture Device	**Price**	**Tracks**	**URL**
Leap Motion	$80	Hands/Arms (Partial)	`https://www.leapmotion.com/`
Gest	$199	Hands	`https://gest.co/`
Ring Zero	$149.99	Index finger	`http://logbar.jp/ring/en/`
Nod	$149.99	One Hand	`https://nod.com/`
Reemo	$250	One Arm	`http://www.getreemo.com/`
Myo	$250	One Arm/Hand	`https://www.myo.com/`
Kinect One	$125	Whole Body	`http://www.xbox.com/en-US/xbox-one/accessories/kinect-for-xbox-one`

Table 2.1: Modern gesture control devices

2.1 Gesture Commands

Through our research, we have found gesture commands split into two broad categories: *posed commands* and *directional commands*.

Posed commands are signals produced by the user to execute a command. These signals are like sign language. A hand pose in sign language correlates to a word, letter, or phrase. Posed gesture commands are like hand signals. A gesture tracking device reads these poses and assigns them to a command. For example, to control a presentation with the Myo, a user would wave to the right with one hand to advance a slide. Once the Myo detects this right wave, it will trigger the presentation to move one slide forward.

Directional commands are gestures in one or multiple directions that perform manipulations (e.g zooming, scaling). They are less detailed than posed commands. Alcoverro et al. [1] uses the Kinect to control panoramic displays: stills and video. User's can pan with-in a panoramic image by using a grab-release method. The “grab” is a posed command, but the motion after is a directional command. The directional command indicates the amount of change that needs to occur. Commands like lowering a hand to dim lights, or raising one to increase volume, are directional commands. Unlike posed commands, most directional commands execute in real time while the command is occurring.

Gesture schemes today are usually made up of both types of commands. Simpler schemes focus on directional commands. Acceleration and direction are easier to detect than hand poses. The touch-enhanced gesture scheme uses only directional gesture commands combined with touchscreen commands. In this scheme, touchscreen interaction will take the place of posed-commands.

We compared our scheme to the Myo armband. As mentioned, the Myo is a wearable gesture device that tracks electrical signals in the wearer’s arm to detect directional and posed commands. We have selected the Myo as the comparative gesture control device because it is one of the few available devices that has a virtual clutch. To toggle on and off listening of the Myo, the wearer must double-tap her thumb and index finger. Also it’s price, at $250 US dollars, makes this device more accessible to the public.

The Myo has five pre-programmed commands: making a fist, waving left, waving right, finger spread, and double-tapping the thumb and middle finger. It is advertised as a presentation controller with potential as a game input. The Myo is capable of controlling drones and robotic prosthetics, to powering a mouse on a computer. For this research, the Myo and touch-enhanced control scheme will power a cursor on an project display.

2.2 Related Work

The purpose of this research is to examine the combination of gesture and touch in effort to design a more accurate and faster control scheme, particularly through the use of smartphones. Projects researching gesture and/or touchscreen interaction explore these interaction types either controlling the input device itself or another standalone interface. Our project explored the latter.

Research continues to improve gesture control while developing different ways to detect commands. Lu et al. [16] designed another SEMG device, like the Myo, that is worn on the user's forearm. The SEMG device and algorithmic framework achieved a 95.0% accuracy with 20 participants when testing 19 of their designed gestures. This was only achievable after the participants trained the gesture classifier by preforming the commands. When the participants did not train, the accuracy rating dropped down to 89.6%. Lu et al. however did not design a virtual clutch or address live mic syndrome. Rekimoto designed the GestureWrist and GesturePad in 2001 [18]. The GestureWrist is a watch that detects gestures by shape changes in the wrist. GesturePad is a way to detect gestures preformed over a user's apparel. Both of these devices do not have a designed virtual clutch as well. While gesture control is still being advanced through research like this, the touch-enhanced gesture control scheme offers an accessible solution, with no algorithmic training needed, by using a device users may already have on hand.

2.2.1 Multimodal Interaction

Another touch-enhanced control scheme can be found in a project called Code Space developed by Bragdon et al. [7]. Code Space was an intelligent room for developers to share and discuss their work. Code Space consisted of a projected display manipulated by gestures tracked from Kinects installed in the room and multiple touchscreen devices. Users could point at the display and use in-air gestures to drag and drop code snippets.

Users could also interact with the display by holding a touchscreen smartphone like a remote and pointing it. Unlike our touch-enhanced control scheme though, motion tracking was not done by the smartphone. Instead, Code Space used the installed Kinects and skeletal tracking to determine the position and movement of the smartphone. Meaning, the touch-enhanced system by Bragdon et al. is not mobile and is tethered to the Code Space room.

Bragdon et al. did also use the touchscreen of the smartphone to receive commands. Users could touch the smartphone's screen to begin dragging code snippets on the projected display. They could also swipe horizontally on the screen to pan/scroll the display. Also, users could share information from their smartphone by pointing at the display and swiping up, or take content from pointing and swiping down.

This utilization of a touchscreen functionality is a great representation of what a touch-enhanced control scheme can do. Although Bragdon et al. do not discuss live mic syndrome, they purposely removed in-air gesture from their control scheme and replaced them with touchscreen commands to reduce false-positive errors. Bragdon et al. also did not use a command as a virtual clutch. Code Space's approach to address live mic syndrome is similar to Charade [5]; users point at the display for the Kinect to begin tracking the smartphone's position. For the touch-enhanced control scheme, users do not have to hold up their arms and point to start interaction. Observations during our study showed that when participants raised their arm for a long period of time, some experienced fatigue. Instead, they can rest their arms on a table or in anywhere that is comfortable to them. Also, by relying on the Kinect for motion tracking, the amount of users that can interact with a display at once is limited to six. These users must be position in front of the Kinects. Our touch-enhanced system would allow for an unlimited amount of participants with no seating requirements needed.

Code Space's formative study did not evaluate the performance of their system. Instead, evaluations focused on the social acceptability of using Code Space for developer meetings and yielded only qualitative results. Overall, participants responded positively to Code Space which tells us touch-enhanced systems would be socially acceptable to use in public. Major concerns only arose regarding peer-to-peer sharing caused by the social awkwardness of pointing at another participant to share files.

Other existing projects take advantage of a smartphone's motion data and use it as a gesture control device. To our knowledge, little research has been done exploring the combination of touch and wearable in-air gesture control in one system to lessen the shortcomings of gesture control. Bragdon et al. [7] and Code Space combined touch with a vision-based gesture control device. Ljubic et al. [15] used gesture to enhance touch interaction by implementing a smartphone's tilt to zoom in on its keyboard for easier text entry. Song et al. [22] extended interaction for smartphones by using the built-in camera and hardware to recognize in-air gestures. By creating a random forest based gesture recognizer, they turned a smartphone into a vision-based gesture device. With this, Song et al. [22] were researching solutions to solve shortcomings of touchscreen, like screen real estate, by combining it with in-air gestures.

Bossuyt et al. [6] combined touch and gesture to create an interactive public display. Bossuyt et al. observed how combining touch and gesture impacts social behavior, like Bragdon et al. [7], and interaction patterns of people walking by the display. The emphasis of their research sought to enhance touch with in-air gestures, whereas our research focuses on improving gesture with touch interaction. Bossuyt et al. was aware of live mic syndrome during their work but actively chose not to address it.

2.2.2 Controlling Input Device

When using smartphones for gesture control, research primarily focuses on using gestures to control the smartphone itself, and not another interface. Weberg et al. [28] proposed to control a PDA's cursor with tilting. Weberg proposed cursor control should be like moving a "pad of butter on a hot pan" [28]. Many projects focus on using gesture as text entry for smartphones [11, 10, 15, 29] and navigation [28, 8, 3].

Ruiz et al. [19] designed a *guessability study* to define best-practices for gestures that "invoke commands" on a smartphone. Ruiz et al. prompted participants to design gestures for 19 commands acting upon the smartphone or navigating within an application. Ruiz et al. broke down the nature of their user-designed gestures into four categories: metaphor, physical, symbolic, and abstract. The motion gestures in our system are physical, and directly manipulating content on a separate display.

2.2.3 Controlling Standalone Interface

Just Dance Now is an online game that allows players to use their smartphone as a *Just Dance* controller. All the player needs is an "internet connected" screen for displaying the game and a smartphone [14]. However, projects like *Just Dance Now* do not utilize the touch screen to address the shortcomings of gesture control. The only action a user can preform on the touchscreen while the device is being used as a gesture input is quitting the current game. In contrast, our touch-enhanced gesture control scheme uses a smartphone's touch screen as an area for users to touch to execute commands *while* using the smartphone as a controller to track the user's movement.

The Wii remote has a similar set up using gesture control with physical buttons that also input commands. Jones et al. [13] used the Wii remote for input text on a custom interface to measure accuracy and speed. Other work studied using gesture to control smart television to control media browsing. Bailly et al. [4] used the Wii remote to propose a method of gesture interaction for media consumption.

Using the Wii remote's buttons to address live mic is not being researched though. Schlomer et al. [20] designed a Wii remote system allowing for training recognition of gestures. Participants in this study would "train" the system 75 different commands. While inputing these commands, participants held down the "A" button. Holding down "A" acts as a virtual clutch in this system, but Schlomer et al. did not study the effects this virtual clutch had against an in-air virtual clutch command. We have chosen not to use the Wii remote, like other gesture control studies [7], due to the limitations of using a specified controlling that users would need to carry around. Also, touchscreen allows us to make more customized interfaces.

3. Project Design

In this study, touch-enhanced gesture control scheme using a smartphone was compared to a wearable in-air gesture device called the Myo. The 30 participants in this study used both schemes, in random order, and execute the same two task sets with each: the selection tasks and the movement tasks. These tasks analyzed both device's selection accuracy and speed by gathering usage data about both devices to observe any differences in speed and accuracy.

While our goal was to find participants of equal familiarity with touch and gesture, with the ubiquity of smartphones, finding participants with either little touchscreen exposure or even more gesture control is difficult. In 2015, it was reported that 64% of adult americans owned a smartphone [21]. To begin the study, first participants completed a pre-study questionnaire to gather their backgrounds and familiarity with gesture control. Here we asked participants about their previous exposure to gesture control and their daily smartphone usage. At the conclusion of the study, all but one participants used a gesture control device of some kind, like the Kinect of Wii remote. Three reported using the Myo armband before. Also, as expected, all participants had a smartphone and used it daily.

After the pre-study questionnaire, participants were then walked through both the selection and movement tasks. Participants were told that they could take a break, or skip a task, at any time. After the task explanations, participants stood 12ft away from a project display with their first device to use. This gave them room to move their arm and body around without risk of bumping into any tables or screens.

Next, participants were given a tutorial on how to use the device. They were showed how to use the virtual clutch of the device, how to select or "click" with it, and also how movement worked. Participants had time to get aquatinted with each device by practicing these commands and then using the devices to click targets on the display. This allowed the participants to become familiar with each devices' sensitivity and commands. When the participants said they were comfortable, and a minimum of 2 minutes had passed, they would next complete the selection and movement tasks with the device. For all tasks, participants used the virtual clutch to start motion tracking and begin the task. Sound effects were added to give feedback when motion tracking was toggled on for both devices. Also, the color of the cursor changed with the state of motion tracking.

After, the participants were given the next device and its tutorial. The participants again completed the same tasks with it. Finally, each participant completed a post-study questionnaire to record if the participants perceived any difference between the devices in terms of selection accuracy, speed, and ease of use.

3.1 The Myo Armband

The Myo armband was selected as the control for this study since its system incorporated and desgined a virtual clutch. If the user double-taps her thumb and middle finger while wearing the Myo, the device is toggled on and off. The Myo armband is mobile, and can travel with the user unlike vision-based gesture devices. It also tracks only one arm of the user, limiting the variables to design for, unlike full-body gesture control devices like the Kinect. And finally, the Myo allows for cross-device, operating system, and programming language development, making it accessible to develop for.

3.1.1 Using the Myo

In our study, participants wore the Myo on their dominant arm right below the elbow. As mentioned, the virtual clutch for the Myo is double-tapping of the thumb and middle finger. Each participant used the virtual clutch to start each task. To "click", or select, with the Myo in the display, the participant made a fist while motion tracking was enabled.

3.2 The Touch-Enhanced Gesture Control Scheme

For the touch-enhanced gesture control scheme, a generic unmodified smartphone was selected to be the input device. Requiring extra equipment can act as a barrier for adoption. Our touch-enhanced control scheme does not use any outside gesture control devices. Like the Myo, a smartphone can be manipulated with one hand/arm. Since it is important to match the Myo's use with our touch-enhanced control scheme, we selected a smartphone over a tablet because smartphones are easy to hold in one hand. Also, like the Myo, smartphones track gyroscopic and acceleration data. Both the Myo's and the touch-enhanced control schemes were built using this data.

Just like the Myo, participants using the smartphone must use it's virtual clutch to start the task. The smartphone's virtual clutch is swiping vertically on the screen. To start motion tracking, participants swiped up on the screen. To stop, participants swiped down. To "click" with the smartphone, participants tapped anywhere on the screen while motion tracking was enabled.

3.2.1 Designing the Touch-Enhanced Gesture Control Scheme

The touch-enhanced control scheme uses a smartphone and was designed to be held like a remote as shown in Figure 3.1. The user can move the smartphone in-air to control a cursor on a projected display. This display is actually a webpage using WebSockets and Node.js to transfer motion data from the smartphone to the display. The touchscreen of the smartphone in our system is designed to directly address the key shortcomings of gesture control and allows it to be more customizable than the

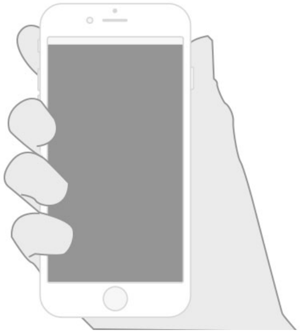

Figure 3.1: Example of holding smartphone as remote.

Myo.

To reduce false-positive errors caused by *live mic syndrome*, the user can toggle whether or not the smartphone is tracking her motion. Instead of a gestural in-air command like the Myo has, our system uses a vertical swipe to toggle motion tracking. Figures 3.2 and 3.3 show a prototype of our system. By default, no motion is being tracked until the user toggles the smartphone to begin listening. To do this, the user is prompted to swipe up with their thumb along the screen. As the user does this, the screen shown in Figure 3.3 appears. Once the active screen is shown, the smartphone begins tracking the user's movements. The active screen is intentionally green to allow the user to easy glance at the smartphone in hand to see if motion tracking has been toggled. Audio feedback also indicate tracking has been turned on. To turn off motion tracking, the user can swipe down on the screen. The inactive screen from Figure 3.3 will then appear. The user can then swipe up at anytime to being tracking again. When the active screen is shown, users can then tap anywhere on the smartphone to "click" with their cursor.

Figure 3.2: Inactive Screen.

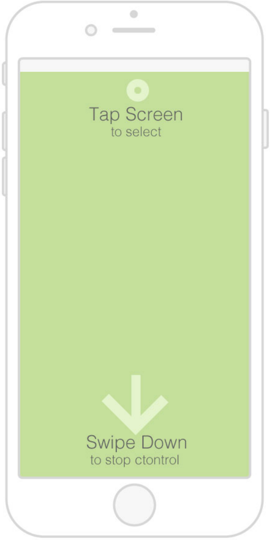

Figure 3.3: Active Screen.

3.3 The Tasks: Selection & Movement

The selection and movement tasks in this study observe differences in selection accuracy and speed. When the participants are preforming these tasks, they are being timed. Timer starts when the participant is presented with the task. Participants must use the virtual clutch of both devices to start motion tracking and move the cursor. The time of each task will also include the time it takes to do this. The timer stops when the task is completed.

For feedback, colors and sounds are used to alert the participants of changes. While the cursor on the screen is usually blue, it changes to green when the virtual clutch is executed and the devices are toggled off. An audio cue also plays when motion tracking is toggled on. When the participants "click", a "popping" noise plays and a navy dot is added at the coordinates of the "click" on the screen.

3.3.1 Selection Task Design

The selection task is based off of the Parhi et al. [17] study analyzing the accuracy of controlling a smartphone with one thumb. In [17], tasks presents the participant with nine boxes in an equally disbursed 3x3 grid on a screen that she must tap. In our study, these 9 boxes were shown on the projected display. Participants used the Myo or smartphone to select each box.

Like [17], our tasks consisted of five screens, with varying target sizes: 100 pixels (px), 80px, 60px, 40px, and 20px. Where and when the participants "click" the target were both recorded. Apple recommends hit states to be at least 44x44px to make is easier for users to tap [24]. Our range included targets both larger and smaller to help identify the ideal hit state size for both wearable gesture control devices and our touch-enhanced system. For these targets, how many attempts it took the participant to select each target was also recorded. All data was stored in a JSON file at the end

Figure 3.4: An example of a wire loop game

of the tasks.

3.3.2 Movement Task Design

The movement tasks were based off of the wire loop game, seen in Figure 3.4. In this game, players must pass a hook or "loop" around a bent wire without touching it. Our movement tasks' goal was to measure a participant's success rate for executing the device's virtual clutch while moving. Participants had to move through the same mazes and use the virtual clutch at the same points to ensure consistently between participants and devices. Like the wire loop game, participants had to guide their cursor, controlled by either the Myo or smartphone, without touching the edges of the maze. If they did, an error sound was played, and the participant had to start at the beginning of the maze. There were a total of three mazes to complete, increasing in difficultly.

4. Methodology

This within-subject user study was designed to evaluate our touch-enhanced gesture control scheme in terms of accuracy and speed through a controlled experiment. It consisted of the following procedures. Users first were questioned about their background and knowledge of gesture control schemes. Next, they would preform the two task sets designed to measure the selection accuracy. Finally, participants filled out a follow-up questionnaire reflecting on the tasks performed.

4.0.1 Subjects

A total of 30 participants were recruited between the ages of 18-25 from the Digital Media undergraduate programs at Drexel University. Participants must have owned a smartphone for over one year and use it at least twice daily. The group consisted of 11 females and 19 males. All but two participants were right handed.

4.0.2 Existing Knowledge Questionnaire

The participants were first asked to complete a questionnaire seen in section 8.1.1 about what gesture control devices they have heard of and used in the past. The questionnaire also gathered background information, like age, gender, education, and occupation.

4.0.3 Controlled Experiment

Task 1: Selection Tasks. Participants were be asked to "click" on different sized targets. Much like Verma et al. and Parhi et al. [26, 17], this task had different pixel sized targets and observed what the threshold is for accuracy. Similar to Parhi et al., the participant was be presented with a screen of a 3x3 grid of targets all the

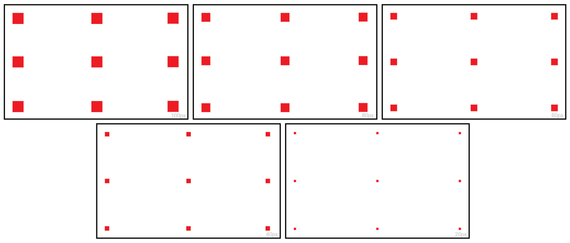

Figure 4.1: Selection Accuracy Tasks

same size. The participant was asked to "click" each one of these targets with the gesture control device while standing 12ft away from the display. Once completed, the participant would proceed to the next screen, almost identical to the previous one, except the targets were smaller. As shown in Figure 4.1, there were 5 different 1680x1050 resolution projected displays with targets of 100px, 80px, 60px, 40px, and 20px large. The amount of "clicks", their coordinates, and time to "click" all targets was recorded for each participant. Also, all participants we asked to click the targets in the same order; starting with the middle target, moving up one, and moving clockwise from there.

Task 2: Movement Tasks. Participants were asked to move a cursor on the display through a maze, as seen in Figure 4.2, without touching the sides of the maze. The cursor was controlled by the participant's movements through the gesture control devices. Each participant completed three different "levels". Each level was a new maze the participant had to navigate through.

These tasks measure how the two gesture systems' virtual clutches performed in motion with a fatiguing user. Within each maze, there are red dots, or virtual clutch points. Once the participant approached these points, she had to use the device virtual clutch to pause and start again. Maze 01 had one point, Maze 02 had two, and Maze 03 had three. The virtual clutch points were in the same locations for each device. Participants could take these points as a break or realign themselves as well.

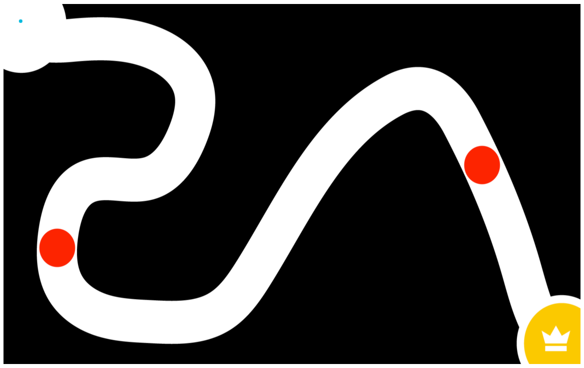

Figure 4.2: Movement Accuracy Tasks

These tasks recorded each participant's attempts to execute the virtual clutch. Participants counted out loud how many times it took them to preform the virtual clutch command until the system actually executed. Also, the number of attempts to complete the maze, and the time to complete each maze was recorded. Our goal was to to compare the affect of our swiping touchscreen virtual clutch and the in-air gesture virtual clutch of Myo.

4.0.4 Subjective Preference Questionnaire

The final part of the study was the Subjective Preference Questionnaire seen in section 8.1.2 using a 7-point Likert scale. This questionnaire asked the participants how satisfied they were using these devices with each task. Next, they were asked how they perceived the selection accuracy and speed of each device. Then, they were asked overall how satisfied they were with the ease of use for both devices. Finally, participants were asked which device of the two did they feel they had the least difficulty using, if any, and to explain their choice.

5. Results

Our system tracked and recorded data (e.g. number of commands executed and time to complete tasks) related to the two set of tasks previously described for 30 participants. This data was used to measure the accuracy and speed of each devices in our study. In order to test our hypothesis that false-positive errors from live mic will be reduced, we preformed analysis of variance (ANOVA) of the data as well as analyzed qualitative data gathered from observations during testing and from pre- and post-study questionnaires.

A total of two participants were excluded from the results since the Myo would not sync with them or recognize any of their gestures. It is unknown why the Myo would not recognize these participants. The Myo was tested during troubleshooting by the study proctor to ensure the device was not recognizing the participant and not just malfunctioning. These participants were able to finish the tasks with the touch-enhanced control scheme using the smartphone. Also, while all 30 included participants used the same smartphone with no customized calibration, four participants needed custom calibration profiles before the Myo would sync with them and recognize their gestures.

5.0.1 Selection Task Results

For the selection tasks, participants were presented with five 3x3 grids of same-sized targets and were instructed to “click” on each using either the smartphone or Myo. Only one grid was shown at a time. Each grid had a target sizes of 100px, 80px, 60px, 40px, and 20px.

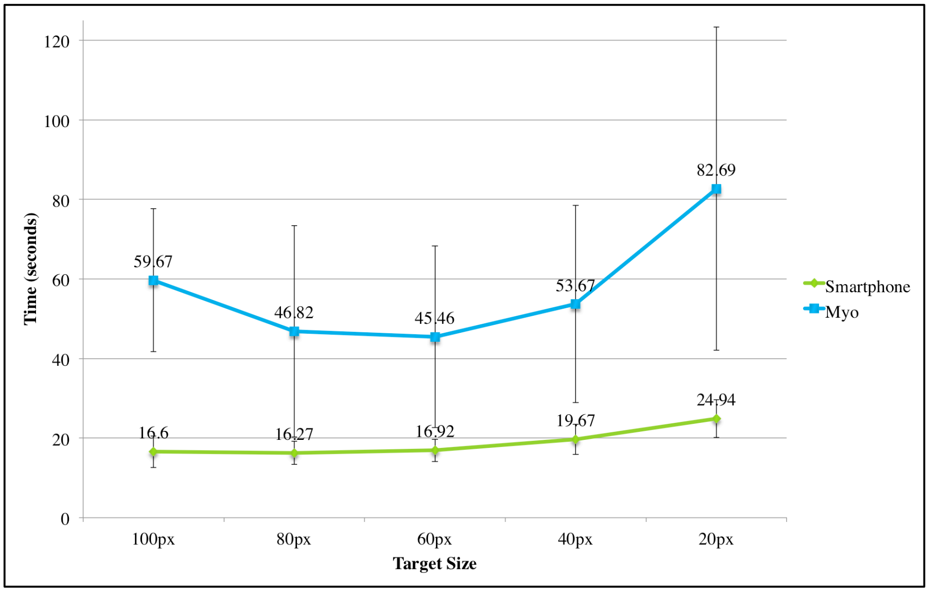

Figure 5.1: Selection Task Time Means and Standard Deviation

Selection Task Times

Task time was measured from the load of the grid, to the completion of clicking on each of the 9 targets. From grid load, participants first activated motion tracking by using the device's virtual clutch. The mean total time and standard deviation for each target size using both the smartphone and Myo were calculated. Results in Figure 5.1 show that participants on average completed all selection tasks quicker with the smartphone than with the Myo. Two-tailed paired t-tests (with a significance level of 0.05) conducted on both devices' data sets for all target sizes revealed statistical significance ($p < 0.0001$).

The standard deviation for these data sets also show the varying range of completion times for each target size. As seen in Figure 5.1, not only did participants using the smartphone on average perform the tasks quicker, the standard deviations are also smaller than the Myo. This closer distribution shows participants using the smartphone had more similar selection task completion times than the when using

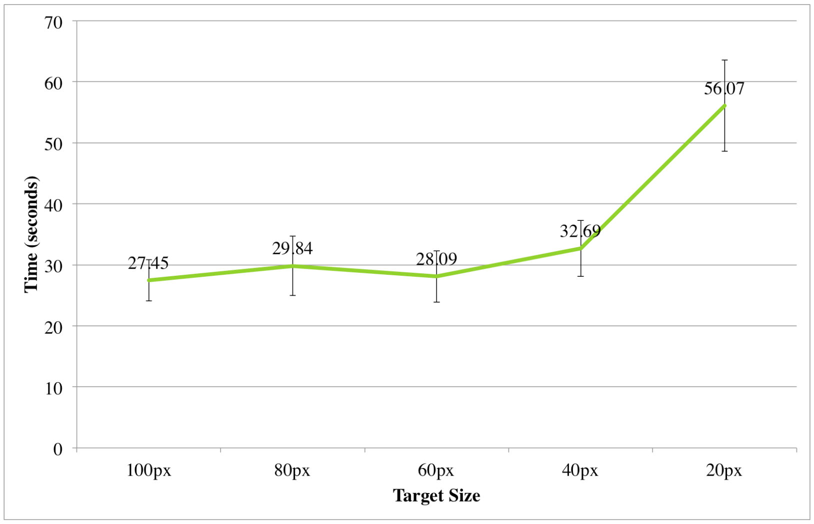

Figure 5.2: Selection Task Time Mean Difference and Standard Error of the Difference

the Myo. As represented in Figure 5.2, for target sizes 100px, 80px, 60px, 40px, and 20px, the mean smartphone selection task times decreased by 27.45s ± 3.36s, 29.84s ± 4.87s, 28.09s ± 4.20s, 32.69s ± 4.58s, 56.07s ± 7.46s respectively when compared to the Myo.

Once acquainted with the smartphone, mean task time decreased for the 80px target grid. From there, mean task time increased as the targets get smaller. It is inconclusive whether the Myo also follows this trend. While the 60px target size time mean was the lowest for the Myo at 45.46s, the difference between the means from 80px and 40px did not show statistical significance.

As Figure 5.1 shows, for both the smartphone and the Myo, the last and smallest target size of 20px took participants the longest to complete. Figure 5.2 shows that as the selection tasks increased in difficulty, the mean difference, in favor of the smartphone, also increased.

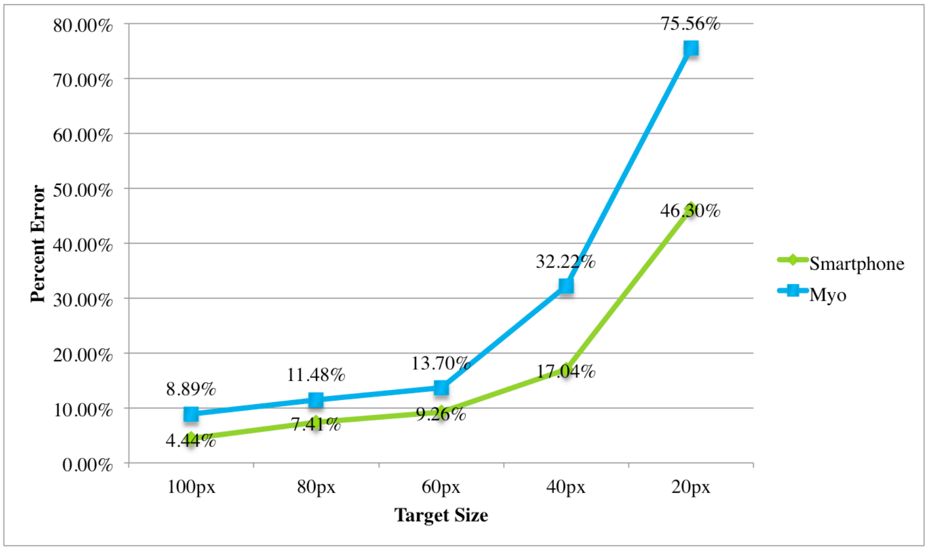

Figure 5.3: Selection Task Percent Error

Selection Task Percent Error

The amount of "clicks" recognized per participant was also recorded during the selection tasks. This recorded amount with a theoretical value of 9 "clicks" was used to calculate the mean percent error for each target size. Figure 5.3 shows that on average, participants using the smartphone yielded a lower percent error while "clicking" the target sizes. T-tests preformed on both devices' data sets for each target size reveal that this difference is only statistically significant for target sizes of 100px ($p < 0.05$), 40px ($p < 0.05$), and 20px ($p < 0.01$).

It must also be noted that amount of "clicks" recorded were only those recognized by the devices. Attempts to click by the participants were not recorded. It was observed during testing that both devices produced false-negative errors. It was, however, very clear during testing that this affected the Myo severely. This issue is captured by the Myo's selection task times. Although the selection task data showed no statistically significant results for clicking 80px and 60px target sizes with both devices, selection task times showed significant improvement for the smartphone. It was observed that the longer selection tasks times for the Myo were not caused by the speed of the hardware or software, but the device's failure to read the user's intent. For many of the targets, participants needed to execute the selection command, making a fist, over and over again until the device read it and executed the command.

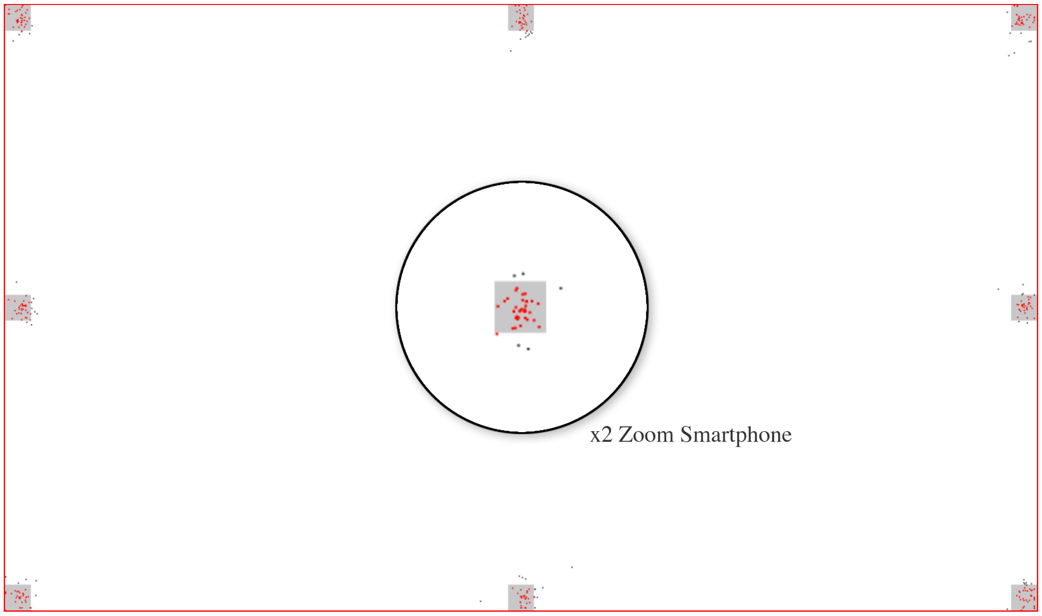

Figure 5.4: Selection Hit Distribution for Smartphone 40px

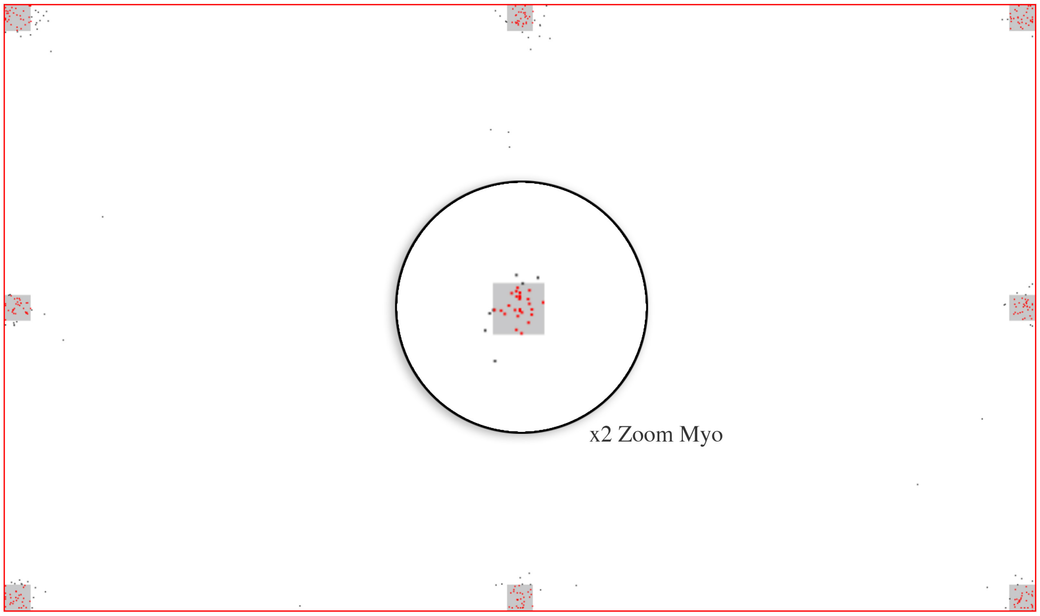

Figure 5.5: Selection Hit Distribution for Myo 40px

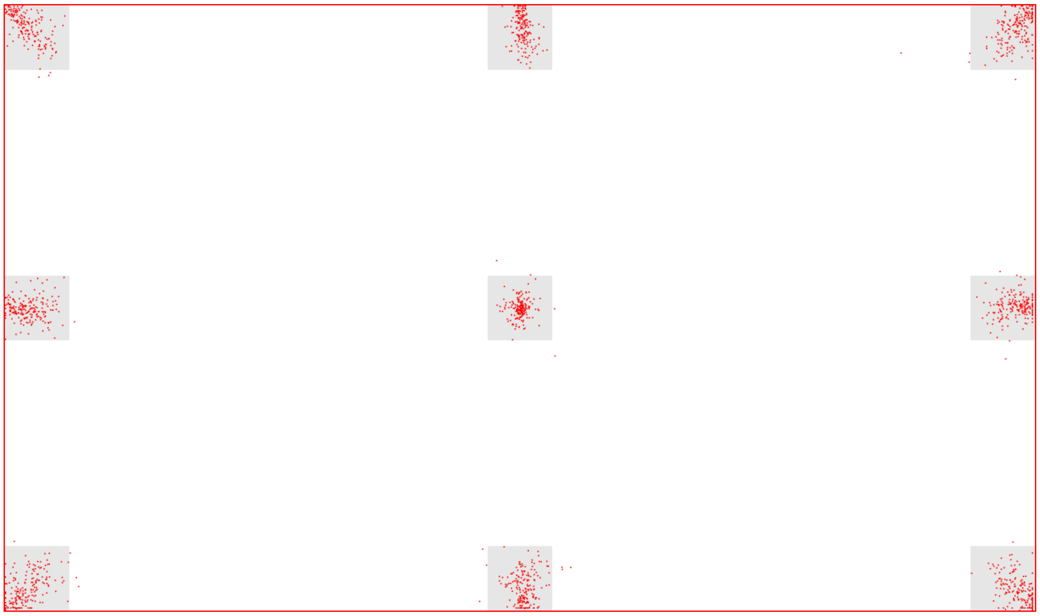

Figure 5.6: Selection Hit Distribution for Smartphone Across All Target Sizes

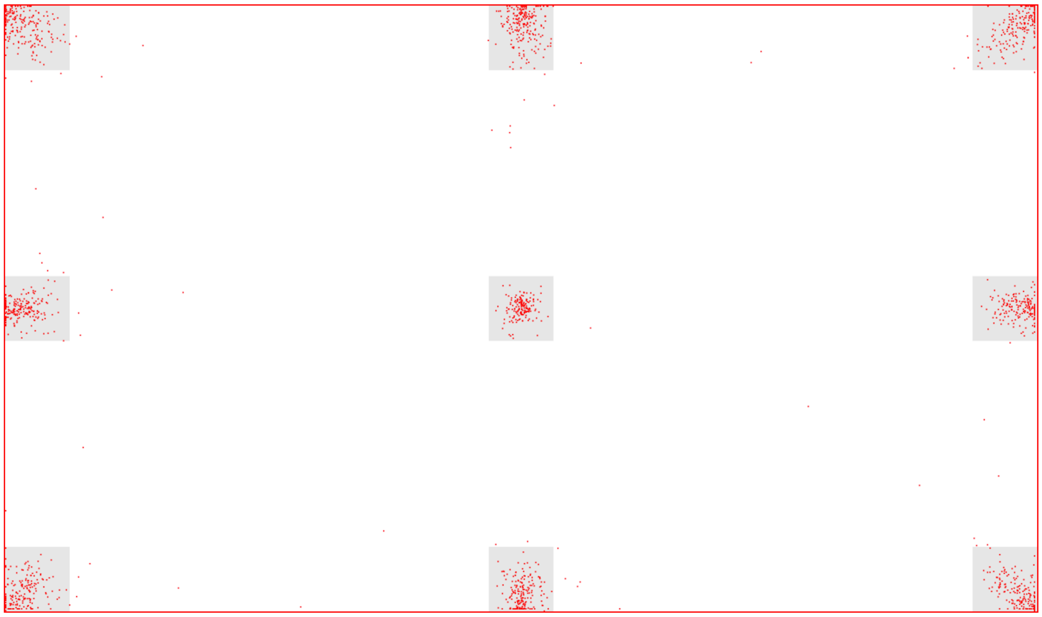

Figure 5.7: Selection Hit Distribution for Myo Across All Target Sizes

Based off these results we recommend target sizes and hit states, should be $\geq$ 60px for an interface powered by the touch-enhanced gesture control scheme to achieve at least a 90% accuracy rating. For in-air gesture control schemes, target sizes should be $\geq$ 100px to achieve at least a 90% accuracy rating.

Selection Hit Distribution

To observe the hit distribution for both devices across target sizes, the coordinates of all successful and non-successful hits were mapped onto their corresponding target size grid. As seen in Figure 5.4 and 5.5, coordinates for all successful hit or "clicks" are marked in red while non-successful hits are marked in black. Figure 5.4 represents all data for the 40px smartphone selection task while Figure 5.5 represents the 40px Myo data set. No concrete difference can be seen in hit distribution while looking at these two figures. Hit distribution for the other target sizes looks similar. If all hit coordinates across all target sizes are viewed at once, like in Figure 5.6 and Figure 5.7, it appears as if the Myo's hit distribution is wider than the smartphone's. This suggests that more of a visible difference could be seen if the study was reproduced with more participants.

5.0.2 Movement Task Results

The movement task consisted of three mazes of increasing difficulty. Participants used the Myo and smartphone to guide a cursor through the maze. If the edges were touched, the participant restarted that maze. During the maze, participants were asked to use the device's virtual clutch to pause at red dots called "virtual clutch points." Whereas the selection task measured selection within a system, the movement task measures the participant's ability to use the virtual clutch during an action (moving through the maze). Maze 1 had one virtual clutch point, Maze 2 had

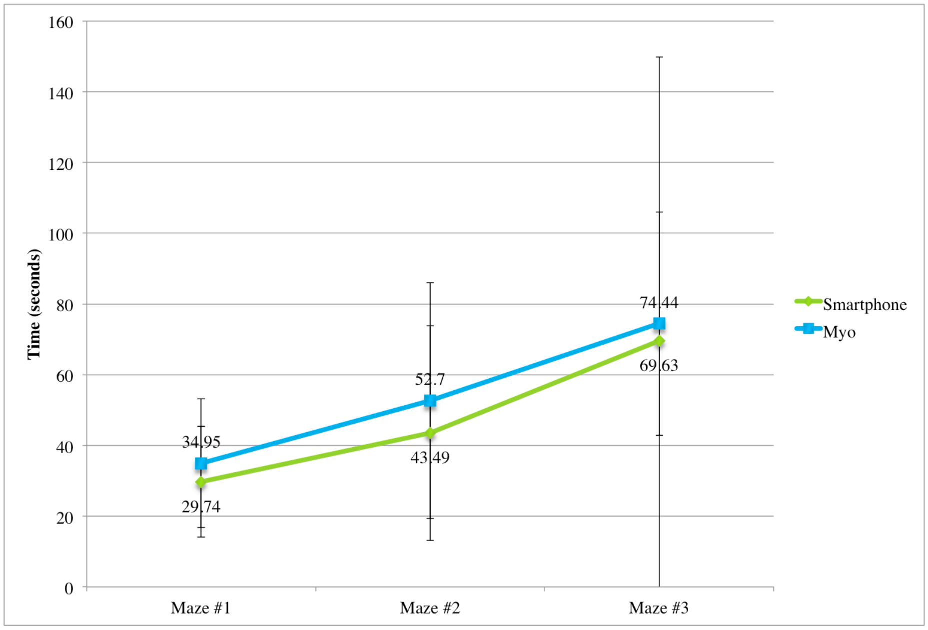

Figure 5.8: Movement Task Time Means and Standard Deviation

	Mean Smartphone Attempts	**Mean Myo Attempts**
Maze 01	2.1	1.4
Maze 02	2.37	1.67
Maze 03	2.7	1.7

Table 5.1: Movement Task Mean Attempts to Complete Mazes

two, and Maze 3 had three.

Movement Task Times

Movement task time was measured from the start of load of the maze to when the participant reached the end of the maze. During this time, participants used the device's virtual clutch to activate motion tracking, and to pause on virtual clutch points. The mean task time for each maze was slightly higher for the Myo, but no statistical significance was found when comparing each devices' data sets. The smartphone also had a slightly higher mean number of attempts to complete the mazes, as seen in Table 5.1, but this also showed no statistical significance. Observations during testing though showed easier navigation through the maze while using the Myo. This could have caused Myo's on average lower number of attempts needed to complete the mazes. This is illustrated in Figure 5.9 and 5.10, showing a participant's path through Maze 02 using both the smartphone and Myo. In Figure 5.9, the participant's green path is shakier than than the Myo's path seen in Figure 5.10. Shorter smartphone mean task times could then be caused by Myo's false-negative errors when participants tried executing the device's virtual clutch to pause on virtual clutch points during the maze. Participants during tested and in the post-study questionnaire noted they had to repeat to virtual clutch command until the Myo did recognize it. Data to support this idea is reviewed in the next section.

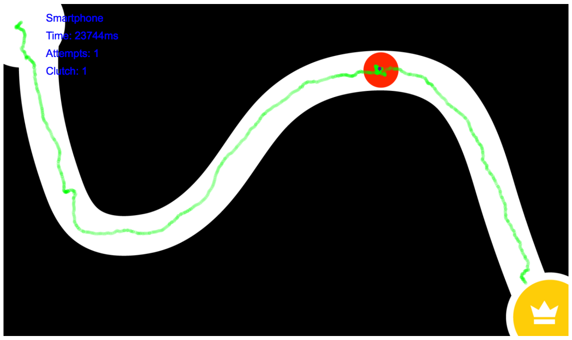

Figure 5.9: Participant 136 Movement Task Maze 02 with Smartphone

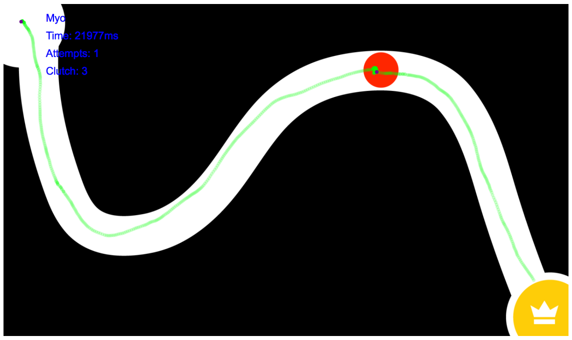

Figure 5.10: Participant 136 Movement Task Maze 02 with Myo

Movement Percent Error

While movement task times showed no statistical significance between devices, percent error of the virtual clutch command's execution during the movement tasks

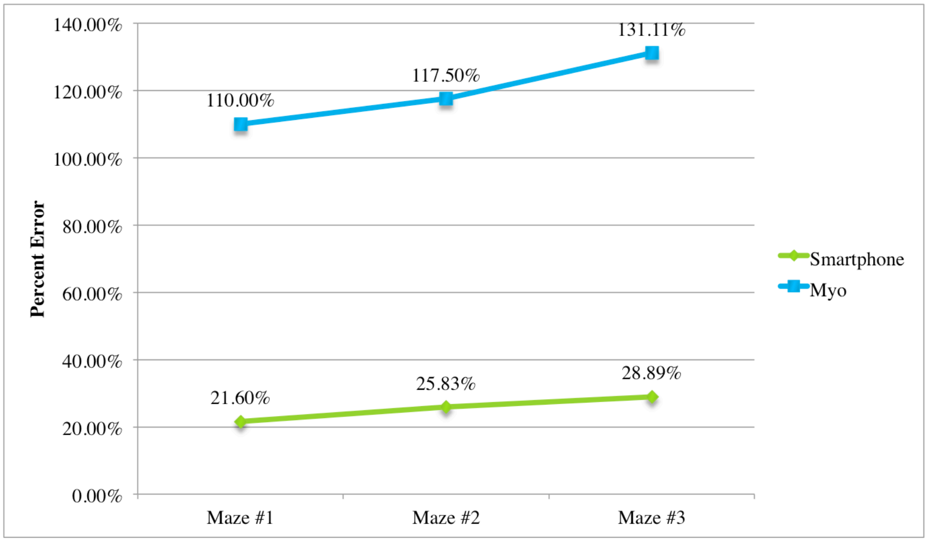

Figure 5.11: Movement Task Percent Error

did for all three mazes (p <0.001). For this task, both attempts to execute the virtual clutch and successfully read virtual clutch commands by the device were recorded. Both devices yielded high percent errors when using the virtual clutch during the movement tasks. Neither produced a mean percent error under 20% for any of the mazes, although the smartphone did come closer. As shown in Figure 5.11, the smartphone had a mean percent error of 21.60%, 25.83%, and 28.89% for Maze 01, 02, and 03 respectively. Although these percentages are high, the Myo failed to produce a mean percent error under 100%. The Myo had a mean percent error of 110%, 117.5%, and 131.11% for Maze 01, 02, and 03 respectively. This mirrors our findings in the selection task percent error results. Participants found execution commands like “clicking” and using the virtual clutch easier and more accurate with the smartphone than the Myo.

	Smartphone	**Myo**
I was satisfied with the ease of completing the selection task (clicking the boxes) with the...	6.17 ± 1.08	3.97 ± 1.5
I was satisfied with the ease of completing the movement task (the maze) with the...	5.43 ± 1.28	5.40 ± 1.55

Table 5.2: Mean of responses on 7-pt Likert scale for device ease of use with selection and movement tasks

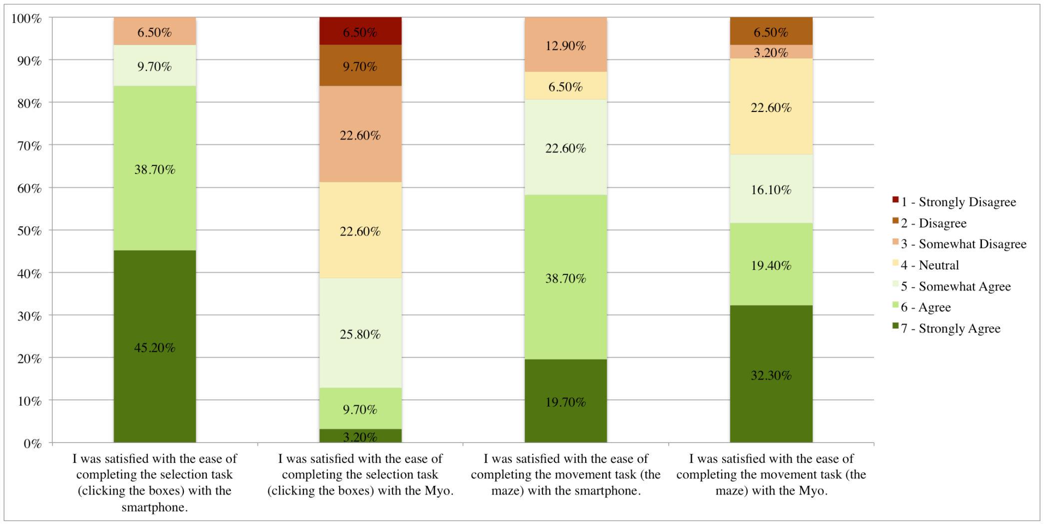

Figure 5.12: Responses for ease of use for both devices for the selection and movement tasks

5.0.3 User Preferences

After using both the smartphone and the Myo for all tasks, participants filled out a Subjective Preference Questionnaire gathering their perceived accuracy and speed of the devices. The questionnaire also focused on what participants thought of the ease of use for both devices and their preference of device, if any.

Selection and Movement Task User Preferences

Four questions were asked about the ease of completing the selection and movement task for both the smartphone and Myo. Participants responded on a 7-point Likert scale with 1 being “Strongly Disagree” and 7 being “Strongly Agree”. When asked to gauge “I was satisfied with the ease of completing the selection task (clicking the boxes) with the smartphone”, the mean response was 6.17 ± 1.08 compared to the Myo with a mean of 3.97 ± 1.5 (Table 5.2). T-test showed statistical significance with a $p < 0.0001$. The same type of questions were asked, but for the movement tasks. The mean response for the smartphone was 5.43 ± 1.28 with 5.40 ± 1.55 for the Myo (Table 5.2) with no statistical significance. The breakdown of the responses can be seen in Figure 5.12.

For these tasks, participants were more satisfied with the ease of the smartphone for the selection tasks compared to the Myo and were almost equally satisfied with both devices for the movement tasks. This aligns with our findings so far when looking at the data gathered for the selection tasks mean times and percent error. For the movement tasks, even though a higher percentage of participants agreed they were satisfied with using the smartphone in the breakdown seen in Figure 5.12, the percentages for agreement and disagreement are very close for both devices, while the mean response and t-test showed no significance.

Perceived Accuracy and Speed

The next section of the questionnaire asked if the participants were satisfied with the selection accuracy and speed of the devices. When prompted with, “I was satisfied with the selection accuracy of the smartphone,” the mean response was 6.13 ± 1.07 compared to the Myo’s mean response of 4.27 ± 1.41 (Table 5.3). Again, this reflects our results from the previous four questions reviewing satisfaction with the devices

	Smartphone	**Myo**
I was satisfied with the selection accuracy of the...	6.13 ± 1.07	4.27 ± 1.41
I was satisfied with the speed of the...	5.83 ± 1.32	5.63 ± 1.43

Table 5.3: Mean of responses on 7-pt Likert scale for device perceived selection accuracy and speed

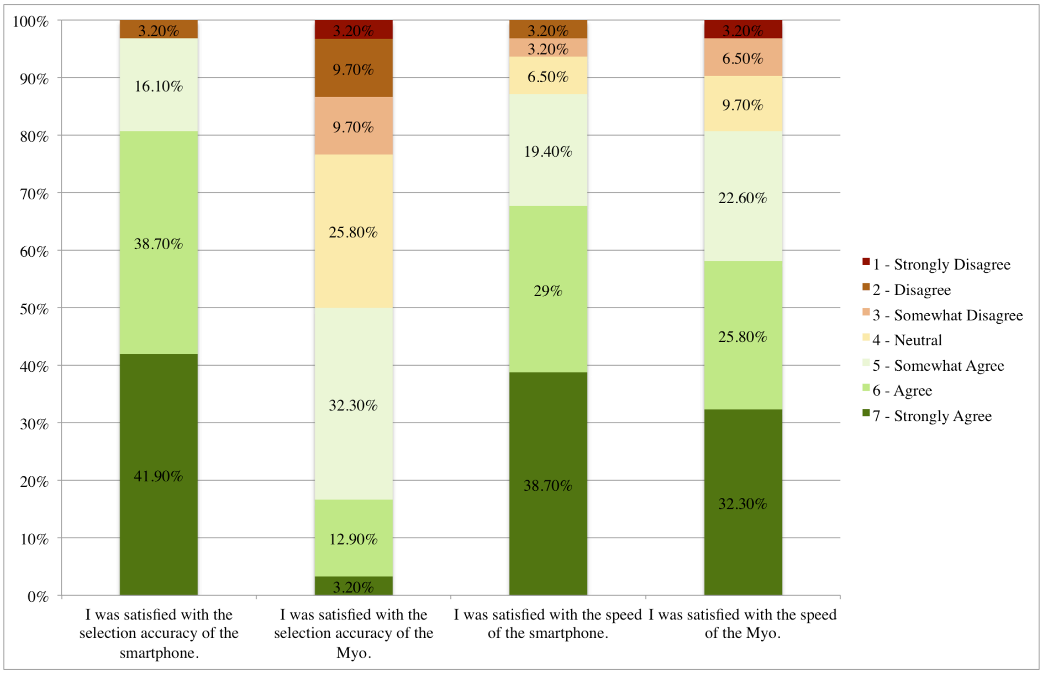

Figure 5.13: Responses for selection accuracy and speed for both devices

for the selection and movement tasks and shows statistical significance ($p < 0.0001$).

Device Ease of Use

At the end of the questionnaire, participants were asked to rank their satisfaction for the overall ease of use for each device and what device they thought they had the least difficulty using.

As seem in Figure 5.14, the majority of participants agreed that they were satisfied

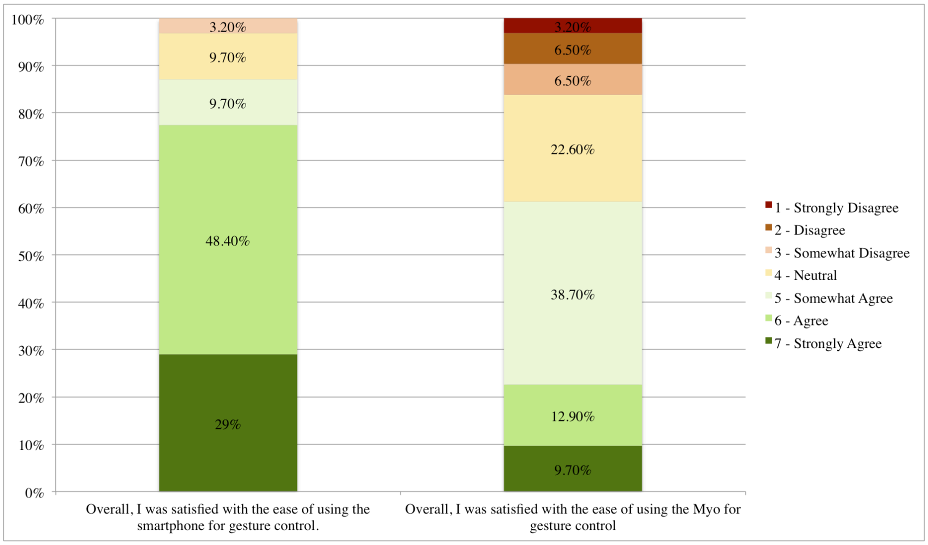

Figure 5.14: Responses for overall ease of use for devices

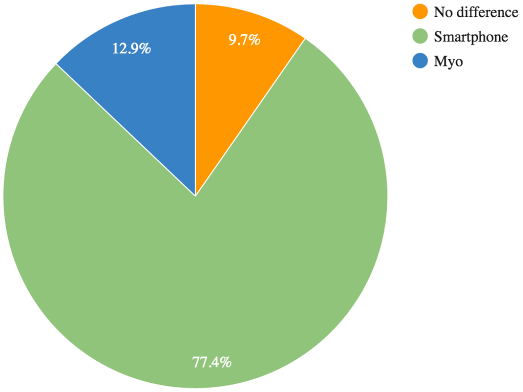

Figure 5.15: Responses for “Overall, what device did you have the least difficulty using?”

	Smartphone	**Myo**
Overall, I was satisfied with the ease of using the ... for gesture control.	5.9 ± 1.06	4.67 ± 1.45

Table 5.4: Mean of responses on 7-pt Likert scale for device satisfaction for overall ease of use.

with the ease of use for the smartphone at 87.1%. The majority of participants also agreed that they were satisfied with the ease of use for the Myo at 61.3%. The mean of these responses in Table 5.4, reveal statistical significance with the smartphone having an average rating of 5.9 ± 1.06 and the Myo 4.67 ± 1.45 ($p < 0.01$). The participants were also asked, "Overall, what device did you have the least difficulty using?" The results as seen in Figure 5.15 show that 77.4% of participants selected the smartphone the smartphone, 12.9% the Myo, and 9.7% saw no difference between the two.

6. Discussion

As seen in the data recorded from the movement tasks, the virtual clutch is recognized more accurately by our touch-enhanced gesture control scheme than the in-air gesture device. Improving the virtual clutch and reducing the impact of live mic syndrome was our starting goal. The movement tasks percent error for virtual clutch recognition resulted in a the smartphone's mean percent error of 21.60% for Maze 01, 25.85% for Maze 02, and 28.89% for Maze 03. The Myo however had incredibly higher percent errors at 110% for Maze 01, 117.5% for Maze 02, and 131.11% for Maze 03. T-tests revealed statistical significance for all three mazes. From this data we assume that stopping and starting motion tracking with the virtual clutch using the touch-enhanced gesture control scheme is faster and more accurate than the in-air wearable Myo. By designing a better virtual clutch, users can stop tracking to prevent the device from executing unintentional movements and reduce false-positive errors.

Also, as seen in the recorded data from the selection tasks, the smartphone shows significant improvement in selecting targets of 100px, 40px, and 20px size as well as improvement in time to complete said tasks. The virtual clutch was not a focus of the selection tasks, but away to ensure the touch-enhanced gesture control scheme performance level was at least equal to the Myo. By combing gesture with touch, we also made it easier for users to execute commands, like selecting, within a system, and have decreased the amount of time it takes them to complete tasks. It seems that by removing in-air commands and replacing them with touchscreen interaction, we not only improve recognition of the virtual clutch but of selection commands as well. This could be why in the post-study questionnaire, 77.4% of participants responded that the smartphone was easier to use than the Myo. In both sets of tasks, the smartphone had a lower mean percent error that was statistically significant when it came to selecting and using the virtual clutch.

6.0.1 Comparing Selection Commands

When it came to executing commands, the smartphone out-preformed the Myo. It must also be noted that during testing, the Myo was observed to execute a different command than the participant intended at least once for 18 out of the 30 participants. One participant was so used to this error, they tried both commands to try and use the virtual clutch in the movement tasks. The participant said, "Sometimes [the Myo] would not register when I double tapped, but it would when I make a fist. So I tried that."

These results indicate that pausing and starting control with the smartphone is faster and more accurate than the Myo. Also, as noted in the Results section, each participant was able to use the smartphone instantly, when two excluded participants could not get the Myo to sync with them, and four needed custom calibration profiles to use the Myo. Our touch-enhanced gesture scheme is more accessible and less discriminant when it comes to the people who can operate it.

During testing it was observed that the Myo's performance in the selection tasks regarding time was inhibited by the device's failure to read the participant's commands. This caused participants to repeat the same command until it was read by the Myo. This occurred with the smartphone as well, as seen in the movement tasks' mean percent errors, but not as severely. For example, for movement task Maze 01, the smartphone had a mean percent error when executing the virtual clutch command of 21.60% compared to the Myo's 110%.

When commands were actually read, the Myo's higher percent errors for the selection tasks 100px, 40px, and 20px, indicate these "clicks" often missed their target.

Participants noted for both devices, cursor movement occurred when executing a command. The movement was reported to be larger and caused more unsuccessful clicks for the Myo.

> "Making a fist caused the pointer to jump too much for an accurate click."

> "The gestures used to 'clutch' and 'click' with the Myo alter the location of the [cursor]. It was much more difficult to click a small target when the act of clenching my fist forced the [cursor] to jiggle."

This jump when executing a command was a point of frustration for the participants. One noted that with the smartphone is was "so much easier to stay where I wanted to stay." And that her arm "will just move with the Myo."

6.0.2 Comparing Speed

Parhi et al. [17] when testing different target sizes for touchscreen with one thumb, saw that task completion time increased with the decrease of target sizes. This study's results follows a similar pattern for the smartphone. Here however, the first and largest target size, 100px, has a higher task time for both devices than the sequential task of 80px target sizes. The difference between these tasks for both devices do not show statistical difference, but it could be attributed to a learning curve. During testing, it was observed that these initial higher task times could be caused to participants learning how to use both devices for the first time. While all participants were given a tutorial on how to use each device before the tasks, participants could have still been warming up.

As for the speed of the devices even though the selection task showed quicker task completion times for the smartphone, in the Subjective Preference Questionnaire, we saw that there was very little difference in the participants' satisfaction rating for the smartphone's and Myo's speed (Figure 5.13). In response to question "Overall, what device did you have the least difficulty using?", participants were asked "Why was that device overall less difficult to use?" and if they had any additional comments. In these responses, the previous differences were noted by some of the participants. One participant that submitted "No Difference" in response to what device was the least difficult to use, said "Movement on Myo felt more natural at times, while the phone was much more responsive to clicking." Another, who also wrote "No Difference," said:

> "I said no difference mainly because they had their difficulties, but in different ways. The smartphone was extremely sensitive which made it difficult for me, a shaky person, to keep my selections accurate at times, but then the Myo was not always picking up the muscle movements in my forearm, and was, at times, picking up the wrong ones."

As mentioned in subsection 5.0.2 Movement Task Times, even though there were no statistical difference between devices for movement tasks' time and attempts, participants noted that the smartphone felt shakier. This shake could have caused the the higher mean attempts to complete the maze with the smartphone. Some participants attributed this to the smartphone being handheld and picking up hand tremors. These differences could also be traced back to the devices' hardware and software. The Myo hardware for gyroscopic and accelerometer detection is more advanced and sophisticated than the smartphone's. Also, the software converting this data into 2D motion are more robust. In Chapter 7.1 Future Work, we discuss what improvements can be made to the touch-enhanced gesture control scheme.

6.0.3 Fatigue

Fatigue was not recorded in either the testing or in the questionnaire. It was observed, however, that the Myo caused fatigue for some of our participants. This was noted for 10 of the participants during testing by either verbal or physical clues. Participants either noted they were tired, took breaks between the Myo's tasks, or shook out the arm they were using with the Myo. Two participants even used their free arm to hold up their Myo-wearing arm to help lessen fatigue. Participants were observed to be using two hands with the smartphone as well. When asked why, they explained it helped them steady their control and lessen shake, and not to be caused by fatigue.

7. Conclusion and Future Work

In summary, we designed and evaluated a touch-enhanced gesture control scheme to aid in reducing live mic syndrome by improving recognition of a virtual clutch command. By using a smartphone as a gesture input device, we developed an accessible input scheme that also uses its touchscreen to toggle motion tracking and handle selection inputs. We found this system executes commands faster than the in-air wearable gesture device, the Myo armband. Also, 77.4% of participants reported the touch-enhanced control scheme as easier to use. Our touch-enhanced gesture control scheme can be used to control digital displays to multiplayer games where user movement is wanted.

7.1 Future Work

Although the touch-enhanced gesture control scheme using the smartphone achieved improvements with the virtual clutch, and lessened issues caused by live mic syndrome, both the Myo and smartphone can be improved. Based off these results, it is hypothesized that the Myo would out-preform the smartphone in strictly movement/motion based tasks, like moving from point A to point B, but not by much. But, if the user needs to take action, like execute a command, the touch-enhanced control scheme would be the better controller. In the former situation, the smartphone can be improved by continuing to work on its JavaScript handler to lessen the impact of hand shake and make for a smoother controller. In the latter situation, the Myo's hardware would need to be altered to increase its sensitivity to recognize in-air gestures. Where technology stands today, improving the smartphone's JavaScript handler is quicker than an hardware update for the Myo. Updating hardware and sensors in technology for gesture control must be balanced against cost. If more advanced sensors will make the total cost of the gesture device not easier accessible, the performance benefit may not be worth affordable [27].

7.1.1 Further Studies

Participants. As mentioned in section 5.0.1 Selection Hit Distribution, more participants might reveal new results. The selection hit distribution aggregated coordinates started to reveal differences between the devices. More participants would make this difference easier to observe, if any.

False-negative Errors. Only during the movement tasks for the virtual clutch command did we record the number of attempts it took participants to execute the command. Participants counted off how many times it took them to swipe on the smartphone and double-tap with the Myo. This was not done with both devices during the selection tasks though.

As mentioned, it was observed during the study that the Myo's longer selection task completion time could be attributed to the increased number of attempts it took participants to execute the selection commands before the Myo actually read it. Setting up a system that could record these attempts, and what target in the 3x3 grid the participant was trying to "click", could provide further insights into how both the smartphone and Myo preform.

In the additional comments section of the questionnaire, and out-loud in the study, participants noted that it felt like the Myo was "less sensitive" or had a harder time reading their commands when trying to "click" the targets in the corners of the grid. Participants thought this was caused by their extended arms as they reached for the targets. Tracking selection attempts and intended targets could reveal a correlation here. It could also provide precent errors depending on target location on a screen for both devices.

Vision-based vs. Wearable Gesture Control. This study focuses only wearable gesture control devices. While both vision-based and wearable gesture devices suffer from that same shortcomings, their hardware and software differ. Future work could compare our touch-enhanced control scheme to a vision-based gesture device, like the Kinect. We focused on wearable gesture control because it is mobile and can travel with the user, like a smartphone. This study could also be reproduced using other wearable gesture devices, like the Gest.

Commands. Participants only used two commands for both devices. Tapping and vertical swiping for the smartphone and making a fist and double-tapping for the Myo. Further research on different commands, double-tapping and vertical swiping for the smartphone and waving left and right foe the Myo, can provide more data on these two systems and how they behave with more complex control inputs.

Other Mutlimodal approaches. We have proposed, created, and analyzed a touch-enhanced gesture control scheme. While only focusing on touch and gesture, their are other mutlimodal approaches that can be explored. For example, voice control and eye-tracking are two other interaction methods that can be combined with gesture control. These interaction types can also be explored to help solve gesture control shortcomings.

7.1.2 Applications

Our touch-enhanced gesture control scheme is a mobile solution with many potential applications. A limitation of the system however is the requirement for the smartphone to be held by the user. Benefits of vision-based gesture control devices include the ability to control other devices or access information with sterility [27]. Even some wearable devices that are not worn on the user's hands, like the Myo, have this benefit as well. Another limitation is accessibility of those with physical handicaps. Again, vision-based gesture devices can help users with disabilities control devices and applications. Although the touch-enhanced system is accessible to those who have a modern smartphone, and was able to be used by all our participants when the Myo was not, it still requires fine motor skills.

The touch-enhanced system works best in situations that afford handheld devices. The touch-enhanced gesture control scheme can be used for applications in place of the Wii remote, PlayStation Move controller, and other remote like gesture control devices. As seen in Bragdon et al. and Code Space [7], the touch-enhanced gesture control scheme works well in meeting situations as a remote for sharing, organizing, and annotating information. The one area where touch-enhanced gesture excels and other gesture remotes do not, is by using the smartphone we can build for massive multi-user systems. Since smartphones are a common consumer product, more people can interact with a game or display at once. Our system uses Node.js and WebSockets to connect the smartphone to a display. Right now, for our study only one person at a time can interact with a display. But, during development we have built this system to support multiple users.

The work done for this study is open-source and available to the public. The code used to translate 3D movement into 2D motion for both the smartphone and Myo will be available. Thalmic Labs, creator of the Myo, is also continuing to create APIs for the Myo in a multitude of programming and scripting languages. Further work will be done to improve the touch-enhanced control scheme create an JavaScript plugin for development. Since work for the touch-enhanced gesture control scheme is open-source, other developers can use it as a starting point with their gesture control work. Custom interfaces can be created for the smartphone's screen. Buttons can be added along with detection for different touchscreen commands. There is also potential to incorporate inputs for 3D touch commands like those released with the iPhone 6s. These commands include detection of a touch's pressure and duration.

Also with this touch-enhanced gesture control scheme, instead of buying a device dedicated to gesture control, like the Myo or Leap Motion, developers can work with equipment they already have, their smartphone. Developers can create gesture control systems that users do not have to buy new controllers for. They too can use their own smartphones. Because of this, the touch-enhanced control scheme can be used to interact with public installations since people are more than likely to already have a smartphone on-hand. This scheme can also be used to power single and massive multiplayer games.

8. Additional Materials

8.1 Questionnaires

8.1.1 Pre-Test Questionnaire

1. What is your age?
2. What is your gender?
3. What is your occupation?
4. Do you own a smartphone? If so, what kind?
5. On average, how many hours per day do you spend on your smartphone?

   ☐ Less than 1 hour a day

   ☐ 1-2 hours

   ☐ 2-3 hours

   ☐ 3-4 hours

   ☐ More than 4 hours a day

6. What is the highest level of education you have completed?:

   ☐ High School

   ☐ Bachelor?s Degree

   ☐ Master?s Degree

   ☐ Doctoral Degree

   ☐ Other

7. Have you ever heard any of the following gesture control devices? Check all that apply.

☐ Kinect

☐ Wii Remote

☐ Leap Motion

☐ Myo

☐ Gest

☐ Ring Zero

☐ Nod

☐ Reemo

☐ Gestor

8. Have you ever used any of the following gesture control devices? Check all that apply.

☐ Kinect

☐ Wii Remote

☐ Leap Motion

☐ Myo

☐ Gest

☐ Ring Zero

☐ Nod

- ☐ Reemo
- ☐ Gestor
- ☐ Other ____________________________

8.1.2 Subjective Preference Questionnaire

On a scale of 1-7, how strongly do you agree with the following statements?

1. I was satisfied with the ease of completing the selection task (clicking the boxes) with the **smartphone.**

- ☐ 1 - Strongly disagree
- ☐ 2 - Disagree
- ☐ 3 - Somewhat disagree
- ☐ 4 - Neutral
- ☐ 5 - Somewhat agree
- ☐ 6 - Agree
- ☐ 7 - Strongly agree

2. I was satisfied with the ease of completing the selection task (clicking the boxes) with the **Myo.**

- ☐ 1 - Strongly disagree
- ☐ 2 - Disagree
- ☐ 3 - Somewhat disagree
- ☐ 4 - Neutral

- ☐ 5 - Somewhat agree
- ☐ 6 - Agree
- ☐ 7 - Strongly agree

3. I was satisfied with the ease of completing the movement task (the maze) with the **smartphone.**

- ☐ 1 - Strongly disagree
- ☐ 2 - Disagree
- ☐ 3 - Somewhat disagree
- ☐ 4 - Neutral
- ☐ 5 - Somewhat agree
- ☐ 6 - Agree
- ☐ 7 - Strongly agree

4. I was satisfied with the ease of completing the movement task (the maze) with the **Myo.**

- ☐ 1 - Strongly disagree
- ☐ 2 - Disagree
- ☐ 3 - Somewhat disagree
- ☐ 4 - Neutral
- ☐ 5 - Somewhat agree
- ☐ 6 - Agree

☐ 7 - Strongly agree

5. I was satisfied with the selection accuracy of the **smartphone.**

    ☐ 1 - Strongly disagree

    ☐ 2 - Disagree

    ☐ 3 - Somewhat disagree

    ☐ 4 - Neutral

    ☐ 5 - Somewhat agree

    ☐ 6 - Agree

    ☐ 7 - Strongly agree

6. I was satisfied with the accuracy of the **Myo.**

    ☐ 1 - Strongly disagree

    ☐ 2 - Disagree

    ☐ 3 - Somewhat disagree

    ☐ 4 - Neutral

    ☐ 5 - Somewhat agree

    ☐ 6 - Agree

    ☐ 7 - Strongly agree

7. I was satisfied with the speed of the **smartphone.**

    ☐ 1 - Strongly disagree

- ☐ 2 - Disagree
- ☐ 3 - Somewhat disagree
- ☐ 4 - Neutral
- ☐ 5 - Somewhat agree
- ☐ 6 - Agree
- ☐ 7 - Strongly agree

8. I was satisfied with the speed of the **Myo.**
   - ☐ 1 - Strongly disagree
   - ☐ 2 - Disagree
   - ☐ 3 - Somewhat disagree
   - ☐ 4 - Neutral
   - ☐ 5 - Somewhat agree
   - ☐ 6 - Agree
   - ☐ 7 - Strongly agree
9. Overall, I was satisfied with the ease of using the **smartphone** for gesture control.
   - ☐ 1 - Strongly disagree
   - ☐ 2 - Disagree
   - ☐ 3 - Somewhat disagree
   - ☐ 4 - Neutral

- ☐ 5 - Somewhat agree
- ☐ 6 - Agree
- ☐ 7 - Strongly agree

10. Overall, I was satisfied with the ease of using the **Myo** for gesture control.
    - ☐ 1 - Strongly disagree
    - ☐ 2 - Disagree
    - ☐ 3 - Somewhat disagree
    - ☐ 4 - Neutral
    - ☐ 5 - Somewhat agree
    - ☐ 6 - Agree
    - ☐ 7 - Strongly agree
11. Overall, what device did you have the least difficulty using?
    - ☐ Smarphone
    - ☐ Myo
    - ☐ No difference
12. Why was that device overall less difficult to use?
13. Any additional comments?

Bibliography

[1] M. Alcoverro, X. Suau, J. R. Morros, A. Lpez-Mndez, A. Gil, J. Ruiz-Hidalgo, and J. R. Casas. Gesture control interface for immersive panoramic displays. *Multimedia Tools and Applications*, 73(1):26, 2014. doi: 10.1007/s11042-013-1605-7.

[2] J. Anderson and L. Rainie. Digital life in 2025, 2014.

[3] M. Baglioni, E. Lecolinet, and Y. Guiard. Jerktilts: using accelerometers for eight-choice selection on mobile devices. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 121–128. ACM, 2011.

[4] G. Bailly, D.-B. Vo, E. Lecolinet, and Y. Guiard. Gesture-aware remote controls: guidelines and interaction technique. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 263–270. ACM, 2011.

[5] T. Baudel and M. Beaudouin-Lafon. Charade: remote control of objects using free-hand gestures. *Commun. ACM*, 36(7):28–35, 1993. ISSN 0001-0782. doi: 10.1145/159544.159562. URL `http://dl.acm.org/citation.cfm?doid=159544.159562`.

[6] T. Bossuyt and N. Hillgren. Combining touch & mid-air gestures on public displays. *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*, 14:65, 2012.

[7] A. Bragdon, R. DeLine, K. Hinckley, and M. R. Morris. Code space: touch+ air gesture hybrid interactions for supporting developer meetings. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 212–221. ACM, 2011.

[8] Y. Chang, S. L'Yi, K. Koh, and J. Seo. Understanding users' touch behavior on large mobile touch-screens and assisted targeting by tilting gesture. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1499–1508. ACM, 2015.

[9] H. Chudgar and S. Mukerjee. S control: Accelerometer-based gesture recognition for media control, 10-11 Oct 2014 2014.

[10] D. Fitton, I. S. MacKenzie, J. C. Read, and M. Horton. Exploring tilt-based text input for mobile devices with teenagers. In *Proceedings of the 27th International BCS Human Computer Interaction Conference*, page 25. British Computer Society, 2013.

[11] M. Goel, L. Findlater, and J. Wobbrock. Walktype: using accelerometer data to accomodate situational impairments in mobile touch screen text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2687–2696. ACM, 2012.

[12] A. Hard. The 2016 bmw 7 series is the first production car with gesture control. here's how it works, 2015. URL `http://www.digitaltrends.com/cars/2016-bmw-7-series-gesture-control-pictures-video-news/`.

[13] E. Jones, J. Alexander, A. Andreou, P. Irani, and S. Subramanian. Gestext: accelerometer-based gestural text-entry systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2173–2182. ACM, 2010.

[14] Just Dance Now. Just dance now, 2014. URL `http://www.justdancenow.com`.

[15] S. Ljubic, M. Kukec, and V. Glavinic. Tilt-based support for multimodal text entry on touchscreen smartphones: using pitch and roll. In *Universal Access in Human-Computer Interaction. Applications and Services for Quality of Life*, pages 651–660. Springer, 2013.

[16] Z. Lu, X. Chen, Q. Li, X. Zhang, and P. Zhou. A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices. *IEEE Transactions on Human-Machine Systems*, 44(2):293–299, 2014.

[17] P. Parhi, A. K. Karlson, and B. B. Bederson. Target size study for one-handed thumb use on small touchscreen devices. In *MobileHCI '06*, pages 203–210, San Jose, CA, 2006. ACM SIGCHI.

[18] J. Rekimoto. Gesturewrist and gesturepad: Unobtrusive wearable interaction devices. In *Wearable Computers, 2001. Proceedings. Fifth International Symposium on*, pages 21–27. IEEE, 2001.

[19] J. Ruiz, Y. Li, and E. Lank. User-defined motion gestures for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 197–206. ACM, 2011.

[20] T. Schlömer, B. Poppinga, N. Henze, and S. Boll. Gesture recognition with a wii controller. In *Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages 11–14. ACM, 2008.

[21] A. Smith. U.s smartphone use in 2015, 2015.

[22] J. Song, G. Sörös, F. Pece, S. R. Fanello, S. Izadi, C. Keskin, and O. Hilliges. In-air gestures around unmodified mobile devices. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 319–329. ACM, 2014.

[23] T. Soppe. Gesture control: Still a pipe dream?, April 15th, 2010 2015. URL `http://www.forbes.com/video/4493023551001/`.

[24] UI Design Basics. Ui design basics, 2015.

[25] J. Underkoffler and K. Parent. System and method for gesture based control system, 09/09/2014 2014.

[26] S. Verma, P. Pillai, and Y. Fun Hu. Development of an eye-tracking control system using aforge.net framework. *International Journal of Intelligent Systems Technologies and Applications*, 11(3/4):18, 2012. ISSN 1740-8865 1740-8873. doi: 10.1504/IJISTA.2012.052485.

[27] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan. Vision-based hand-gesture applications. *Communications of the ACM*, 54(2):60–71, 2011.

[28] L. Weberg, T. Brange, and Å. W. Hansson. A piece of butter on the pda display. In *CHI'01 Extended Abstracts on Human Factors in Computing Systems*, pages 435–436. ACM, 2001.

[29] D. Wigdor and R. Balakrishnan. Tilttext: using tilt for text input to mobile phones. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 81–90. ACM, 2003.

[30] D. Wigdor and D. Wixon. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Elsevier, 30 Corporate Drive, Suite 400, Burlington, MA 01803, USA, 2011. ISBN 9780123822314.