IN-AIR UN-INSTRUMENTED POINTING PERFORMANCE

MICHELLE A. BROWN

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

GRADUATE PROGRAMME IN COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

APRIL 2014

ABSTRACT

I present an analysis of in-air un-instrumented pointing and selection. I look at the performance of these systems and how this performance can be improved, with the eventual goal that their throughput reaches that of the mouse. Many potential limiting factors were explored, such as latency, selection reliability, and elbow stabilization. I found that the un-instrumented in-air pointing as currently implemented performed significantly worse, at less than 75% of mouse throughput. Yet, my research shows that this value can potentially reach mouse-like levels with lower system latencies, user training, and potentially improved finger tracking. Even without these improvements, the large range of applications for un-instrumented 3D hand tracking makes this technology still an attractive option for user interfaces.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Wolfgang Stuerzlinger for all his help, guidance, and dessert. All three of these were very important to the completion of this work. I would like to thank my sister, Nicole, for being my best friend and for always having my back. Your absolute confidence that I can move the world is the only reason I can. I would also like to thank Andriy Pavlovych for building the “ring button” used in one of my experiments, for modifying my desk, and for always being crazy enough to go on adventures with me. I would like to thank Steven Castellucci for the creation of the thesis template, his moral support, his excellent conversation abilities, and, of course, his thoughtful supply of milk flavoured pocky. Lastly, I would like to thank my parents for bringing me up in a world filled with books and magic.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1
Introduction

In the everyday world, pointing at objects to reference them is a fundamental task that spans across 2D and 3D selection. When people are using computers with others they will often point to objects visible on screen to indicate them to another person. This suggests that exploring the benefits and trade-offs of in-air pointing as an interaction modality is a worthwhile endeavour.

In-air pointing, especially un-instrumented pointing, promises to be less intrusive and possibly more convenient than current interaction methods. I identify characteristics of this technology, measure throughput based on the ISO 9241-9 standard relative to the mouse, identify possible technological improvements, and identify situations where this type of selection method is beneficial.

1.1 Motivation

A number of hand tracking devices have recently appeared on the market and many more have been announced. This includes the Leap Motion, the DUO, the CamBoard pico, the touchless control system by Elliptic Labs, and 3 Gear Systems' hand tracking camera setup. These new devices promise revolutionary and "natural" control of your computer. However, the performance of such in-air un-instrumented hand tracking devices relative to other pointing devices has yet to have been evaluated.

While it is widely assumed that 3D un-instrumented tracking does not perform as well as the mouse, there is no scientific study that quantitatively shows this. Furthermore, the reasons for a possibly lower throughput have not been explored. Currently it is unclear whether these systems have a lower throughput due to various human hand or arm postures, differences in device latency, poor click detection methods, sub-optimal tracking algorithms, or potentially even human limitations. As such, it is unclear what developers could do to improve the performance of such systems.

Un-instrumented in-air pointing technologies are ideal in situations when users are concerned with sterility, situations where there is no mouse such as with a laptop or tablet, and situations where smudges are a concern, such as when using a mobile device while cooking. This last scenario occurs when people today consult the Internet to find new recipes to use. Instead of printing the recipe, users then bring their laptops, tablets, or phones into the kitchen to reference while cooking. Yet, currently all interaction with the device requires users to first wash their hands or risk getting their device dirty. Un-instrumented in-air pointing avoids this problem and enables simple interaction with scrolling, unit conversions, cooking timers, or the music player. In order for this vision to become a reality, the performance of such systems needs to be evaluated and the benefits and shortcomings clearly understood.

Before investigating how well un-instrumented 3D pointing works, I first devised experiment 1 to determine the best operational method for un-instrumented pointing. This experiment also explores two possible explanations for a possibly lower throughput:

pointing method (finger vs. hand) and the effect of elbow stabilization. Past work has shown that pointing with the finger alone (with the arm immobilized) affords significantly lower throughput compared to the whole hand [5]. Elbow stabilization has the potential to improve performance through better accuracy. I then designed experiment 2 to directly compare un-instrumented 3D tracking to the mouse by using the same selection technique for both devices. This second experiment thus also explores the effect different selection methods have on user performance. The reason I decided to directly compare un-instrumented pointing to the mouse was to enable my work to be calibrated against other pointing studies, including MacKenzie and Jusoh's work [24]. Next, I designed pilot study 1 to determine if using a chopstick as a pointing device (instead of a finger) yielded a performance benefit. Then, pilot study 2 was devised to look at how much reducing latency could improve the throughput results found so far. Experiment 3 was designed to help narrow down the causes of throughput differences between pointing with a finger and pointing using the chopstick. Finally, experiment 4 was designed to look at the effect that un-reliable selection methods have on throughput. The purpose of this final study was to quantify the effects of a known problem precisely, rather than to determine weaknesses and possible improvements to un-instrumented pointing systems as the other studies aim to do. By quantifying the effect of this problem, developers can then determine how much they can expect the performance of their systems to decrease based on the reliability of their selection mechanisms.

1.2 Contributions

My contributions are:

- The first accurate performance measurement for in-air un-instrumented pointing with the ISO 9241-9 methodology.
- A comparison of two in-air target acquisition methods: finger pointing and whole hand movement.
- An analysis of the effect of elbow stabilization on selection performance.
- An evaluation of the performance benefit of using a ring button vs. selection with the other hand.
- An evaluation of the effect selection reliability has on performance.
- An analysis of the effect of a stable pointing device (chopstick) on selection performance.
- An analysis of the effect of a perfectly rigid, cylindrical finger (finger cast) on selection performance.
- An analysis of the effect of bending fingers toward the tracking device and how this impacts selection performance.
- An analysis of the effect of a reduction in latency on selection performance.

1.3 Leap Motion

The device I selected to represent un-instrumented systems was the Leap Motion (see Figure 1-1). The Leap Motion system was selected as the tracking system for this work after experimentation with other in-air hand tracking systems.

Figure 1-1: The Leap Motion hand tracking device.

The first system I considered was 3 Gear Systems' hand tracker version 0.9.22 (see Figure 1-2). This Kinect-based system had a very high end-to-end latency of about 170 ms and yielded very noisy 3D data.

Figure 1-2: Picture of the 3 Gear System setup. Two Kinect™ cameras on either side of metal frame track hands operating in the space above the desk.

In a pilot study with this system, throughput was estimated to be 1.5 bps (bits per second) in the best performing condition, with selection using the space bar. I also considered other 3D tracking systems. Ultimately, I decided not to use these, as they require the user to put on extra equipment to use the device. Overall, I found the Leap Motion to be very competitive. It easily affords pixel-accurate pointing, due to its' low noise level. My first latency measurement of this system based on version 0.8.0 firmware identified 85 ms of end-to-end latency. As this was reasonably close to the 75 ms reported for a 3D tracking system in a study of effects of latency on 3D interaction [38], I decided to use this device in my research.

The Leap Motion is 1.27 cm in height, 3 cm in width, 7.62 cm depth, and weighs 45 grams. It contains three infrared lights and two infrared cameras and as such, it is a tracking system based on stereovision (see Figure 1-3). This system tracks predefined objects of fingers, hands, and tools. A tool by the developers is defined as any object that is longer, straighter, and thinner than a finger.

In another pilot study I found that only about half of all performed “click” motions, a short up and down motion of a finger, registered with the version 0.8.0 of the Leap Motion system as a successful “click”. I deemed this success rate much too low to be practically competitive as a method to indicate selection. Thus, I did not consider using this selection method in my research, and used alternate selection methods. These methods relied either on the space bar operated with the non-dominant hand or a ring button operated with the dominant hand. The poor performance of the gesture selection technique caused me to wonder about the impact of un-reliable selection devices on throughput. This motivated experiment 4 (see section 3.6).

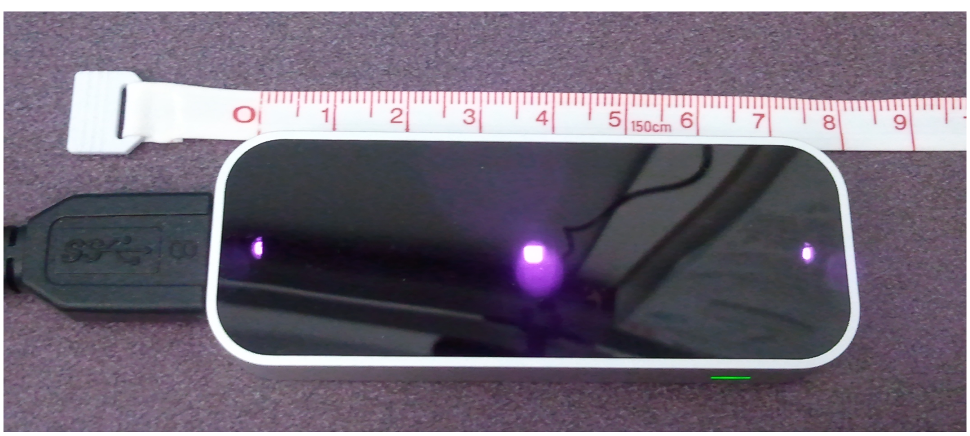

Figure 1-3: Leap Motion pictured next to tape measure for scale (depth value being measured). The three infrared LED’s show up as bright dots.

Chapter 2
Background Literature

This section will discuss the related research for my topic.

2.1 Fitts' Law and Pointing

Fitts' Law is an empirical model that describes the speed accuracy tradeoff in pointing tasks [25]. It can be used both as a predictive model and as a way to calculate throughput. The model is $MT = a + b \times log_2(\mathrm{D} / W + 1)$. In this model, MT is movement time, D is the target distance, W is the target size, and a and b are derived from linear regression. The log term in this model is referred to as the index of difficulty (ID). The ID describes the difficulty of selecting a particular target by combining the distance and the size of the target into a single value. Fitts' law implies that the further away or the smaller a target is, the harder it will be for a user to select. Although Fitts' law was originally developed for one-dimensional pointing tasks, it has been successfully adapted to 2D pointing tasks and can describe even some 3D pointing tasks [40].

Building on decades of Fitts' law based studies, the ISO 9241-9 standard [15] has been developed to standardize experimental methodologies and to improve the quality of Fitts' law data. In this standard, throughput is the primary measure of performance. Throughput is calculated as $TP = log_2(D_\mathrm{e} / W_\mathrm{e} + 1) / MT$. In this equation, D_e is the effective distance and W_e the effective width. These effective values measure the task that the user actually performed, not the one that she or he was presented with [25]. This reduces variability in identical conditions, which also facilitates comparisons between different Fitts' law studies

2.2 Ray Pointing

Ray pointing is a method for pointing at objects. In this method, the user will move a tracked object, such as a pen or laser pointer, or a tracked arm or finger and orient it in the direction she or he wishes to point to, such that the tracked entity forms a ray toward the desired pointing location. The first object along that ray is then traditionally highlighted and selected when the user indicates selection, e.g., through a button click. Ray pointing remains a popular selection method for large screen and virtual reality systems. Many studies have investigated this technique, including [6,9,11,12,16,21,24,29,31,40,41,43,48]. All these comparisons used various 3D tracking systems to implement ray pointing. However, users cannot simply walk up to these systems and start using them; they must first put on the proper tracking equipment or grab an instrumented device.

Ray pointing uses 3D input to afford control over a 2D cursor. Effectively users rotate the wrist (or finger) to move the cursor. Balakrishan and MacKenzie quantified the relative bandwidth of the fingers, wrist, and forearm and found that the wrist and forearm afford bandwidth of about 4.1 bits/s, while a finger exhibited only 3.0 bits/sec [5]. Another option for 2D cursor control is to directly map 3D motion to 2D motions by dropping the third degree of freedom recorded by the 3D tracking system. With this input method, the user then has to move her or his whole hand. The cursor can be controlled efficiently by either moving the finger or by moving the whole hand [4]. On the other hand, tracking very small hand rotations with 3D tracking systems with sufficient accuracy is difficult, as any amount of tracking noise is effectively magnified increasingly along the ray. This is the most likely explanation why ray pointing has been identified as inferior to other pointing methods in small spaces, such as *desktop* environments, e.g., [39].

2.3 Un-instrumented Pointing

Un-instrumented pointing has been studied in the past. Here, the system tracks the hand or finger(s) of the user without instrumentation on the user. Typically this is done with some form of cameras. This body of work also explores gestures as a method of interaction. Yet, ray pointing with the finger or arm is typically used as the main pointing method even in gesture-based systems. Work by Gallo et al. [10] explored the benefits of an un-instrumented hand tracking device in a medical context, where sterility is a major concern. Here contact-less technologies offer clear benefits. Devices that do not require contact with the surgeon's hands, such as the Microsoft Xbox Kinect™, help avoid contamination. In a paper by Kolarié et al., finger pointing was enabled through a two camera stereo setup [20]. In this system the hand was tracked based on a database of hand positions and human skin colour. Matikainen et al. tracked multiple users and their pointing gestures with a camera system [26]. In work on "Bare-Hand Human-Computer Interaction" [13], the authors look at the requirements of an un-instrumented tracking system and present an implementation similar to previous work. The authors developed three sample applications to demonstrate its use: FingerMouse, FreeHandPresent, and Brainstorm. Most relevant to my context is the FingerMouse application, which tracks a user's index finger to position a mouse pointer on screen. In this application, a one second dwell time is used for selection [13]. Song et al. also used finger pointing to select and move virtual objects [35].

There has been research into the Leap Motion tracking device that I used in my experiments. Weichert et al. looked into the accuracy and robustness of the Leap Motion in real environments [42]. A 0.2mm deviation was found for static positions and a deviation of 1.2mm for dynamic motions. These accuracy results are substantially better than results for similarly priced systems, such as the Microsoft Xbox Kinect™. The Leap Motion's performance was also considered in a medical environment by Ogura et al. [30]. In this research, a gesture system was developed to control the manipulation of angiographic images. After 30 minutes of practice technologists were able to perform common commands more quickly with the developed gesture system compared to the mouse.

None of the above work evaluates the performance of un-instrumented in-air pointing with the ISO standard.

2.4 Digital Jewellery

Last, but not least, there has been research into the design and use of wearable digital jewellery, specifically rings. This research looks at creating a device that not only functions well, but that is also comfortable and attractive to the user. Many of these rings are made of conventional jewellery materials [3,27], but some are made of more unconventional materials such as elastic [33] and Velcro [14]. These unconventional materials permit users with different finger sizes to use the same ring. Such rings can be used for 3D selection, in particular for indicating which object to select.

2.5 Input Method Reliability

There are a number of in-air hand gesture recognition systems that have been proposed, Most of these systems achieve recognition reliabilities of 80% or higher. Kang et al. looked at using a gesture spotting system to control a video game [17]. Their system achieved a reliability level of 93.36% for 10 gestures. Lee and Kim propose a gesture spotting method based on a Hidden Markov model (a statistical modeling approach) [22][18]. This model was adapted to include a threshold value. The adapted model achieved a reliability of 93.14% - 93.38%. Hand gesture recognition with the Hidden Markov model was also studied by Yoon et al. [46]. They found a reliability level of 93% for batch test results and a success rate of 85% for real time gesture recognition. Other research looked at gesture recognition with a Wii controller [34] again based on a Hidden Markov model. In this research, five different gestures were analyzed and reliability was found to be between 85% and 95%. Liang and Ouhyoung used a Hidden Markov model and a DataGlove™ to develop a gesture recognition system for Taiwanese Sign Language [23]. The real-time average reliability rate for this system is 80.4%. The best recognition models for head and eye gestures were presented by Morency et al. [28]. In their work, they developed a new recognition system and compared it against leading recognition models. In their experiments their developed system achieved the highest reliability at 85.1%. Other methods had accuracies as low as 50.5%.

The effect of reliability on user performance has also been investigated. In the work by Arif and Stuerzlinger the performance cost of errors was analyzed in the context of text entry [1]. They found that a system reliability in the range of 99% - 98% has a small negative effect on performance (about 7% to 8%). However, systems that are 95% reliable cause performance to drop by 26%. In other work by the same authors, a faulty unistroke based text entry method was investigated [2]. There, they looked at user adaptation to the system. All users were able to identify the most error prone character (which had a reliability of 90%) and users adapted to alternative methods much faster if the character was less reliable.

2.6 Other Related Work

Kitamura et al. conducted research on a virtual chopstick interface for object manipulation [19]. In their work, they describe a virtual reality system where objects can be picked up and manipulated by way of virtual chopsticks. Users hold their hands as if they were holding actual chopsticks and imitate the motions they would perform with real chopsticks in order to control the virtual chopsticks. In this work, two virtual chopsticks are used for their grabbing capabilities, whereas in my work, a single chopstick is used for pointing.

Chapter 3
Examining Un-Instrumented In-Air Pointing

This next section will detail the experiments and pilot studies conducted to measure and classify the various aspects of un-instrumented in-air pointing.

3.1 Experiment 1

The objective of this study was to determine the interaction method that yields the best throughput for in-air operation. The two compared interaction methods were the *Pointing* interaction method and the *Whole Hand* method (see the section on Interaction Techniques for details). These two interaction styles were tested both with the user's elbow resting on the table and with the user's elbow in the air. When using a mouse, one's hand benefits from the natural stabilization afforded by resting the arm on the desk. I wanted to see if there would be a performance benefit if users grounded their elbows on the table thereby stabilizing their hands and fingers so that they would be less prone to natural hand tremor. On the other hand, grounding the elbow has the potential for restricting the movement of users and thus may also decrease performance from a participant's arm being more restricted in movement by being confined to the surface of the desk. This work is similar to the work by Cockburn et al. [8] which investigated this in a scenario where the user stands. I look at these interaction methods in a desktop environment where participants are seated and the range of motion required was much smaller than in Cockburn's work. The range of motion in my experiment was at a level comparable to the range of motion required by a mouse.

3.1.1 Participants

I recruited 16 participants for this study (mean age 23 years, *SD* 8.5). Seven were female and one was left-handed. None had used in-air devices before to interact with a computer. All users kept their hands in an open relaxed position when using the *Whole Hand* interaction method.

3.1.2 Setup

The Leap Motion sensor was placed directly in front of computer display so that it was centered with the middle of the monitor. The sensor was then calibrated to the screen with the default calibration process, which uses a wooden chopstick for a more precise calibration. The Leap Motion device driver and hardware used for this first study were versions 0.8.0 and v.05, respectively. The software used for this Fitts' Law study was FittsStudy [45]. I added support to read data from the LeapMotion to this package.

3.1.3 Interaction Techniques

I studied two types of interaction techniques in this work (see Figure 3-1). The first method is the *Whole Hand* method. With this method the user moves her or his dominant hand in space to indicate the area on the screen the cursor should move to. For example, if the user held her or his hand at the bottom left area of the screen the cursor would appear there, if she or he then began moving her or his hand up then the cursor would gradually move with the hand towards the upper left area of the screen. The second is the *Pointing* method, based on ray casting. With this method the user uses her or his index finger of the dominant hand to point to the relative location on the screen that she or he wishes the cursor to move to.

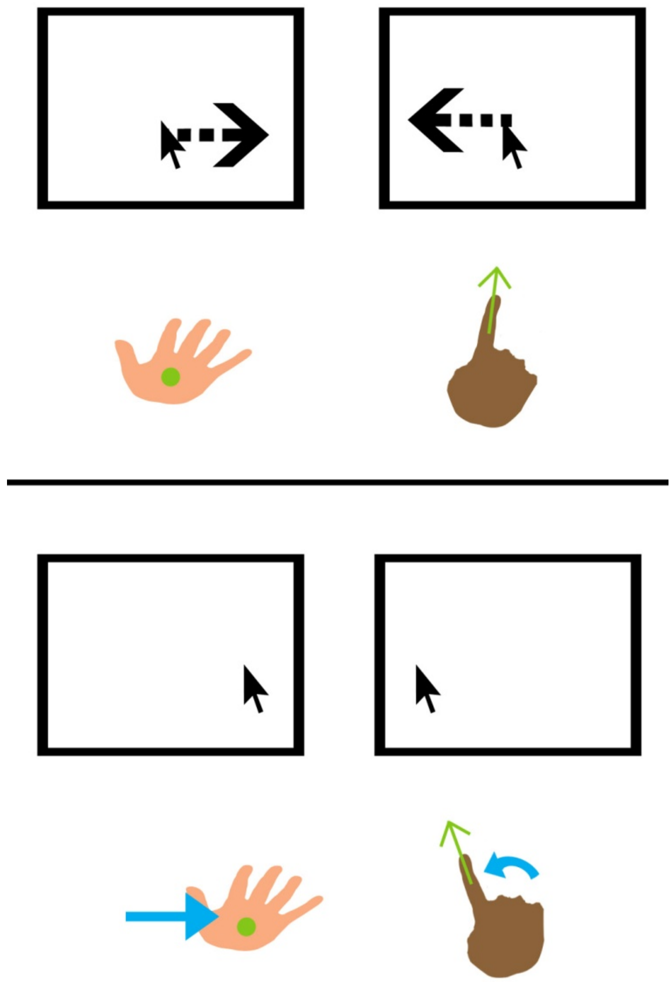

Figure 3-1: The two interaction methods. The *Whole Hand* method is pictured on the left and the *Pointing* method on the right. The top frame indicates the cursor position before and the frame below depicts the cursor position after the movement. The green dot and green arrow show what is being tracked by the system.

3.1.4 Input Conditions

For this experiment there were four input conditions for selecting targets. These were the *Whole Hand* method with the participant's elbow supported by the table or by a stack of books (depending on what was more comfortable), the *Whole Hand* method with the participant's elbow raised above the table, i.e., in the air and not supported, the *Pointing* method with the elbow supported, and the *Pointing* method with the elbow unsupported. After targets had been acquired using one of these four methods, targets were selected using the spacebar on a keyboard. The spacebar was operated by the non-dominant hand of the participant and was placed in a comfortable operating position so that the dominant hand used for object acquisition was not obstructed. Figure 3-2 illustrates the setup. In all these conditions the distance between the participant's hand and the computer was relatively consistent, but an exact distance was not enforced to prevent unnatural poses.

Figure 3-2: Photo of experimental setup in the condition where users were pointing with the finger and where the elbow was off the table. This user operates the space bar with the left thumb.

3.1.5 Procedure

First, each participant was given a brief questionnaire about her or his background. The questionnaire recorded gender, age, and handedness. Then, the participant was instructed in the use of one of the four input conditions and was encouraged to practice with this input method until she or he felt comfortable. The order of presentation of input conditions was determined with a Latin square design across all participants. Once the participant was comfortable with the input method, she or he completed a series of Fitts' law selection tasks using her or his dominant hand to move the cursor to the desired location and operating the spacebar with the other hand for selection. The participant was instructed to select these targets as quickly and accurately as possible. She or he was also instructed that breaks should be taken between circle groups if her or his arm was getting tired. Each such block consisted of 9 Fitts' law "circles", with 13 trials per circle. The 9 circles used all combinations of target widths of 32, 64, and 96 and amplitudes of 256, 384, and 512 pixels, respectively. This range of ID values was chosen, as target widths below 32 pixels would often be missed, not due to participant error, but due to spatial jitter in the tracking system. This amount of jitter was dependent on the distance of the hand to the screen. I wanted to get a good measurement of the achievable performance with a well-operating system that had the monitor at a typical distance and thus did not include these targets. Participants were free to move their arm around while using the *Pointing* method and, as such, movement angles and distances are potentially not consistent across all participants. Consequently, I report ID values in pixels. The participant would then be presented with another “block” for the next input method and the above process would be repeated until all four input conditions had been completed. Overall, the study took about one hour per participant.

3.1.6 Results

Data was first filtered for participant errors, such as hitting the spacebar twice on the same target or pausing in the middle of a circle to focus her or his attention elsewhere. Removing these errors and outliers, i.e., results more than three standard deviations from the mean, amounted to a 3% loss of total data collected.

Throughput

The data was not normally distributed. Also, Levene’s test (See Appendix A) for homogeneity of variance showed no significance for elbow placement ($F_{1,15} = 0.4332$, ns), but movement type and the interaction between movement type and elbow placement were statistically significant ($F_{1,15} = 57.708$, $p < .001$ and $F_{3,13} = 27.724$, $p < .001$). This invalidates the assumption of similar differences between groups variances needed for parametric repeated measures ANOVA. To address these concerns, I used the *Aligned Rank Transform (ART)* for nonparametric factorial data analysis in Human-Computer Interaction treatment (See Appendix B) [44].

Overall, there was no significant effect for movement type ($F_{1,15} = 0.52$, ns) or for elbow placement ($F_{1,15} = 0.67$, ns) on (effective) throughput (see Figure 3-3). There was also no significant effect from the interaction of movement type and elbow placement ($F_{1,15} = 0.44$, ns).

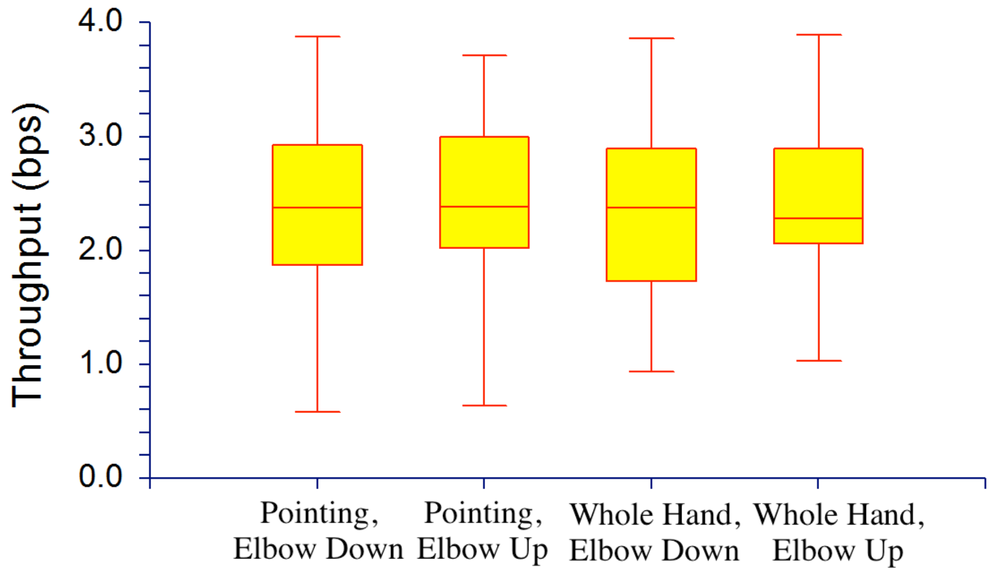

Figure 3-3: Box plot depicting the average throughput (bps) for each condition. The difference between methods was not significant.

Movement Time

The data was not normally distributed and also failed Levene's test for homogeneity. ART was used again to address this concern.

There was no significant effect on movement time for both movement type ($F_{1,15} = 1.42$, $p > .05$) and elbow placement ($F_{1,15} = 0.92$, ns). There was also no significant effect for the interaction between movement type and elbow placement ($F_{1,15} = 0.04$, ns). See Figure 3-4 for mean movement times.

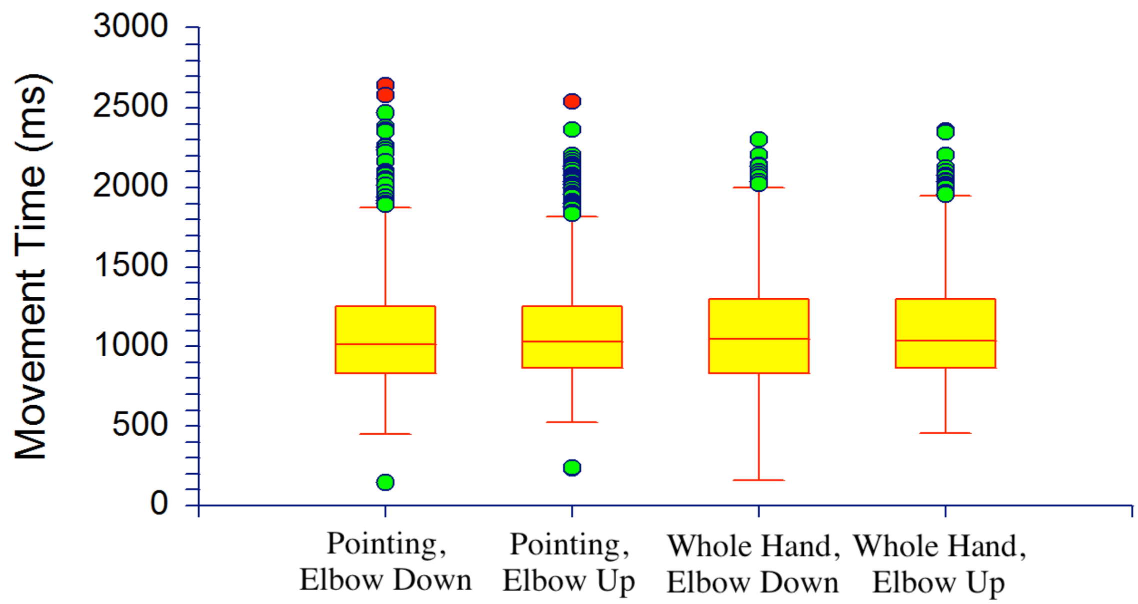

Figure 3-4: Box plot depicting the average movement time (ms) for each condition. The difference between methods was not significant. The large number of outlier points is due to skewed data.

Error Rate

The data was not normally distributed and also failed Levene's test for homogeneity. ART was used again to address this concern.

The placement of a participant's elbow had no significant effect on error rate ($F_{1,15}$ = 0.29, ns), but the movement type did identify a significant difference ($F_{1,15}$ = 4.93, $p < .05$) with the *Whole Hand* method producing fewer errors. Although technically significant, the statistical power of this result is rather low at 0.55, so I cannot claim this to be a strong result.

Index of Difficulty (ID)

The data was not normally distributed and also failed Levene's test for homogeneity. ART was used again to address this concern.

There was no significant effect on the interaction between ID and movement type ($F_{1,15} = 1.22$, $p > .05$) or ID and elbow placement ($F_{1,15} = 0.23$, ns) on movement time. See Figure 3-5 for the data for all conditions. The equations for the Fitts' law trend lines are as follows: *Whole Hand* elbow down: $y = 302.36\,x + 151.97$, $R^2 = 0.986$, *Whole Hand* elbow up: $y = 295.87\,x + 156$, $R^2 = 0.9918$, *Pointing* elbow up: $y = 273.15\,x + 198.42$, $R^2 = 0.9976$, *Pointing* elbow down: $y = 283.81\,x + 190.5$, $R^2 = 0.9742$.

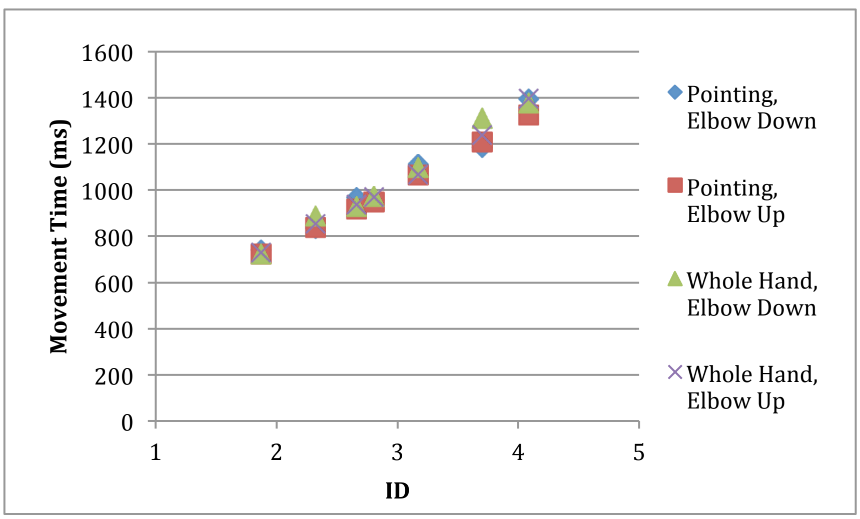

Figure 3-5: Graph depicting Fitts' law models for input conditions in experiment 1.

Learning

As there was only one block per input condition for this experiment it is unclear how strong learning was. Over all participants, no significant learning effects could be detected, but this does not mean that users did not learn. For more information about learning effects with in-air operation see experiment 2.

Other Results

I also analyzed in which condition each *individual* participant showed the highest throughput. I classified throughput as being equal if the two throughput values were within 5%, otherwise as different. The results of this analysis identify that the similar results seen for throughput between each method are not due to individual participants performing equally well with each method. Rather, there seems to be a half-half split among the population, both for the question as to which input method is better and if it is better if the elbow is placed on the table or not (see Figure 3-6 and Figure 3-7 respectively).

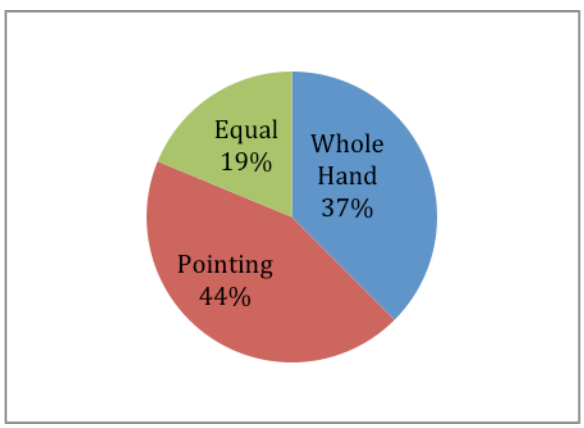

Figure 3-6: Chart displaying the percentage of participants that demonstrated the highest throughput with the different interaction methods.

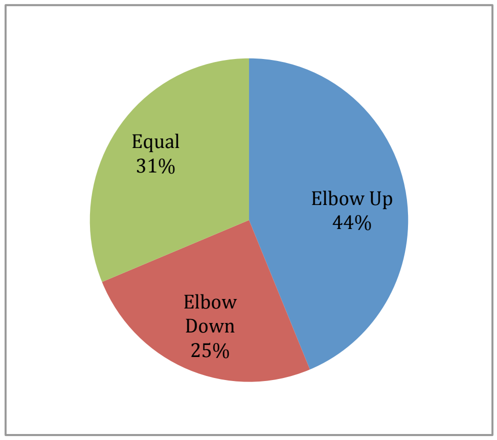

Figure 3-7: Chart displaying the percentage of participants that demonstrated the highest throughput with the different elbow placements.

3.1.7 Discussion

The overall conclusion from this study is that there seems to be no particular method of using in-air devices that works best for all people. As such, I believe that the differences between these methods are not a major contributing factor to lower in-air throughput. Statistical significance between movement types was only found for error rate, but even this significance had low statistical power. These results are in line with the results found by Cockburn et al. and show that their insights hold even in desktop environments and with modern desktop in-air pointing sensors. Performance for each method was heavily dependent on the individual participant. Some participants exhibited no difference between the methods, while others found that a particular method yielded substantially better results. As such, there seems to be no universal solution that exhibits uniformly high performance. This indicated to us that in-air systems should be configurable to account for different input methods.

The throughput observed for in-air interaction in this study was at best about 2.8 bps, which is lower than the usually observed range of 3.7–4.9 bps for the mouse in Fitts' law studies [36]. With 85 ms end-to-end latency in whole the system, the finger tracking system ranks comparable to other 3D tracking systems in terms of delay (though there are better ones). Even so, 85 ms of latency is much higher than the measured 28 ms of end-to-end latency for the mouse in my setup. While this difference of 57 ms in latency can be expected to have an impact on the performance of in-air interaction, it should decrease throughput by only about 0.3 bps according to previous work on the effects of latency [32]. If this latency were eliminated it would bring the Leap Motion's throughput to about 3.1 bps. This is still not even close to levels that can be easily achieved with a mouse.

3.2 Experiment 2

The purpose of this study was to directly compare in-air interaction with the mouse. The two input methods were compared using the ISO 9241-9 test procedure using both the spacebar and a button click for selection. Both spacebar and button were used in order to determine the effect different click detection methods have on performance. Specifically, I wanted to identify if clicking the left button on the mouse affords an advantage in terms of throughput. The purpose of this study was not to suggest in-air interaction as a mouse replacement, but rather to provide a comparative measurement to the mouse that could be used to calibrate the performance of this system against other systems. This enables comparisons to other hand tracking devices and other input methods.

3.2.1 Participants

I recruited 16 *different* participants for this within subjects study (mean age 27 years, *SD* 9.5). Six were female and one was left-handed. That participant still preferred to operate the mouse and to point with the right hand. None had used an in-air system before. All users kept their hands in an open relaxed position when using the *Whole Hand* interaction method.

3.2.2 Setup

The sensor was placed directly in front of computer display so that it was centered at the middle of the monitor and carefully calibrated. The Leap Motion software and hardware versions used were 1.0.5+7357 and LM-010, respectively (a more up-to-date version relative to experiment 1). The measured end-to-end latency of the overall system was 63 ms. The mouse was a Microsoft IntelliMouse Optical set to the default pointer speed in the Windows 7 operating system. The system had an end-to-end latency of 28 ms with the mouse.

I developed a ring button as an alternative method to indicate object selection in the in-air condition. I wanted this button to mimic the left click on the mouse as close as reasonably possible. The ring button consisted of a button glued to a *Hook and Loop* strip. I used a *Hook and Loop* strip to accommodate participants with diverse finger sizes, much as Harrison and Hudson did with their Velcro ring [14]. The button, an Omron B3F-1020 tactile switch, is a 6×6 mm square box, 5 mm tall with a 3.5 mm cylindrical tip. The operating force is specified at 0.98 N and the tip travel during a click is about 0.5 mm. The *Hook and Loop* strip was 20 mm wide and 100 mm long and was glued to the button with a hot-melt adhesive. The button was wired in parallel to the left-button of a desktop mouse. The thin wire connecting the ring to the mouse was held on participants' arms with two 50 mm bands of 3M™ Coban™ Self-Adherent tape in order to prevent the wire from interfering with the arm motions and from possibly confounding the sensor's view. Figure 6 illustrates the ring button, its placement as well as the wire placement. The software used for conducting the Fitts' law tasks was again FittsStudy [45].

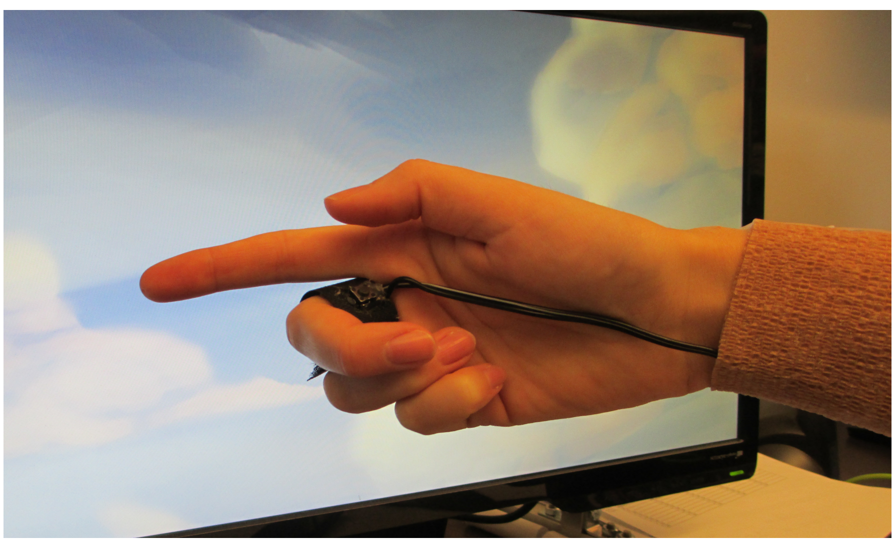

Figure 3-8: Photo of finger ring with the button visible below the users thumb (raised more than normal for illustration). Self adhesive tape was used to keep the cable from interfering with user motions.

3.2.3 Input Conditions

For this experiment there were four input conditions that the participants selected targets with. These were in-air with spacebar selection (using whatever in-air technique the participant preferred), in-air with ring button selection (using the same in-air technique chosen), the mouse with spacebar selection, and the mouse with left mouse button selection. For selection with the spacebar, the keyboard was operated and placed as described above in experiment 1.

3.2.4 Procedure

First, each participant was given a brief questionnaire about her or his background. The questionnaire recorded gender, age, and handedness. Then, the participant was introduced to the two different interaction styles for in-air operation, the *Whole Hand* method and the *Pointing* method, as described above under Interaction Techniques. After trying both methods, participants were asked to choose which method they wanted to use. That method was then used for the rest of the experiment. I permitted participants to choose their own in-air method based on the results of my first experiment, where different people performed better with different interaction methods. Keeping with the work by Sparrow and Newell [37], I assumed that participants would choose the in-air style that would yield the best throughput through self-optimization. For the same reason, I also did not control for user elbow placement in this study.

Once a particular input method was chosen, participants completed 2 blocks of 9 Fitts' law circles with 11 trials per circle as practice with this method. This practice period accounted for the inexperience of participants with in-air interaction. My study design thus also analyzes the performance possible in the early adaptation stages of in-air interaction. The performance in these early stages is frequently just as important as potential top performance: if performance is much lower than an alternative method, users will often just give up and use that alternative method, even if their performance could ultimately be better by adapting the higher performing method. One well-known example of this is QWERTY vs. Metropolis for touchscreen keyboards [47].

After the practice period, the participant started with one of the four conditions, and then experienced the others. The presentation order to participants was determined with a Latin square design.

With each given condition the participant completed 3 blocks of 9 Fitts' law circles with 11 trials per circle. Target widths of 32, 64, and 96 and amplitudes of 256, 384, and 512 pixels were used, the same as for experiment 1. Between blocks, participants were encouraged to rest for about a minute before starting the next one. Participants were also instructed that breaks should be taken between "circles" if they experienced fatigue. At the end of all four conditions, participants were given a brief questionnaire about arm strain they might have experienced while using in-air interaction.

3.2.5 Results

Data was first filtered for participant errors, such as hitting the spacebar twice on the same target or pausing in the middle of a circle. Removing these errors amounted to less than .005% loss of total data collected. There were no outliers.

Throughput

The data for throughput was not normally distributed. Also, the data failed Levene's test for homogeneity. Consequently, I again used the *Aligned Rank Transform (ART)* for nonparametric factorial data analysis [44] and performed a repeated measures parametric ANOVA on the transformed data.

For throughput there was a significant effect of type of device of ($F_{1,15} = 90.76$, $p < 0.0001$), with effect size (η^2) of 0.3 and power ($1-\beta$) of 0.99. There was no significant effect of selection method on throughput ($F_{1,15} = 0.47$, ns). There was a significant interaction of selection method and type of device ($F_{1,15} = 17.10$, $p < 0.001$), with an effect size (η^2) of only 0.03 and very weak power ($1-\beta$) of 0.07. The mouse exhibited throughput around 4 bps, while the in-air conditions showed less than 3 bps, see Figure 3-9.

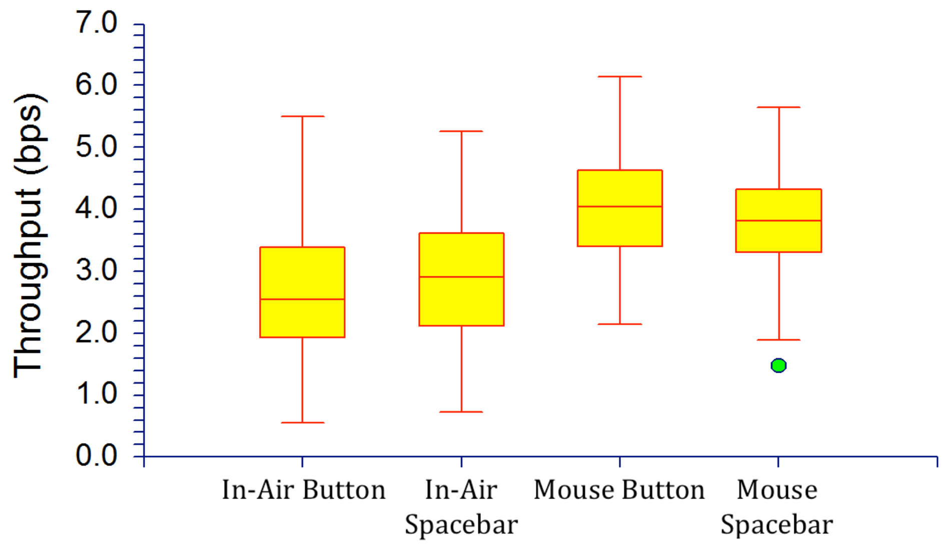

Figure 3-9: Box plot depicting the average throughput (bps) for each condition in the second study. The difference between methods was significant.

Movement Time

The data for movement time was not normally distributed and also failed Levene's tests for homogeneity. Consequently, I used *ART* again.

For Type of device there was a significant effect on MT_e ($F_{1,15} = 49.66$, $p < 0.0001$), with effect size (η^2) of 0.23, and power $(1-\beta) = 0.96$. There was no significant effect of selection method on movement time ($F_{1,15} = 0.8$, *ns*). Similar to the results for throughput, there was a significant effect for the interaction of Selection Method and Type of device on MT_e ($F_{1,15} = 14.37$, $p < 0.05$), with an effect size (η^2) of only 0.03, and with very weak power $(1-\beta) = 0.065$. See Figure 3-11 and Figure 3-10 for mean movement times.

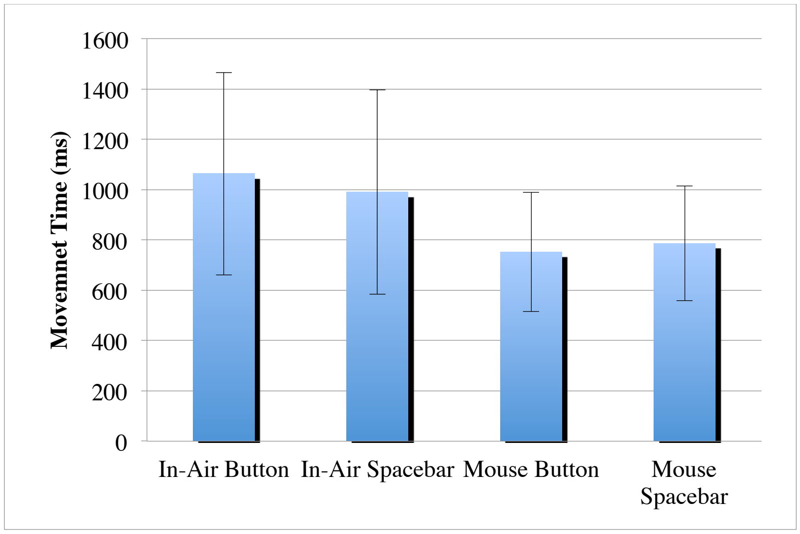

Figure 3-10: Graph depicting the average movement time (milliseconds) for each condition in the second study with error bars. The difference between methods was significant. Error bars show standard deviation.

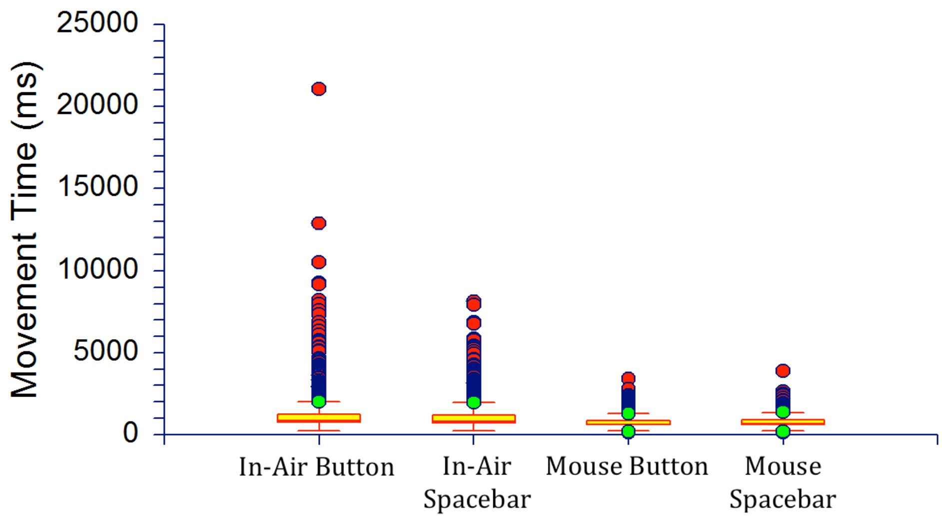

Figure 3-11: Box plot depicting the average movement time (milliseconds) for each condition in the second study. The difference between methods was significant. The large number of outlier points is due to skew.

Error Rates

I did not find any significant effects in the data for error rates.

Learning

The data was not normally distributed and also failed Levene's test for homogeneity. ART was used again to address this concern.

There was no statistically significant difference between blocks on throughput ($F_{1,15}$ = 0.49, ns). There was also no statistically significant difference for the interaction of block and selection method on throughput ($F_{1,15}$ = 0.6, ns).

Index of Difficulty (ID)

The data was not normally distributed and also failed Levene's test for homogeneity. ART was again used to address this concern.

There was no significant effect on the interaction between *ID* and selection method ($F_{1,15} = 2.67$, $p > .05$). See Figure 3-12 for a depiction of the Fitts' law models for the data for this study.

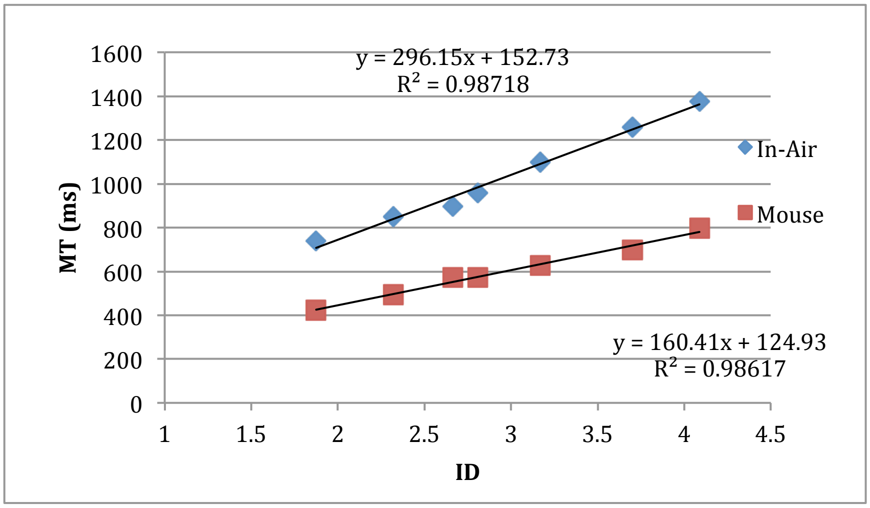

Figure 3-12: Graph depicting Fitts' law model for in-air interaction in experiment 2 and a linear regression.

3.2.6 Discussion

The results of the second experiment show that the mouse performs significantly better than in-air interaction, regardless of the target selection method used. The performance of the mouse is around 4 bps, which is in line with other work. The performance of in-air conditions is slightly less than 3 bps, which corresponds (approximately) to my first study and to the results of other recent evaluations of ray pointing techniques [40] with traditional 3D tracking systems. A caveat is that these results are not fully comparable, as my work did not use a stereo display.

One result of this study is that even in a condition designed to be as optimal for in-air interaction as possible, the mouse still outperforms this method by a substantial margin. Moreover, and in all conditions, the selection mechanism (spacebar or ring button) was 100% reliable. Consequently, I am fairly confident in stating that using finger gestures, such as “up-down” click motions, would not improve throughput. The reasoning here is that any inaccuracies in gesture detection will very likely have a negative effect on error rates, as un- or misrecognized button presses will result either in a “miss” or correction movement. Both alternatives will decrease throughput. Moreover, and even assuming that there is a 100% reliable gesture recognizer, I point out that to make an “up-down” click gesture in free air easily recognizable, the user needs to move the finger a distance that is sufficiently large to be detectable. This distance is likely larger than the small motion needed to operate a mouse or ring button. Consequently, such a motion likely takes longer, which again can only decrease throughput. Other gestures, such as a pinch, making a fist, and closing the gap between the thumb and the side of the hand, suffer from the same problem and are also not likely to exhibit increased throughput.

Moreover, all participants in the study reported some level of arm strain while using the in-air system regardless of whether they were using the *Pointing* or the *Whole Hand* method. All participants reported either ‘mild discomfort’ or ‘discomfort’ (values 2 and 3 respectively) on a five-point scale from ‘none’ to ‘pain’ with an average reported value of 2.5. There was no correlation between interaction method used and level of arm strain.

To get a better idea of the performance potential of in-air interaction, one would need to run a longitudinal study, where participants get several hours of practice.

3.3 Pilot Study 1

This study was to determine if using a chopstick instead of a participant's finger would improve the Leap Motion's throughput and bring it closer to the values traditionally seen with a mouse. I hypothesized that using a chopstick would increase throughput because I reasoned that the chopstick could be held more stably in a pencil grip, i.e., between three fingers like a pen, compared to just a single finger. The chopstick also provides an easier shape to track compared to a human finger, as it is a rigid body, fairly long, and perfectly cylindrical. The two interaction methods that were compared were the *Chopstick* interaction method and the *Mouse* method.

3.3.1 Participants

I recruited 6 participants for this study (mean age 27 years, SD 12.9 years). One participant was male and all but one were right handed. The left-handed participant preferred right-handed mouse operation.

3.3.2 Setup

The Leap Motion sensor was attached to the computer used for the experiment by a USB2 cable and was placed directly in front of computer display so that it was centered with the middle of the monitor. The diagnostic Leap Motion Visualizer was then used to detect and fix any possible interference with the Leap Motion system. The Leap Motion software used for this first study was version 1.0.9+8410 and the Leap Motion hardware device was LM-010. The end-to-end latency in this system was 63 ms. The software used for this Fitts' Law study was FittsStudy, which is available online [45]. I only added support to read data from the LeapMotion to this package.

The mouse used was a Microsoft IntelliMouse Optical set to the default pointer speed on the Windows 7 operating system. The mouse had an end-to-end latency of 41 ms.

3.3.3 Input Conditions

For this experiment there were two input conditions for selecting targets for the participants to use. These were the *Chopstick* method and the *Mouse*. The *Chopstick* method required the user to hold a standard disposable wooden chopstick in her or his dominant hand as she or he would normally hold a pencil. Targets were then selected by aiming the tip of the chopstick toward the target on the screen. The *Mouse* method required the user to operate a computer mouse as they normally would. After targets had been acquired using one of these two methods, targets were selected using the left click button in the *Mouse* method and the spacebar on the keyboard in the *Chopstick* method. The spacebar was operated by the non-dominant hand of the participant and was placed in a comfortable operating position so that the dominant hand used for object acquisition was not obstructed. Figure 3-13 illustrates the setup.

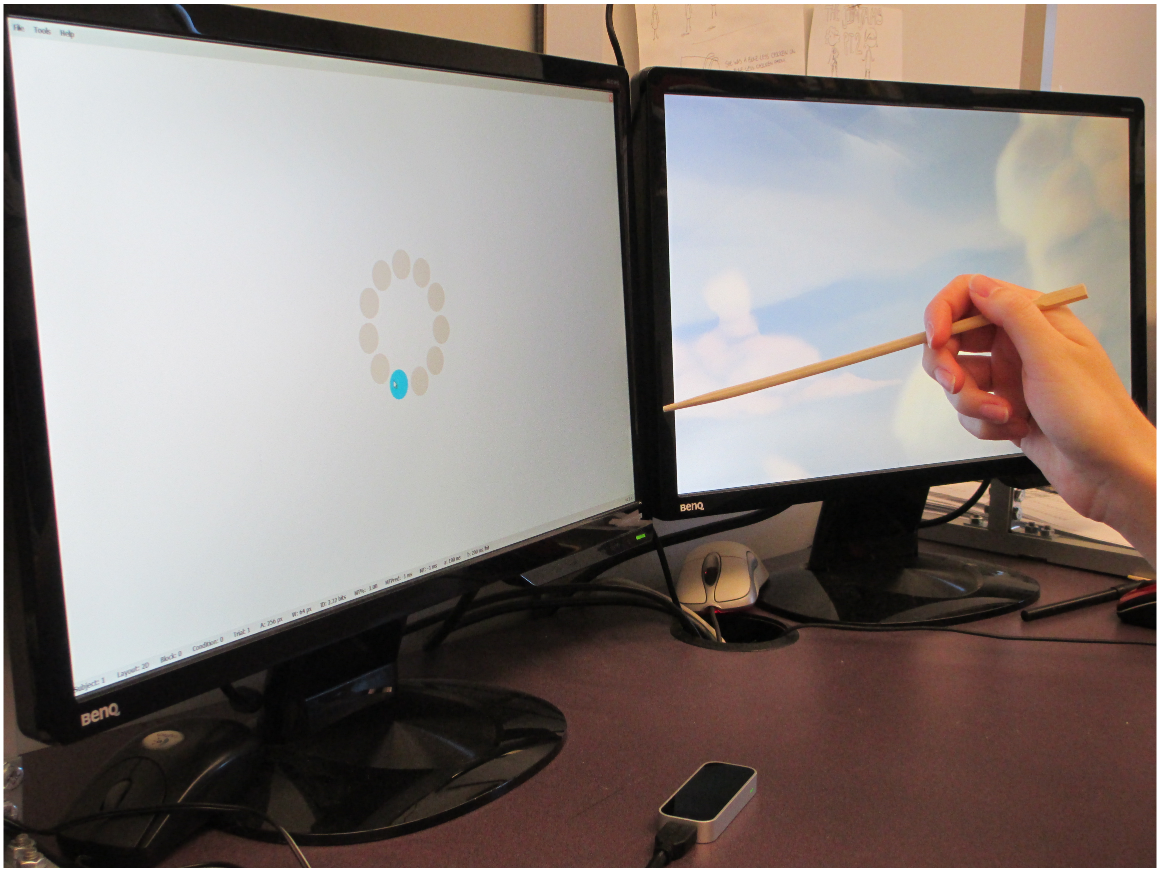

Figure 3-13: The setup of pilot study 1 and pilot study 2.

3.3.4 Procedure

First, the participant was given a brief questionnaire about her or his background. The questionnaire recorded gender, age, and handedness. Then, the participant was introduced to the *Chopstick* condition and shown how it worked. After she or he was comfortable with the basic operation, one of the input conditions was explained to the participant. The order that participants were given each of the input methods was counterbalanced so that each of the possible orders was represented equally.

Once the participant was comfortable with the input method, she or he completed a series of Fitts' law selection tasks using either the mouse or the chopstick in her or his dominant hand. Ten blocks of 9 Fitts' law conditions with 11 trials per condition were completed. Target widths of 32, 64, and 96 and target amplitudes of 256, 384, and 512 were used. A total of 990 trials were completed for each condition. The participant would then be presented with the other input method and the above process would be repeated. At the end of both conditions, participants were given a brief questionnaire about any discomfort they might have experienced while using un-instrumented tracking and the mouse.

3.3.5 Results

Data was first filtered for participant errors, such as hitting the spacebar twice on the same target or pausing in the middle of a circle. Removing these errors amounted to less than .006% loss of total data collected.

Throughput

The data was not normally distributed. Also, the data failed Levene's test for homogeneity. This invalidates the assumption of similar differences between groups variances needed for parametric repeated measures ANOVA. To address these concerns, I used the *Aligned Rank Transform (ART)* for nonparametric factorial data analysis in Human-Computer Interaction treatment presented by Wobbrock et al. [44].

There was a significant effect for device used ($F_{1,5} = 1748$, $p < .001$) with a power ($1 - \beta$) of .99 and a large effect size (η^2) of .6. See Figure 3-14 for average throughput values (3.26 bps for chopstick and 4.01 bps for mouse).

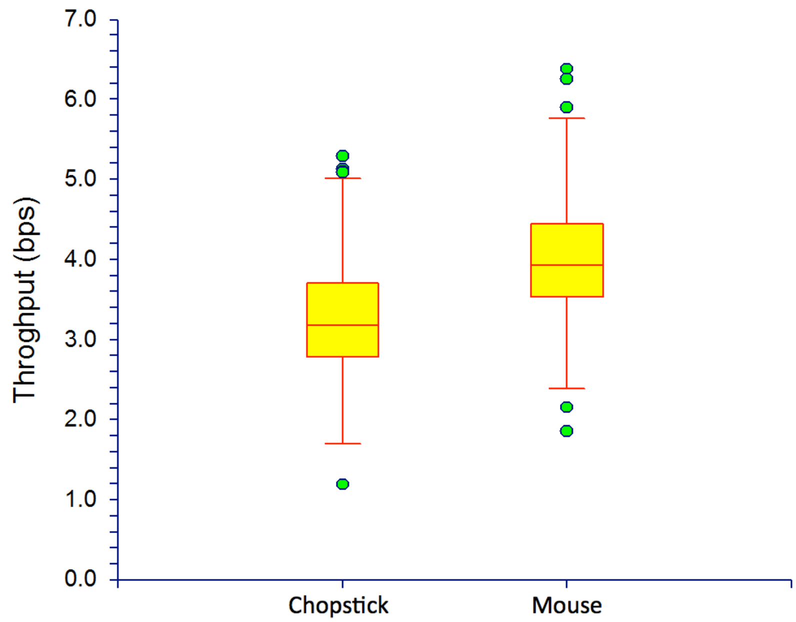

Figure 3-14: Box plot of average throughput values (bps) for Chopstick and Mouse. The difference was significant.

Movement Time

The data for movement time was not normally distributed and also, failed Levene's tests for homogeneity. Consequently, I used *ART* again.

There was a significant effect for device used ($F_{1,5} = 334$, $p < .001$) with a power ($1 - \beta$) of 0.99 and a small effect size (η^2) of .08. See Figure 3-15 for average movement times.

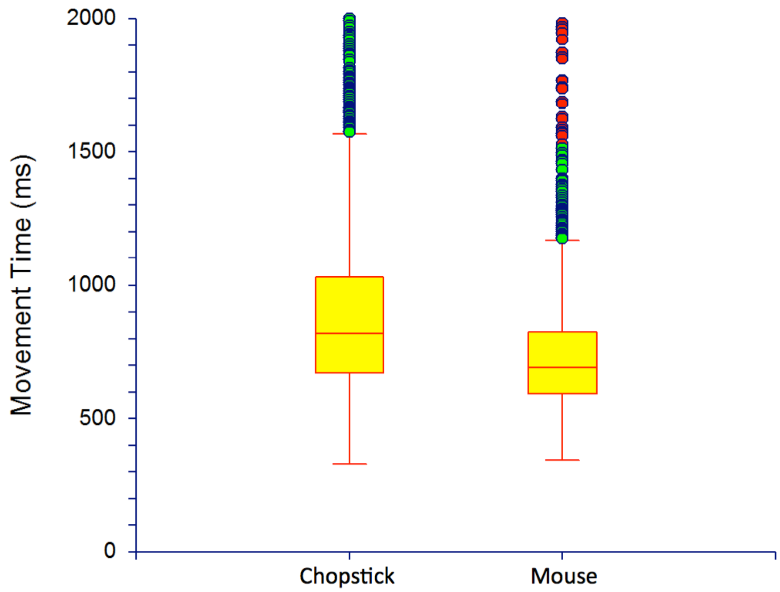

Figure 3-15: Box plot of average movement times (ms) for the Chopstick and Mouse. The difference was significant. The large number of outliers is due to skewed data.

Error Rate

The data was not normally distributed and also failed Levene's test for homogeneity. ART was used again to address this concern.

Device used (chopstick or mouse) had no significant effect on error rate ($F_{1,5}$ = 0.17, ns).

Learning

The data was not normally distributed and also failed Levene's test for homogeneity. ART was used again to address this concern.

There was a statistically significant learning affect observed over all data ($F_{1,5}$ = 10349, $p < .001$) with a power ($1 - \beta$) of .99 and an effect size (η^2) of .08. There was no significant difference in the learning curve between devices ($F_{1,5}$ = 0.48, ns). See Figure 3-16 for a graph of learning over time.

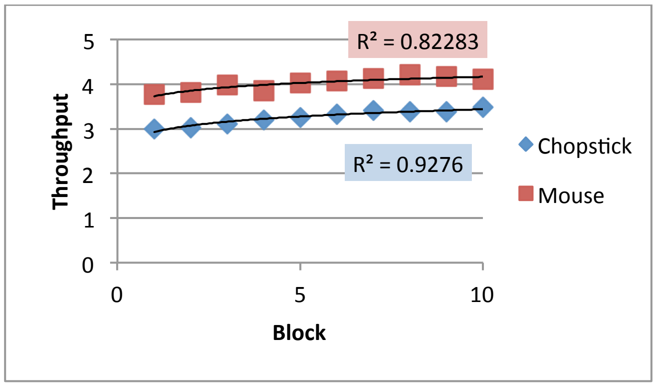

Figure 3-16: Graph of average throughput for each device for each block of circles with power curves fitted to the data.

Index of Difficulty (ID)

The data was not normally distributed and also failed Levene's test for homogeneity. ART was used again to address this concern.

Device used (chopstick or mouse) crossed with ID value had no significant effect on throughput ($F_{1,5}$ = 0.35, ns).

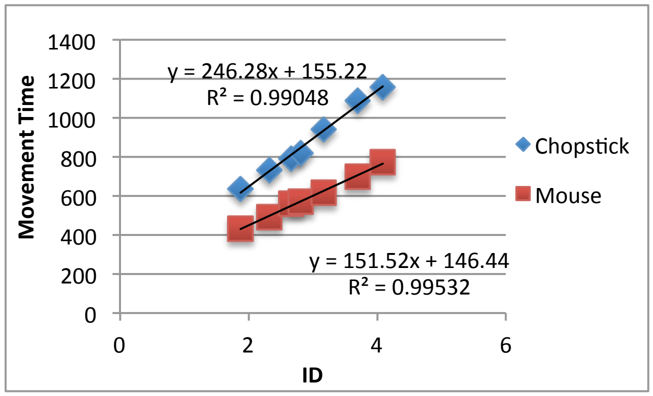

Figure 3-17: Graph of the average movement times for each ID value for both the chopstick and the mouse. Linear trendlines and their equations are displayed.

3.3.6 Discussion

While the use of a chopstick substantially improves pointing performance relative to the results found above for finger pointing, it is not enough to bring un-instrumented pointing to the levels seen with the mouse [7]. The chopstick is still only at 3.49 bps by block ten which is more than .5 bps less than the mouse. Still, the results for the chopstick appear to be far more promising than the results for finger pointing and the performance of this interaction method merits further exploration. A reduction in latency might improve the chopstick's performance enough to bring it to the throughput levels of the mouse.

3.4 Pilot Study 2

This study was to determine the effect a reduction in latency would have on throughput and if this modification would bring un-instrumented pointing to the throughput levels seen with the mouse. The design of this experiment was exactly the same as pilot study 1 except for the reduction in end-to-end latency. The two interaction methods that were compared were the *Chopstick* interaction method and the *Mouse* method.

3.4.1 Participants

I recruited 8 *different* participants for this study (mean age 21 years, SD 4.4 years). Two participants were male and all were right handed.

3.4.2 Setup

The computer that the experiment was run on had vsync turned off for this experiment to reduce display latency. The Leap Motion sensor was attached to the computer by a USB3 cable instead of the USB2 cable used in pilot study 1 to reduce the system latency even further. The Leap Motion sensor was placed underneath the participants' hands in a location that was most comfortable to them. The diagnostic Leap Motion Visualizer was then used to detect and fix any possible interference with the Leap Motion system. The Leap Motion software used for this pilot was version 1.0.9+8410 and the Leap Motion hardware device was LM-010. The system was measured to have an end-to-end latency of 48ms. The software used for this Fitts' Law study was FittsStudy, which is available online [45]. I added support to read data from the LeapMotion to this package.

The mouse used was a Microsoft IntelliMouse Optical set to the default pointer speed on the Windows 7 operating system. The system used with the mouse had an end-to-end latency of 32 ms.

3.4.3 Input Conditions

For this experiment the same exact input conditions as in pilot study 1 were used.

3.4.4 Procedure

The same exact procedure as in pilot study 1 was also used.

3.4.5 Results

Data was first filtered for participant errors, such as hitting the spacebar twice on the same target or pausing in the middle of a circle. Removing these errors amounted to less than .004% loss of total data collected.

Throughput

The data was not normally distributed. Also, the data failed Levene's test for homogeneity. This invalidates the assumption of similar differences between groups variances needed for parametric repeated measures ANOVA. To address these concerns, I used the *Aligned Rank Transform (ART)* for nonparametric factorial data analysis in Human-Computer Interaction treatment presented by Wobbrock et al. [44].

There was a significant effect for device used ($F_{1,5} = 21$, $p < .001$) with a power ($1 - \beta$) of .97 and a effect size (η^2) of .25. For graph of average throughput values see Figure 3-18 (3.54 bps for the chopstick and 4.13 bps for the mouse).

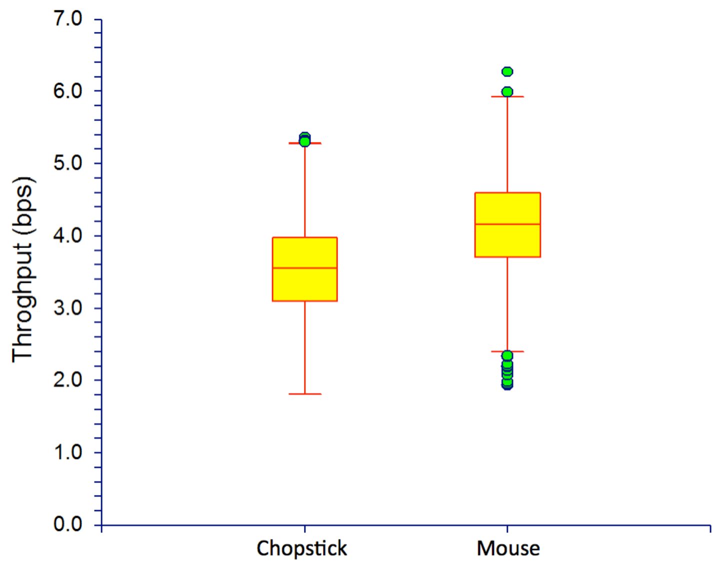

Figure 3-18: Box plot of average throughput (bps) for chopstick and mouse. The difference is statistically significant.

Movement Time

The data for movement time was not normally distributed and also, failed Levene's tests for homogeneity. Consequently, I used *ART* again.

There was a significant effect for device used ($F_{1,5} = 12$, $p < .001$) with a power ($1 - \beta$) of 0.86 and a very small effect size (η^2) of .05. See Figure 3-19 for average movement times.

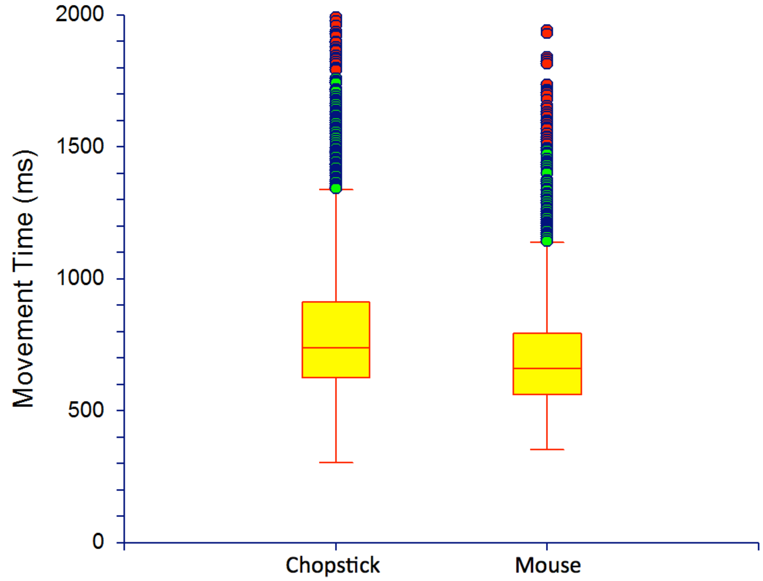

Figure 3-19: Box plot of average movement times (ms) for the chopstick and the mouse. The difference is statistically significant. The large number of outliers is due to skewed data.

Error Rate

The data was not normally distributed and also failed Levene's test for homogeneity. ART was used again to address this concern.

There was a significant effect for device used on error rate, with the mouse producing fewer errors, ($F_{1,5} = 8$, $p < .05$) with a power ($1 - \beta$) of .68 and a negligible effect size (η^2) of .01.

Learning

The data was not normally distributed and also failed Levene's test for homogeneity. ART was used again to address this concern.

There was no observed statistically significant learning affect over all data ($F_{1,5}$ = .03, ns) or in the learning curve between devices ($F_{1,5}$ = 0.7, ns). See Figure 3-20 for graph of learning effects.

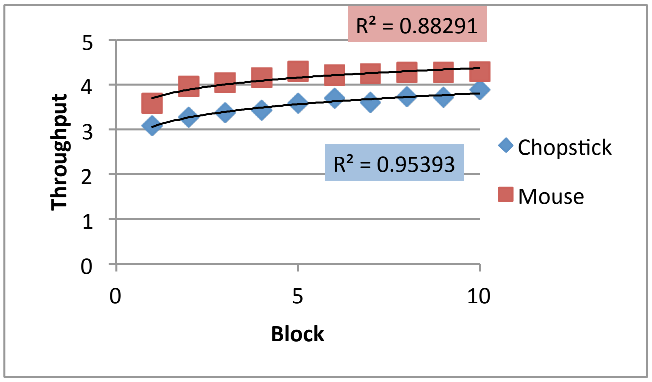

Figure 3-20: Graph of the learning over time. Average throughput for each block is displayed. Power curve displayed over points.

Index of Difficulty (ID)

The data was not normally distributed and also failed Levene's test for homogeneity. ART was used again to address this concern.

Device used (chopstick or mouse) crossed with ID value had no significant effect on throughput ($F_{1,5}$ = 0.02, ns). See Figure 3-21 for graph of average movement times for each ID value.

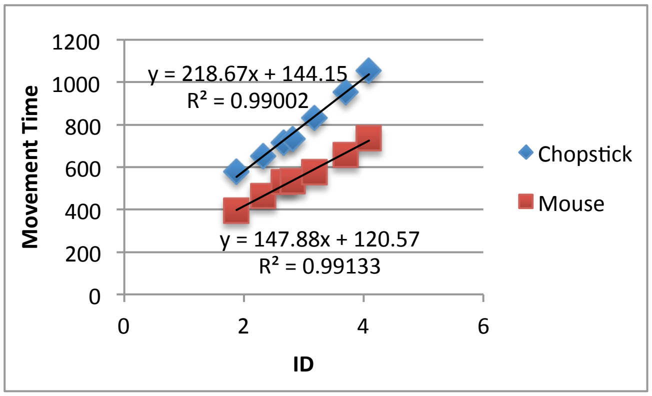

Figure 3-21: Graph of the average movement time (ms) required for each ID value for both the mouse and the chopstick. Fitts law model equations are shown as well.

3.4.6 Discussion

Comparing across the two pilot studies, the throughput results from this pilot study are statistically significantly different from the throughput values found in the higher latency environment for the chopstick ($F_{1,5} = 15$, $p < .001$) with a power ($1 - \beta$) of .97 and an effect size (η^2) of .01. Thus, I can state that reducing the latency did cause a noticeable improvement in user performance. Yet, even with the reduction in latency from 63 ms to 48 ms the throughput for the chopstick still has a .39 bps difference in throughput from the mouse by the last block (3.89 bps vs. 4.28 bps). That being said, the chopstick displays an average throughput of 3.89 bps by the tenth block in this pilot study. This value is within the lower end of observed throughput values for the mouse found in other experiments (3.7 bps – 4.9 bps) [36]. With more practice this throughput value may increase even more as evidenced by the fact that two participants in the study reached a crossover point where the chopstick had a throughput greater than the mouse. Moreover, an expert user exhibited an average throughput of 4.75 bps with the chopstick and 4.73 bps with the mouse. This user had been practicing various pointing methods for several hours a day over four months. So while mouse-like levels appear to be attainable with more training, the amount of training is prohibitive for a simple experiment.

3.5 Experiment 3

The main objective of this user study was to determine if a perfectly cylindrical, rigid finger would be capable of achieving the same levels of throughput seen with a chopstick in a comparable environment.

The conditions where participants wore the "cast" determine the effect a rigid cylindrical finger had on throughput and the effect that restricting finger bend had on throughput. I chose to focus on this as I found in pilot studies that if the finger was bent too far towards the tracking device the direction of the finger (as indicated by the tracking system) was not reliably detected. See Figure 3-22 for a depiction of this problem.

In this figure, the top two frames show an un-curved finger and the corresponding finger direction arrow. The next two frames show a slightly bent finger and the corresponding bent arrow. The final two frames depict the problem. Here the finger is bent even more than in the middle two frames but the system is now showing that the pointing direction is almost straight.

Moreover, I observed that some users had significantly more curved fingers than others. A good example for this finger curve is visible in the index finger in Figure 3-23, top right.

I also speculated that finger tracking might behave differently depending on whether the users held their hands palm facing down or palm rotated 90° inwards. The conditions where the participants held their hand rotated were included as these were situations where it might be easier for the device to track the position of the finger and determine the pointing direction – if finger curvature plays a significant role.

In summary, the four interaction methods that were compared were the Cast Normal method, the Cast Side method, the Normal method, and the Side method, see Figure 3-23).

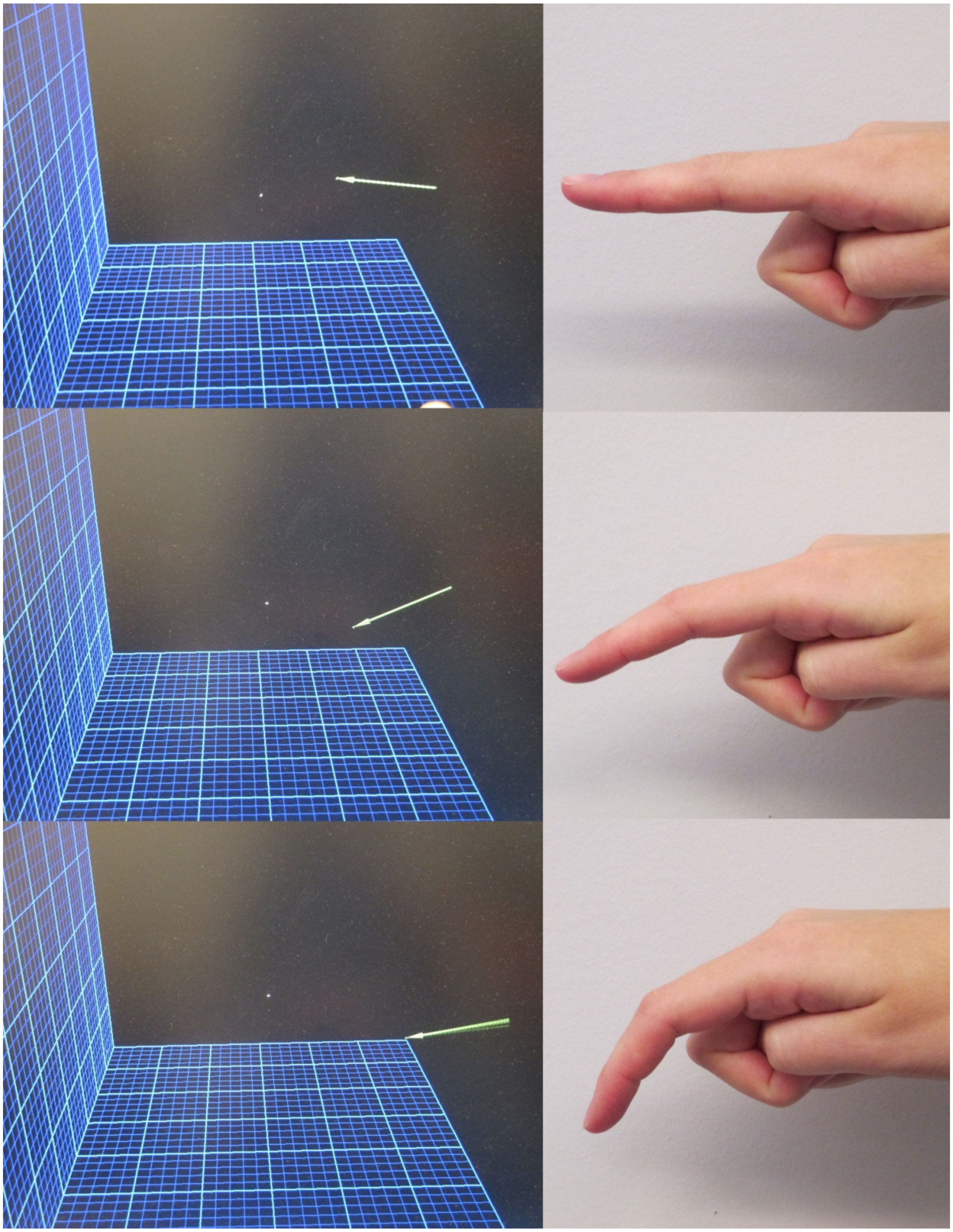

Figure 3-22: This figure depicts the issue observed with tracking bent fingers.

3.5.1 Participants

I recruited 8 participants for this study (mean age 20 years, SD 2.3 years). Three participants were male and all but one were right handed.

3.5.2 Setup

The computer that the experiment was run on had vsync turned off for this user study. The Leap Motion sensor was attached to the computer by a USB3 cable. The Leap Motion sensor was placed underneath the participants' hands in a location that was most comfortable to them. The diagnostic Leap Motion Visualizer was then used to detect and fix any possible interference with the Leap Motion system. The Leap Motion software used for this pilot was version 1.0.9+8410 and the Leap Motion hardware device was LM-010. The whole system was measured to have an end-to-end latency of 48ms. The software used for this Fitts' Law study was FittsStudy, which is available online [45]. I added support to read data from the LeapMotion to this package.

3.5.3 Input Conditions

For this experiment there were four input conditions for selecting targets for the participants to use. These were the *Cast Normal* method, the *Cast Side* method, the *Normal* method, and the *Side* method. The *Cast Normal* method required the user to wear a paper "cast" (illustrated in Figure 3-23) around her or his dominant pointer finger. This cast was specially designed and adapted to each user's finger. A piece of regular computer paper was cut so that it was wide enough to wrap around the user's finger and long enough to cover the finger to the tip. This piece of paper was then wrapped around the user's finger and taped with clear adhesive tape to form the "cast". Targets were then selected by wearing the "cast" and pointing the finger to move the cursor on the screen. The finger was held in the "normal" pointing orientation with the bottom of the user's palm facing down. In the *Cast Side* method, the "cast" was again worn on the user's finger but this time the finger was held in the "side" position with the user's palm perpendicular to the desk. The *Normal* method required the user to hold their hand with the palm facing down, toward the desk. No cast was worn in this condition. In the *Side* condition the user's palm was held perpendicular to the desk and no cast was worn on the finger. In all conditions, after targets had been acquired through pointing, selection was indicated via the spacebar on the keyboard. The spacebar was operated by the non-dominant hand of the participant and was placed in a comfortable operating position so that the dominant hand used for object acquisition was not obstructed.

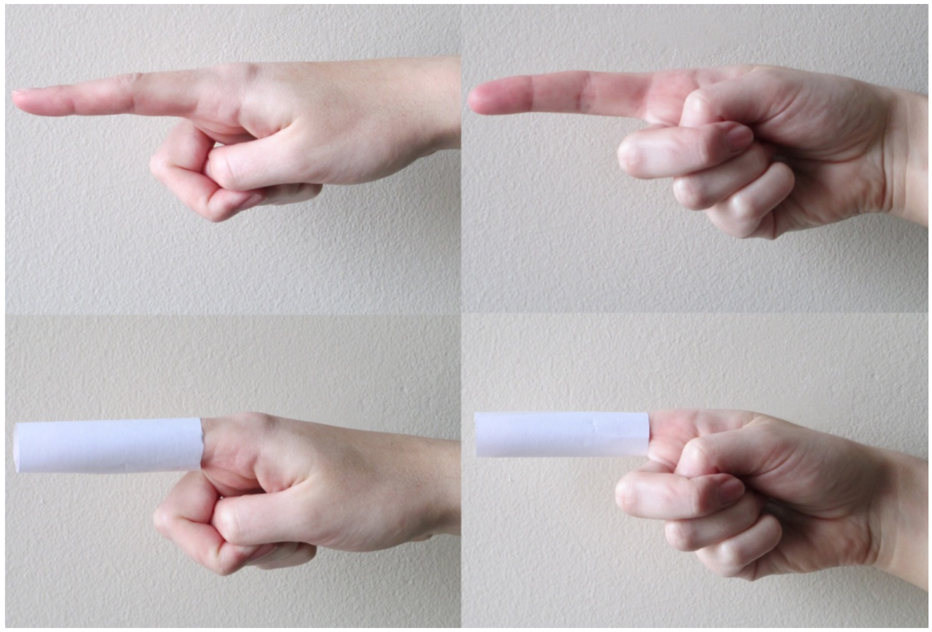

Figure 3-23: Picture of the four input conditions. From top left to bottom right: *Normal*, *Side*, *Cast Normal*, and *Cast Side*.

3.5.4 Procedure

First, the participants were given a brief questionnaire about their backgrounds. The questionnaire recorded gender, age, and handedness. Next, a “cast” was created for each participant as described in the Input Conditions section of the paper. Then, the participant was introduced to the finger tracking system and the experimenter demonstrated how it worked. After she or he was comfortable with the basic operation, one of the input conditions was explained to the participant. The order that participants were given each of the input methods was determined by a Latin Square design.

Once the participant was comfortable with the current input method, she or he completed a series of Fitts’ law selection tasks using one of the four input conditions.

Five blocks of 9 Fitts' law conditions with 11 trials per condition were completed. Target widths of 32, 64, and 96 and target amplitudes of 256, 384, and 512 were used. Thus, a total of 495 trials were completed for each condition. The participant would then be presented with the next input method and the above process would be repeated.

3.5.5 Results

Data was first filtered for participant errors, such as hitting the spacebar twice on the same target or pausing in the middle of a circle. Removing these errors amounted to less than .01% loss of total data collected.

Throughput

The data failed multiple normally tests. This invalidates a critical assumption needed for parametric repeated measures ANOVA. To address these concerns, I used the *Aligned Rank Transform (ART)* for nonparametric factorial data analysis in Human-Computer Interaction treatment presented by Wobbrock et al. [44].

There was no significant effect for the used interaction method ($F_{1,7} = 1.35$, $p >$ .05). See Figure 3-24 for average throughput values.

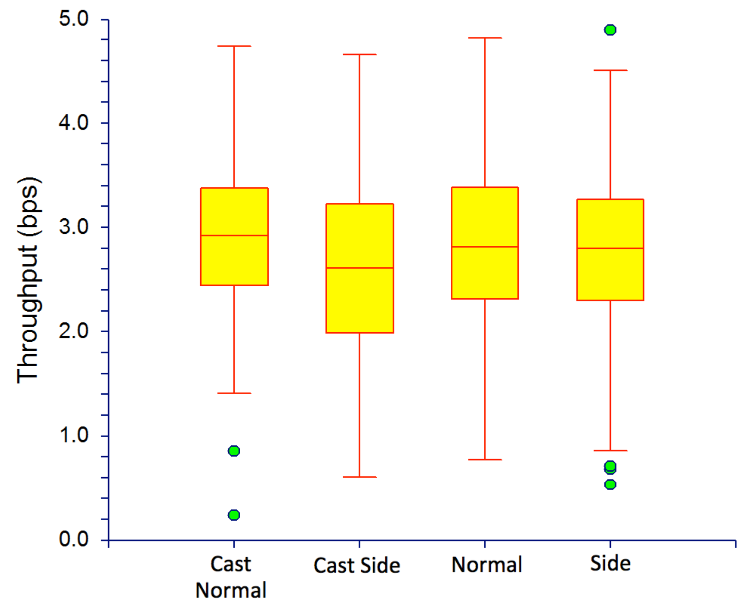

Figure 3-24: Box plot of average throughput values for each condition. No significant difference.

Movement Time

The data for movement time also failed multiple normality tests. Consequently, I used *ART* again.

There was no significance effect for the used interaction method ($F_{1,7} = 2.35$, $p > .05$). See Figure 3-26 and Figure 3-25 for average movement times.

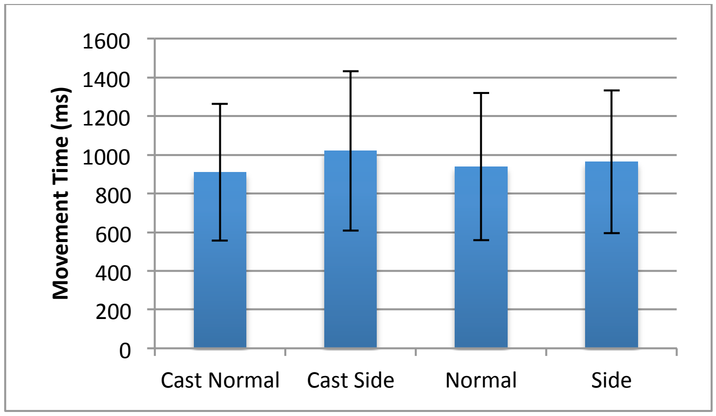

Figure 3-25: Graph of the average movement time for each condition. Error bars show standard deviation. No significant difference.

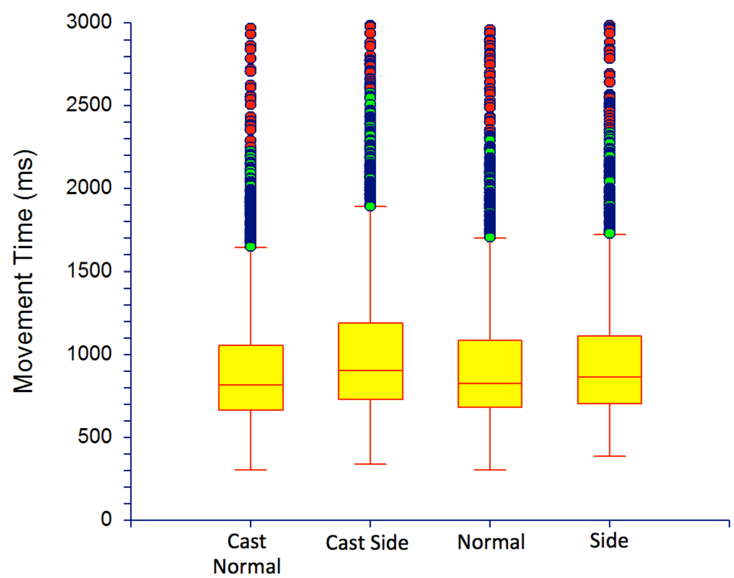

Figure 3-26: Box plot of the average movement time for each condition. The difference is not significant. The large number of outliers is due to skewed data.

Error Rate

The data for error rate also failed multiple normality tests. ART was used again to address this concern.

The used interaction method had no significant effect on error rate ($F_{1,7} = 0.27$, ns).

Learning

The data was not normally distributed and also failed Levene's test for homogeneity. ART was used again to address this concern.

There was no significant effect on learning overall ($F_{1,7} = 1.09$, $p > .05$) and no effect on learning crossed with the used interaction method ($F_{1,7} = 0.01$, $p > .05$). See Figure 3-27 for graph of average throughput for each of the five blocks for each condition.

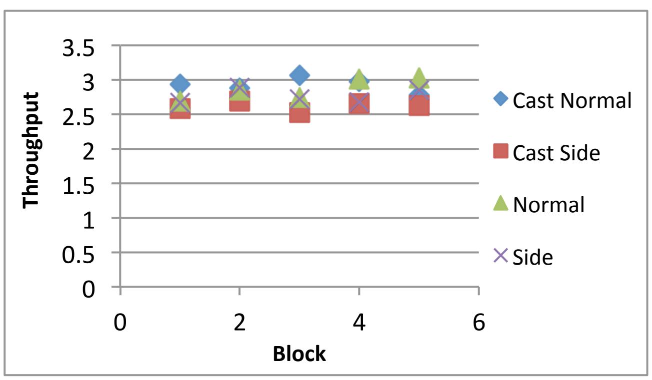

Figure 3-27: Graph of learning over time for each condition.

Index of Difficulty (ID)

The data for index of difficulty also failed multiple normality tests. ART was again used to address this concern.

The used interaction method crossed with ID had no significant effect on throughput ($F_{1,7} = 0.001$, ns). See Figure 3-28 for the data for all conditions. The equations and fit values for the Fitts' law models are as follows: *Cast Normal*: $y = 310.58\,x - 7.9434$, $R^2 = 0.9857$, *Cast Side*: $y = 356.31\,x - 26.25$, $R^2 = 0.9743$, *Normal*: $y = 337\,x - 49.201$, $R^2 = 0.9826$, *Side*: $y = 329.88\,x - 8.9465$, $R^2 = 0.9879$.

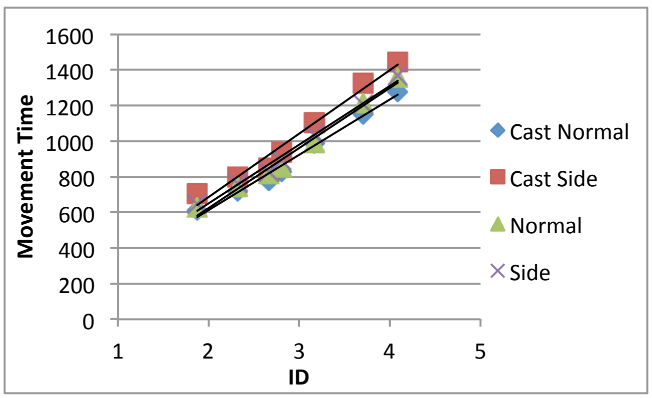

Figure 3-28: Graph depicting Fitts' law model for input conditions.

3.5.6 Discussion

These results indicate that the natural curvedness and potential flexibility of human finger is an unlikely cause for lower pointing throughput relative to a rigid object. It is also likely that any effect that the previously discussed finger bend tracking issue has on performance is negligible. After ruling out finger curvedness and the tracking reliability with bent fingers, there is still a 15+% difference (0.6 bps) between the throughputs of finger operation and chopstick operation that remains unaccounted for. Other remaining possible explanations for this difference are: the different operation method required for a chopstick (three finger stable grip) or the difference in chopstick and finger length. The difference in chopstick and finger width is also an unlikely explanation as Weichet et al. found that different pen widths did not have a significant effect on the tracking abilities of the tracking system I used [42].

3.6 Experiment 4

The purpose of this study was to look at the performance effect that varying degrees of click detection reliability have on throughput. This was done using a mouse, as I could easily and reproducibly control the reliability of the mouse to be a particular percentage. Moreover, the mouse button is normally 100% reliable (or at least appears to be). The results of such an experiment can be used to infer how much of a performance penalty is imposed by using a selection method that is not 100% reliable, e.g., by using gesture recognition systems.

3.6.1 Participants

I recruited 10 participants for this study (mean age 23 years, SD 4.7 years). Four participants were male and all but one were right handed. The left handed participant preferred to operate the mouse with the right hand.

3.6.2 Setup

The mouse used was a Microsoft IntelliMouse Optical set to the default pointer speed on the Windows 7 operating system. The system used with the mouse had an end-to-end latency of 28 ms. The software used for conducting the Fitts' law tasks was again FittsStudy [45].

3.6.3 Procedure

First, the participant was given a brief questionnaire about her or his background. The questionnaire recorded gender, age, and handedness. Then, the participant was informed that during this experiment the mouse button used for clicking would not always be reliable and that sometimes it might need to be clicked again. There were five levels of reliability tested in this experiment: 100%, 99%, 98%, 95%, and 90%. The order that participants received each of these conditions was counterbalanced so that each of the possible orders was represented equally. Participants then completed 2 blocks of 12 Fitts' law conditions with 11 trials per condition. Target widths of 16, 32, 64, and 96 and target amplitudes of 256, 384, and 512 were used. Participants completed a total of 264 targets per condition.

3.6.4 Results

Throughput

The data was not normally distributed. Also, the data failed Levene's test for homogeneity. This invalidates the assumptions needed for parametric repeated measures

ANOVA. To address these concerns, I used the *Aligned Rank Transform (ART)* for nonparametric factorial data analysis in Human-Computer Interaction treatment presented by Wobbrock et al. [44].

There was a significant effect for reliability level ($F_{4,9} = 13$, $p < .001$) with a power ($1 - \beta$) of .99 and an effect size (η^2) of .09. A Tukey-Kramer Multiple-Comparison test identified two statistically different groups. Group one consists of 90% and 95% reliability and group two of 98%, 99%, and 100% reliability. Conditions within groups were not statistically significant from each other. See Figure 3-30 and Figure 3-29 for average throughput values.

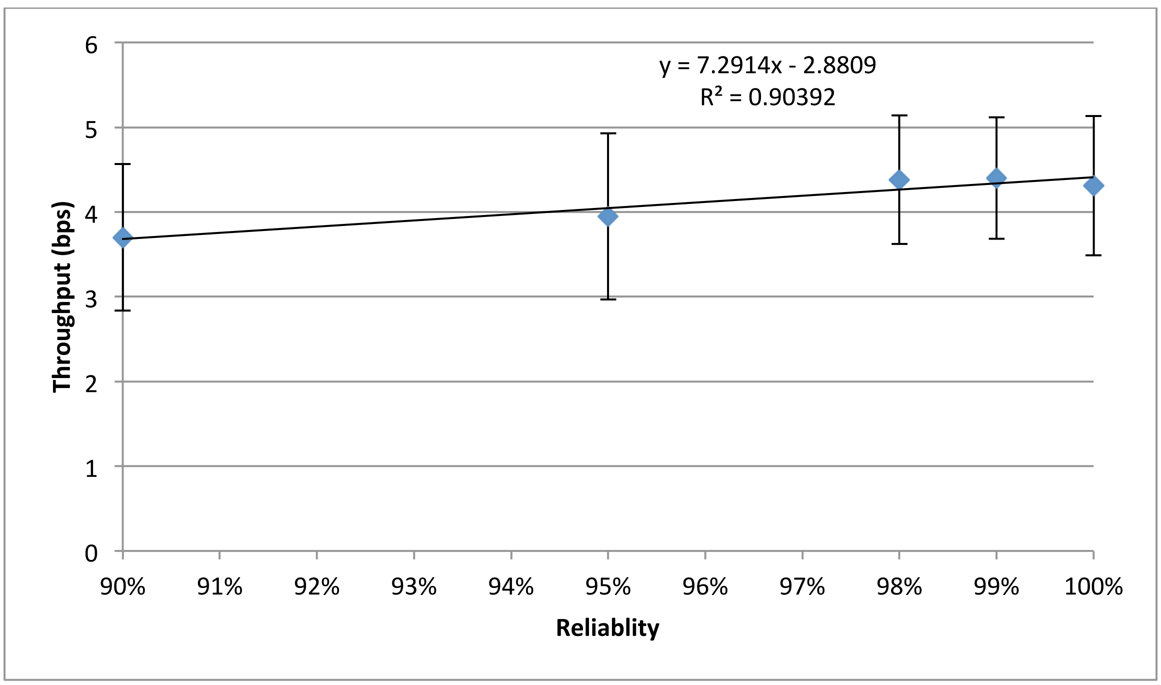

Figure 3-29: Average throughput values for each reliability level. Error bars show standard deviation. A linear trendline and its corresponding equation are also shown.

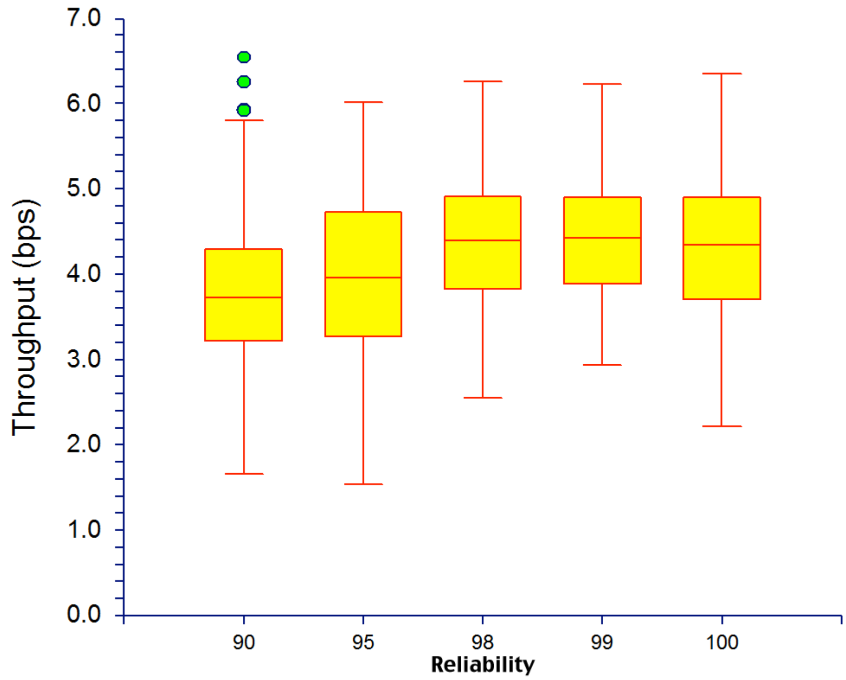

Figure 3-30: Box plot of average throughput values for each reliability level. There is a significant difference between conditions.

Movement Time

The data for movement time was not normally distributed and also, failed Levene's tests for homogeneity. Consequently, I used *ART* again.

There was a significant effect for reliability level ($F_{4,9} = 8$, $p < .001$) with a power ($1 - \beta$) of .99 and an effect size (η^2) of .01. A Tukey-Kramer Multiple-Comparison test again identified two statistically significant groupings. However, these two groups were different than for throughput. Group one consisted of 90%, 95% and 98% reliability and group two consisted of 98%, 99% and 100% reliability. In other words, 98% was not statistically different from any of the other conditions. See Figure 3-32 and Figure 3-31 for average movement time values.

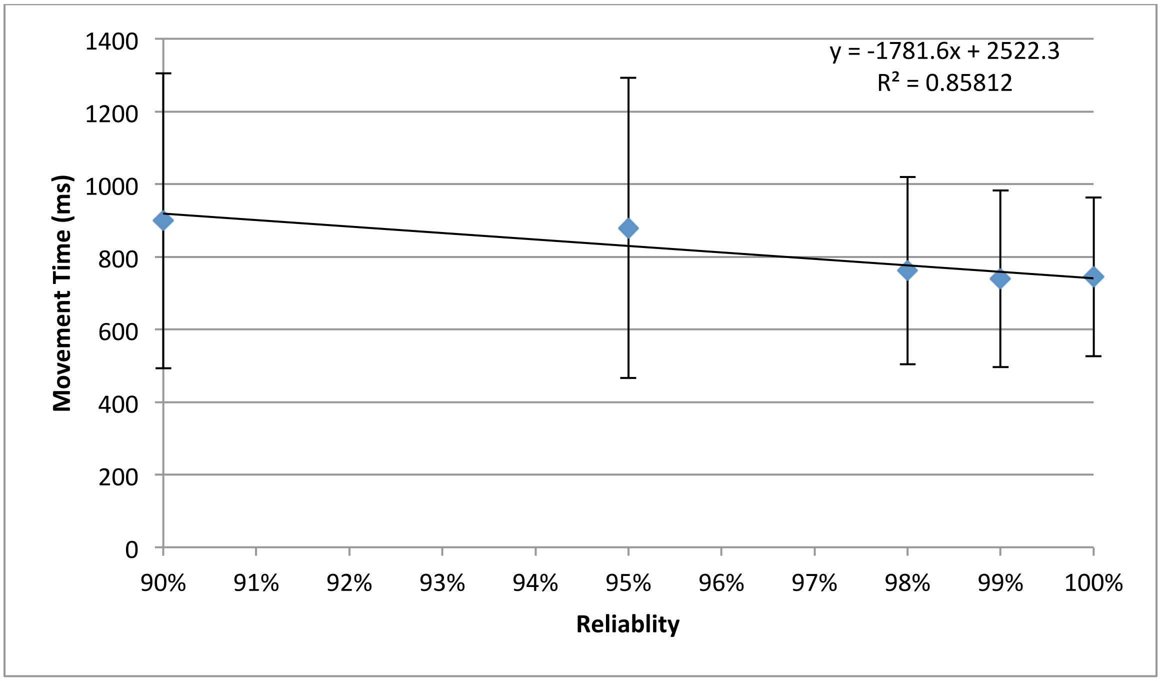

Figure 3-31: Average movement time values for each reliability level. Error bars show standard deviation. A linear trendline and its corresponding equation are also shown.

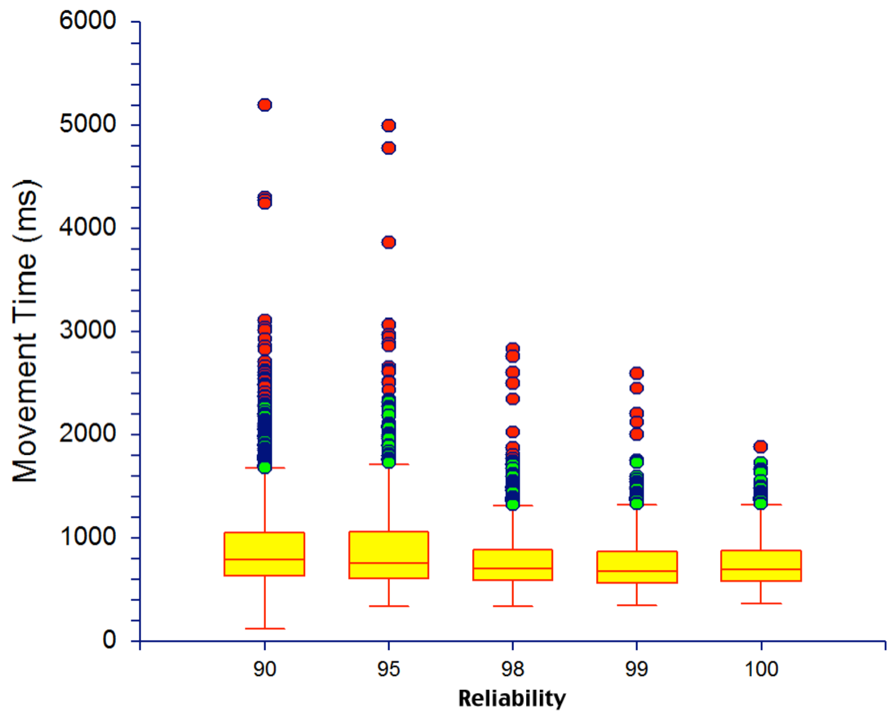

Figure 3-32: Box plot of the average movement time values for each reliability level.

Error Rate

The data was not normally distributed and also failed Levene's test for homogeneity. ART was used again to address this concern.

The used reliability level had no significant effect on error rate ($F_{2,9} = 1.86$, $p > .05$).

Learning

The data was not normally distributed and also failed Levene's test for homogeneity. ART was used again to address this concern.

There was no significant effect on learning overall ($F_{4,9} = 0.06$, ns) and no effect on learning crossed with level of reliability ($F_{4,9} = 0.06$, ns).

Index of Difficulty (ID)

The data was not normally distributed and also failed Levene's test for homogeneity. ART was used again to address this concern.

Reliability level crossed with ID had no significant effect on throughput ($F_{4,9} = 6$, $p > .05$). See Figure 3-33 for the data for all conditions. The equations for the Fitts' law models are as follows: 90%: $y = 170.82\,x + 155.98$, $R^2 = 0.988$, 95%: $y = 148.78\,x + 212.89$, $R^2 = 0.988$, 98%: $y = 167.44\,x + 95.258$, $R^2 = 0.987$, 99%: $y = 156.39\,x + 103.94$, $R^2 = 0.996$, 100%: $y = 141.48\,x + 138.47$, $R^2 = 0.998$.

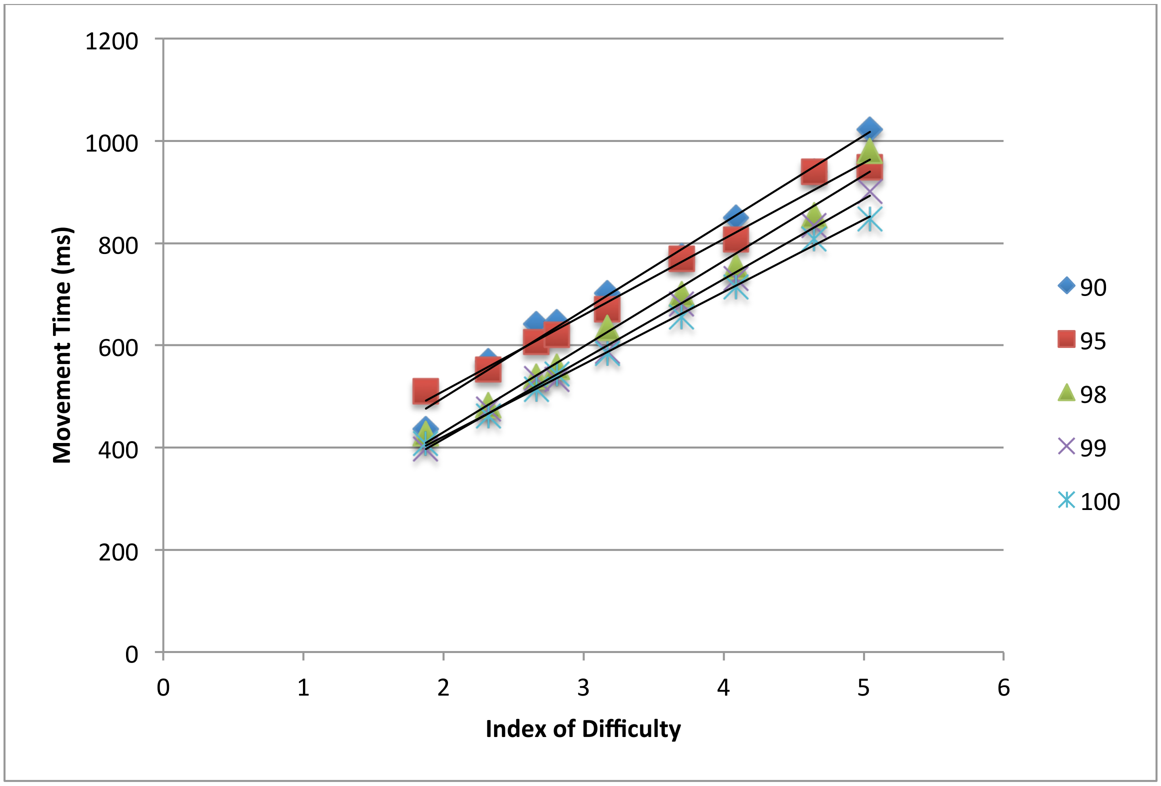

Figure 3-33: Graph depicting Fitts' law model for different levels of reliability.

3.6.5 Discussion

These results indicate a linear drop-off in pointing performance as the selection system becomes more and more unreliable. The 90% and 95% conditions performed significantly worse than the 98%, 99%, and 100% conditions in terms of throughput. It is clear from this that any gesture recognition system that is 95% or less reliable is going to noticeably and negatively impact interaction performance with a system. While this is not good news for many of the gesture systems developed so far, it also provides a benchmark value for developers to aim for.

While there was no significant difference in performance between 100%, 99%, and 98% (as seen in previous work on errors in typing [1]), participants did still notice when they were not in the 100% condition. Every time a click recognition failure occurred, they would have to stop midway on their path to the next target and go back to select the previous target again. Signs of annoyance sometimes accompanied this process (i.e. sighing and other interjections). This indicates that while a system with a reliability above 95% might not suffer much in terms of throughput, its failures will still be noticeable to the users. A possible explanation for why even small amounts of errors are so noticeable is that participants are used to using a 100% reliable mouse from their daily computer usage. Thus, people expect this device to always work. When a user feels the depression of the button and the click is not registered this failure becomes glaring to them. Small amounts of errors might be less recognisable in systems that have no force feedback or systems that users expect to be unreliable.

During the experiment there was a distinct behavioural difference detected for the 90% condition. In this condition, it was observed that most participants would pause after selecting a target before moving onto the next one. In all other conditions, participants would immediately move to select the next target and then have to backtrack when they noticed that selection had not worked. In the 90% condition it seemed like the participants expected failure rather than success. Even though failures were just happening 10% of the time, most participants felt that they could no longer rely on the system working as they had in all the other conditions. I suspect that as the reliability gets even lower *all* participants would anticipate a failure, not just most of them. It is a very interesting observation, that at just 10% failures, systems are already seen as more unreliable than reliable.

Chapter 4
Overall Discussion

In this research work, I largely rule out seven potential explanatory factors (arm stabilization, hand vs. finger pointing, end-to-end latency, finger shape, tracking difficulties due to finger bend, and button vs. spacebar selection method) for the lower throughput observed with un-instrumented in-air tracking systems. I also show how much of a performance penalty can be expected from selection methods that are not 100% reliable. This is particularly relevant to the use of un-instrumented in-air pointing systems as with these systems the use of a physical button for selection is unlikely as this would invalidate the contact-free benefits such systems provide. This considerably narrows the search space for future research and brings the community closer to tracking down the factor or factors that contribute to the reduced throughput for such systems. The fact that I present this research using the ISO 9241-9 standard enables my research to be easily compared to other past and future work.

In experiment 1, I showed that both arm stabilization and cursor control method had no significant affect on participant performance overall. However, for *individual* participants the type of control and/or whether their elbows were on the desk or not, had a significant impact on their performance. While elbow placement is user adjustable, system control methods are not. If a developer only supports one control method for an un-instrumented pointing device, then a subset of users might have a problem with efficient operation of the system. This finding is important for system developers to keep in mind when designing their products. As my research indicates that there is no "one size fits all" solution to control of un-instrumented in-air pointing systems, it would be advantageous to design a system that works with the control method the user performs best with.

Experiment 2 provides a direct comparison of in-air un-instrumented pointing to the mouse. This enables the results of in-air un-instrumented operation to be compared with other devices in other work, as the mouse results can be used as a calibration point. This experiment also compares the mouse with both a button and spacebar selection to in-air un-instrumented pointing with both a button and spacebar selection. Not only does this permit a more direct comparison of the two devices, it also shows the effect of using the same hand for both target acquisition and selection. This study shows that there is no statistically significant performance difference between using the same hand for both target acquisition and target selection compared to using different hands.

Pilot Study 1 investigates the effect chopstick operation has on performance. I hypothesized that a wooden chopstick would be a much easier object for the system to track and so any difficulties the system had with tracking the finger would not be present in the results of this pilot study. I show that while using a chopstick does improve participant performance it is not enough to bring it to the levels of throughput seen with the mouse. The increase in throughput was promising though and Pilot Study 2 was designed to see if a reduction in latency would be enough to bring performance to the levels seen with the mouse.

Pilot Study 2 did find a substantial increase in throughput for chopstick performance, but not quite enough to reach the performance levels seen with the mouse. These pilot studies showed chopstick performance to be much better than the throughput values obtained when participants were using their fingers to point.

A potential explanation for the fact that pointing with a chopstick performed better than finger pointing is unreliable finger tracking. To investigate this I developed experiment 3. This experiment was designed to test the hypothesis that this difference is due to natural finger curvedness or due to the tracking issues with fingers that are bent too far towards the device.

In experiment 3 participants operated the tracking system while wearing a paper cast to keep their fingers straight and rigid (to prevent the finger bend tracking issue described in section 3.5) and so that their fingers would appear as perfect cylinders to the tracking system. I found that there was no statistically significant performance difference between any of these conditions. As such, it is unlikely that these are the causes of the decreased throughput for finger pointing relative to a chopstick. The remaining explanations for the throughput difference are the length difference between fingers and chopsticks (width differences are unlikely, as other research already eliminates this possibility [42]) or the different muscle groups required for pointing operation with each of these two methods. If the explanation for poor finger pointing performance is the length difference, then better tracking code might result in better overall finger performance. However, if the explanation is the different muscle groups required for operation of each of the two methods, then the issue is likely a human limitation that cannot be circumvented.

In experiment 4 participants used a mouse with a button with varying degrees of reliability to select targets. I found that participants would notice even a small amount of unreliability, but that their pointing performance was significantly impacted only for reliabilities of 95% and worse. At 90% reliability, participant behavior completely changed and participants would hesitate before moving on to the next target to make sure that the selection had actually been preformed. This indicates that at 90% reliability participants consider selection to no longer be dependable enough to expect that it will always work.

In general, I can state that modern instantiations of in-air interaction already achieve throughput comparable or better to that of state-of-the-art 3D tracking systems. Yet, based on my results, I cannot recommend un-instrumented in-air hand tracking for user interfaces that require the best possible pointing performance or that need to compete with mice or touch screens. Having said that, un-instrumented hand tracking is attractive for many other applications, such as user interfaces that require the use of multiple fingers at the same time, casual games, bi-manual interaction, gesture-based systems, and applications where the user cannot touch or hold a device.

4.1 Limitations

In order to study in-air device performance it was necessary to choose an un-instrumented system to perform the research with. In the case of this research, the Leap

Motion was selected. While this is a necessary step, it potentially causes particulars of the selected system to be studied, rather than the class of devices it is meant to represent. While I have tried to make this research as device-independent as possible, it is possible that particular aspects that are Leap Motion-specific have exhibited themselves in the results.

Chapter 5
Conclusion

I conducted four experiments and two pilot studies to explore the performance of un-instrumented 3D tracking for controlling a cursor on a screen. I show that elbow support is not uniformly advantageous for in-air interaction and as such does not need to be controlled for in experiments. Likewise, the best method of cursor control, either *Pointing* or *Whole Hand*, varies from one individual to the next. This suggests that one method of operation is not universally beneficial and that multiple methods should be permitted to accommodate different users.

Experiment 2 provided evidence that un-instrumented, in-air tracking is unlikely to reach levels of throughput equivalent to the mouse, even if the technical implementation has minimum latency. Consequently, I believe that some other factor accounts for the decrease in throughput. I also show that there is no statistically significant benefit to using a button for selection with the same hand that is being used to move the cursor. As such, this is another factor that does not need to be controlled for in future experiments.

I also show that user performance can be improved by operating the system with a chopstick. Even with a reduction in latency, chopstick operation does not quite reach the levels of throughput seen with the mouse, but it performs much better than the finger. The exact reason for this substantial difference is still unclear. However, through experiment 3, the potential list has been narrowed down to length of the pointing item or the different muscle groups required by the two methods.

Finally, I show that unreliable selection affects performance (approximately) linearly and I have identified some key values in the range of reliabilities between 90% and 100%. At 95% reliability and below, participant performance it significantly impacted. Moreover, at 90% reliability and below participant behavior switches from a model of expected success to expected failure. This result enables designers of tracking systems to better estimate how varying degrees of gesture system unreliability impact the performance of their system overall.

As for the outlook of these systems and if finger-pointing performance can be improved to the level of a chopstick, then the future of these devices looks promising. If finger pointing cannot be improved to that level, due to basic human limitations, then I suggest using un-instrumented 3D pointing systems only in situations where interaction would benefit from using a hands-free, sterile, multi-hand, or multi-finger device. In other situations, it is likely that the mouse is still better suited to the task.

Bibliography

1. Arif, A.S. and Stuerzlinger, W. Predicting the Cost of Error Correction in Character-Based Text Entry Technologies. *ACM SIGCHI Conference on Human Factors in Computing Systems*, ACM Press (2010), 5 – 14.
2. Arif, A.S. and Stuerzlinger, W. User Adaptation to a Faulty Unistroke-Based Text Entry Technique by Switching to an Alternative Gesture Set. *Graphics Interface*, (2014).
3. Ashbrook, D., Baudisch, P., and White, S. Nenya: subtle and eyes-free mobile input with a magnetically-tracked finger ring. *ACM SIGCHI Conference on Human factors in computing systems*, ACM (2011), 2043–2046.
4. Balakrishnan, R., Baude, T., Kurtenbach, G., and Fitzmaurice, G. The Rockin'Mouse : Integral 3D Manipulation on a Plane. *ACM SIGCHI Conference on Human factors in computing systems*, ACM (1997), 311–318.
5. Balakrishnan, R. and MacKenzie, I.S. Performance differences in the fingers, wrist, and forearm in computer input control. *ACM SIGCHI Conference on Human factors in computing systems*, (1997), 303–310.
6. Banerjee, A. and Burstyn, J. Pointable: an in-air pointing technique to manipulate out-of-reach targets on tabletops. *ACM International Conference on Interactive Tabletops and Surfaces*, ACM (2011), 11–20.
7. Brown, M.A., Stuerzlinger, W., and Mendonça Filho, E.J. The Performance of Un-Instrumented In-Air Pointing. *Graphics Interface*, (2014).

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

9. Das, K. and Borst, C.W. An evaluation of menu properties and pointing techniques in a projection-based VR environment. *2010 IEEE Symposium on 3D User Interfaces*, (2010), 47–50.

10. Gallo, L., Placitelli, A., and Ciampi, M. Controller-free exploration of medical image data: Experiencing the Kinect. *Computer-Based Medical Systems*, (2011), 1–6.

11. Gokturk, M. and Sibert, J.L. An analysis of the index finger as a pointing device. *Extended Abstracts on Human Factors in Computing Systems*, ACM (1999), 286.

12. Grossman, T. and Balakrishnan, R. Pointing at trivariate targets in 3D environments. *ACM SIGCHI Conference on Human factors in computing systems*, ACM Press (2004), 447–454.

13. Von Hardenberg, C. and Bérard, F. Bare-hand human-computer interaction. *Workshop on Percetive user interfaces*, (2001).

14. Harrison, C. and Hudson, S.E. Abracadabra : Wireless , High-Precision , and Unpowered Finger Input for Very Small Mobile Devices. *ACM symposium on User interface software and technology*, ACM (2009), 121–124.

15. ISO. 9241-9 Ergonomic requirements for office work with visual display terminals (VDTs)-Part 9: Requirements for non-keyboard input devices (FDIS-Final Draft International Standard). *International Organization for Standardization*, (2000).
16. Jota, R., Nacenta, M., and Jorge, J. A comparison of ray pointing techniques for very large displays. *Graphics Interface*, Canadian Information Processing Society (2010), 269–276.
17. Kang, H., Woo Lee, C., and Jung, K. Recognition-based gesture spotting in video games. *Pattern Recognition Letters 25*, 15 (2004), 1701–1714.
18. Kim, J.H. An HMM-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 21*, 10 (1999), 961–973.
19. Kitamura, Y., Higashi, T., Masaki, T., and Kishino, F. Virtual chopsticks: Object manipulation using multiple exact interactions. *Virtual Reality*, (1999), 198–204.
20. Kolaric, S., Raposo, A., and Gattass, M. Direct 3D manipulation using vision-based recognition of uninstrumented hands. *X Symposium on Virtual and Augmented Reality*, (2008), 212–220.
21. Kunert, A., Kulik, A., Lux, C., and Fröhlich, B. Facilitating system control in ray-based interaction tasks. *ACM Symposium on Virtual Reality Software and Technology*, ACM (2009), 183–186.
22. Lee, H.-K. and Kim, J.-H. Gesture spotting from continuous hand motion. *Pattern Recognition Letters 19*, 5-6 (1998), 513–520.

23. Liang, R. and Ouhyoung, M. A Real-time Continuous Gesture Recognition System for Sign Language. *Automatic Face and Gesture Recognition*, (1998), 558–567.

24. MacKenzie, I. and Jusoh, S. An evaluation of two input devices for remote pointing. *Engineering for Human-Computer Interaction 2254*, (2001), 235–250.

25. MacKenzie, I.S. Fitts' Law as a Research and Design Tool in Human-Computer Interaction. *Human-Computer Interaction 7*, 1 (1992), 91–139.

26. Matikainen, P., Pillai, P., Mummert, L., Sukthankar, R., and Hebert, M. Prop-free pointing detection in dynamic cluttered environments. *Face and Gesture*, (2011), 374–381.

27. Miner, C., Chan, D., and Campbell, C. Digital jewelry: wearable technology for everyday life. *Extended Abstracts on Human Factors in Computing Systems*, ACM (2001), 45–46.

28. Morency, L.-P., Quattoni, A., and Darrell, T. Latent-Dynamic Discriminative Models for Continuous Gesture Recognition. *Computer Vision and Pattern Recognition*, Ieee (2007), 1–8.

29. Oakley, I., Sunwoo, J., and Cho, I.-Y. Pointing with fingers, hands and arms for wearable computing. *Extended Abstracts on Human Factors in Computing Systems*, ACM (2008), 3255–3260.

30. Ogura, T., Sato, M., Ishida, Y., Hayashi, N., and Doi, K. Development of a novel method for manipulation of angiographic images by use of a motion sensor in operating rooms. *Radiological physics and technology*, (2014).

31. Oh, J. and Stuerzlinger, W. Laser pointers as collaborative pointing devices. *Graphics Interface 2002*, (2002), 141–149.

32. Pavlovych, A. and Stuerzlinger, W. The tradeoff between spatial jitter and latency in pointing tasks. *ACM SIGCHI symposium on Engineering interactive computing systems*, ACM (2009), 187–196.

33. Rhee, S., Yang, B., and Asada, H. Artifact-resistant power-efficient design of finger-ring plethysmographic sensors. *IEEE Transactions on Biomedical Engineering 48*, 7 (2001), 795–805.

34. Schlomer, T., Poppinga, B., Henze, N., and Boll, S. Gesture Recognition with a Wii Controller. *Conference on Tangible and Embedded Interaction*, (2008), 11–14.

35. Song, P., Yu, H., and Winkler, S. Vision-based 3D finger interactions for mixed reality games with physics simulation. *ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, (2008).

36. Soukoreff, R.W. and MacKenzie, I.S. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. *International Journal of Human-Computer Studies 61*, 6 (2004), 751–789.

37. Sparrow, W. and Newell, K. Metabolic energy expenditure and the regulation of movement economy. *Psychonomic Bulletin & Review 5*, 2 (1998), 173–196.

38. Teather, R.J., Pavlovych, A., Stuerzlinger, W., and MacKenzie, I.S. Effects of tracking technology, latency, and spatial jitter on object movement. *2009 IEEE Symposium on 3D User Interfaces*, Ieee (2009), 43–50.

39. Teather, R.J. and Stuerzlinger, W. Pointing at 3D targets in a stereo head-tracked virtual environment. *2011 IEEE Symposium on 3D User Interfaces*, Ieee (2011), 87–94.

40. Teather, R.J. and Stuerzlinger, W. Pointing at 3d target projections with one-eyed and stereo cursors. *ACM SIGCHI Conference on Human Factors in Computing Systems*, ACM (2013), 159.

41. Vogel, D. and Balakrishnan, R. Distant freehand pointing and clicking on very large, high resolution displays. *ACM symposium on User interface software and technology*, ACM Press (2005), 33.

42. Weichert, F., Bachmann, D., Rudak, B., and Fisseler, D. Analysis of the accuracy and robustness of the leap motion controller. *Sensors (Basel, Switzerland) 13*, 5 (2013), 6380–93.

43. Wingrave, C. and Bowman, D. Baseline factors for raycasting selection. *Proc. of HCI International*, ACM (2005).

44. Wobbrock, J.O., Findlater, L., Gergle, D., and Higgins, J.J. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only ANOVA Procedures. *ACM SIGCHI Conference on Human Factors in Computing Systems*, ACM (2011), 143–146.

45. Wobbrock, J.O., Shinohara, K., and Jansen, A. The Effects of Task Dimensionality, Endpoint Deviation , Throughput Calculation , and Experiment Design on Pointing Measures and Models. *ACM SIGCHI Conference on Human Factors in Computing Systems*, ACM (2011), 1639–1648.

46. Yoon, H.-S., Soh, J., Bae, Y.J., and Seung Yang, H. Hand gesture recognition using combined features of location, angle and velocity. *Pattern Recognition 34*, 7 (2001), 1491–1501.

47. Zhai, S., Hunter, M., and Smith, B. a. The metropolis keyboard - an exploration of quantitative techniques for virtual keyboard design. *ACM symposium on User interface software and technology*, ACM Press (2000), 119–128.

48. Zigelbaum, J., Browning, A., Leithinger, D., Bau, O., and Ishii, H. G-stalt: a chirocentric, spatiotemporal, and telekinetic gestural interface. *International Conference on Tangible, Embedded, and Embodied iIteraction*, ACM (2010), 261–264.

Appendix A: Levene's Test

Levene's test is a statistical test that determines the equality of variances for a variable calculated for two or more conditions. It is used in this work to check to see if the assumption of homogeneity of the variances holds before an Analysis of variance (ANOVA) test is performed.

Levene's test tests the null hypothesis that the population variances are equal. In my research I rejected this null hypothesis when the p-value was $< .05$. When the null hypothesis was rejected, it indicated that one of the assumptions of the ANOVA test had been violated and that it was unsuitable to use the ANOVA test at this time.

Appendix B: Aligned Rank Transform

The Aligned rank transform (ART) is a data transformation usable for data that violates the ANOVA test assumptions. It transforms data into a form that is suitable for an ANOVA test, by simply ranking the data. If this transform were not performed the results of the ANOVA would have an inflated risk of Type I error rates (an incorrect rejection of the null hypothesis).

This transform provides accurate nonparametric treatment for both main and interaction effects and so is useful in the analysis of my data. This transform has the pre-processing step of aligning the data for each effect before assigning ranks, which makes nonparametric treatment of interaction effects possible. This is one of the innovations in the ART method over previous work [44].

There are five steps to this procedure. Step one is computing the residuals. Step two is computing the estimated effects for all main and interaction effects. Step three is computing the aligned response (Y'). Step four is assigning averaged ranks (Y"). Lastly, step five is preforming the ANOVA on the computed Y". For more interested readers, these steps are described in greater detail in the work by Wobbrock et al. [44].