Summer 2012

Assistive strategies for people with fine motor skills impairments based on an analysis of sub-movements

Guarionex Jordan Salivia
University of Iowa

ASSISTIVE STRATEGIES FOR PEOPLE WITH FINE MOTOR SKILLS IMPAIRMENTS BASED ON AN ANALYSIS OF SUB-MOVEMENTS

by

Guarionex Jordán Salivia

An Abstract

Of a thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Computer Science
in the Graduate College of
The University of Iowa

July 2012

Thesis Supervisor: Associate Professor Juan Pablo Hourcade

ABSTRACT

Four studies describe the pointing performance of individuals with fine motor skills impairments. First, we describe the pointing performance of two individuals with Parkinsons disease via a sub-movement analysis and compare them with similar results found in the literature from young children and older able-bodied adults. The analysis suggests the need of an individual assessment of pointing difficulties and the personalization of the methods of assistance and motivates sub-sequent studies. Two experiments followed where we tested *PointAssist*, software that assists in pointing tasks by detecting difficulty through a sub-movement analysis and triggering help, with adjustments proposed to personalize the assistance provided. A within-subjects study with sixteen individuals with fine motor skills impairments resulted in statistically significant effects on accuracy using Friedman's test with $(\chi^2(1) = 6.4, p = .011)$ in favor of personalized *PointAssist*. A five week longitudinal study with three participants with Cerebral Palsy and other fine motor skills impairments shows the long term effects of *PointAssist*. The longitudinal study logged real-world use of pointing devices validating the results for real-world interactions. *PointAssist* had statistically significant effect of reduced sub-movement length and speed with $p < .00001$ and $p < .0002$ respectively for one of the participants. These results suggest better motor control near a target and statistically significant results on the sub-movement duration confirmed this. Finally, we designed, developed and tested a new assistive technology for individuals with severe motor skills impairments that we call the

Reverse Funnel. Three participants, two with Cerebral Palsy and one with an undisclosed disability, participated and positive early results are presented as well as future developments of the newly developed strategy.

Abstract Approved: ______________________________
Thesis Supervisor

____
Title and Department

____
Date

ASSISTIVE STRATEGIES FOR PEOPLE WITH FINE MOTOR SKILLS IMPAIRMENTS BASED ON AN ANALYSIS OF SUB-MOVEMENTS

by

Guarionex Jordán Salivia

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Computer Science
in the Graduate College of
The University of Iowa

July 2012

Thesis Supervisor: Associate Professor Juan Pablo Hourcade

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Guarionex Jordán Salivia

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Computer Science at the July 2012 graduation.

Thesis committee: ______________________________
Juan Pablo Hourcade, Thesis Supervisor

____
Steve Bruell

____
Alberto Segre

____
Aaron Stump

____
Larissa Samuelson

A mi hija Bianca que me dio la fuerza para inspirarme a trabajar

ACKNOWLEDGEMENTS

First and foremost, thanking the following person does not come close to describing my love and gratitude. Angelique Dwyer has been not only my life partner but my confidant and my primary support. Without her, I would be emotionally empty. I thank her for always being patient with me and letting me have my way throughout this odyssey.

I would like to thank my mother Gloria Salivia for her continuous support and love, for always believing in me, and for tirelessly helping me during the most difficult times.

My father Francisco Jordan taught me never to think that my work is completed, encouraging me to think that if I ever though I completed my work then I did not work enough. Although this stage of my life has passed, I will always thrive to follow his wisdom and continue to work until my goals are fulfilled and beyond.

I would like to thank my advisor Juan Pablo Hourcade for allowing me to demonstrate that I was capable of finishing this work and for giving me the guidance necessary to learn and to develop my ideas. He helped me rediscover a passion for the Computer Science area that I though I had lost, and for that I will always have him in my highest regard.

I give special thanks to Steve Bruell, who has always been my advisor, mentor and friend. Steve is not only one of the best professors I ever had, but also one of the few people that never stopped believing in my potential.

I thank Donald Dwyer for being my friend and for accompanying me on those long trips to visit the participants. I thank him and Linda Dwyer for always being there for us whenever we needed without questioning, unselfishly giving their time so that I may complete this project.

I would also like to thank my good friends Marta Hamity, Cristian Bejarano and Jorge Ernie for their unconditional friendship and moral support.

Special thanks to Janet Motika, Lisa Dwyer, Leslie Dwyer, Kristin Beusse and to all the participants for their support, their time, their efforts and their warm wishes.

Finally, I would like to thank my daughter Bianca for being the light in my life, for giving me with every smile the energy I needed to succeed.

ABSTRACT

Four studies describe the pointing performance of individuals with fine motor skills impairments. First, we describe the pointing performance of two individuals with Parkinsons disease via a sub-movement analysis and compare them with similar results found in the literature from young children and older able-bodied adults. The analysis suggests the need of an individual assessment of pointing difficulties and the personalization of the methods of assistance and motivates sub-sequent studies. Two experiments followed where we tested *PointAssist*, software that assists in pointing tasks by detecting difficulty through a sub-movement analysis and triggering help, with adjustments proposed to personalize the assistance provided. A within-subjects study with sixteen individuals with fine motor skills impairments resulted in statistically significant effects on accuracy using Friedman's test with ($\chi^2(1) = 6.4, p = .011$) in favor of personalized *PointAssist*. A five week longitudinal study with three participants with Cerebral Palsy and other fine motor skills impairments shows the long term effects of *PointAssist*. The longitudinal study logged real-world use of pointing devices validating the results for real-world interactions. *PointAssist* had statistically significant effect of reduced sub-movement length and speed with $p < .00001$ and $p < .0002$ respectively for one of the participants. These results suggest better motor control near a target and statistically significant results on the sub-movement duration confirmed this. Finally, we designed, developed and tested a new assistive technology for individuals with severe motor skills impairments that we call the

Reverse Funnel. Three participants, two with Cerebral Palsy and one with an undisclosed disability, participated and positive early results are presented as well as future developments of the newly developed strategy.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1
INTRODUCTION

We spend time designing and developing assistive devices and technologies to help people with disabilities have an easier time in an ever changing world. When we think of assistive devices we think of wheelchairs and mechanical arms that are developed specifically to aid people with disabilities with some of their daily tasks. By definition, an assistive device is "any device that allows an individual to perform a task they would otherwise be unable to do" [12]. But what about devices to help individuals with the use of their computers? Devices that assist people with the use of their computers may include input devices such as keyboards with changed layouts and pointing devices such as trackballs and trackpads, though none of these are necessarily developed for individuals with disabilities. In fact, efforts have been made to maintain the use of common input devices such as the mouse as opposed to specialized alternative input devices for people with disabilities. The use of specialized input devices makes sense in cases of severely disabled individuals. But for other individuals, it makes sense to insist in the use of the mouse instead of alternative input devices. First, the mouse is one of the most common forms of indirect input. Second, it has been reported that the use of certain assistive devices can be the cause of social stigmatization and embarrassment [42, 18]. Third, the use of common devices like the mouse, trackballs and touchpads seem to be the preferred choice of individuals with mild motor impairments [56]. So, instead on focusing on assistive devices that would intend to substitute the mouse, we would like to direct our efforts on improving and developing assistive technologies that would help individuals with disabilities continue using the devices they have come to associate with the norm.

Of particular interest to us is the population of individuals with fine motor skills impairments. Motor impairments, such as those associated with Parkinson's disease, Cerebral Palsy and Carpal Tunnel Syndrome among others, can have a negative effect in the ability to engage in daily activities [46, 42]. It has been shown that, for individuals with fine motor impairments, the performance of complex simultaneous and sequential movements is much more affected than the performance of simple movements [4]. This translates to difficulties performing pointing tasks on the computer. Individuals with motor impairments provide a spectrum of difficulties that will make the effort of improving and adapting existing assistive technologies exceptionally challenging. If we consider the already existing difficulties that some groups of individuals face when engaging in computer usage activities, we can argue that it would be even more challenging to study and develop help for individuals that also have to face challenges related to motor impairments. Thus, our main goal and the objective of this dissertation is to help individuals with fine motor skills impairments improve their pointing tasks on the computer.

1.1 Why do we need to improve pointing tasks?

Pointing tasks have become ubiquitous in any computing environment and can be a source of frustration to many users if they do not perform these in an accurate and timely manner. In a recent survey on the use of assistive technologies in everyday living [42], about 10% of the participants said that they would be “pleased to use electronic technologies…and computer improvements”. One participant mentioned that such an improvement could be “something to steady the mouse”. We need to understand that in a graphical user interface, the pointer serves as a ”proxy to the users real movements” [21]. Thus, the problem of improving the user’s ability to point when using an indirect pointing device such as the mouse becomes an issue of utmost importance since frustration may come from interacting with a system that is not specifically designed for individuals with impairments.

We can contrast the performance of individuals with motor impairments with populations of individuals such as young children and able-bodied older adults since assistive technologies have been developed and proven to work for both of these populations. This will give us a stating point as we dwell on the difficulties of developing software assistive technologies for individuals with fine motor skills impairments. For example older adults, defined roughly as adults over 60 years of age, are estimated to have nearly 1.5 to 2 times slower movement times than younger individuals [15]. It is also more difficult for older adults to hit a target [57]. Both populations of young children and older adults were found to have difficulties with pointing tasks and effective assistive technologies have shown statistically significant positive effects on target acquisition [24, 11, 25, 23]. An interesting result regarding the performance of skilled tasks by able-bodied individuals states that the decay in performance associated with age is due to a combination of factors such as reduced perceptual feedback, and the employment of different strategies performing skilled tasks [15].

To illustrate this point let us look at figure 1.1 taken from [25]. This figure shows the paths taken by a four year old child and individual with Parkinson's disease performing one of our test studies. In the case of young children, studies have shown that four and five year olds have problems when approaching a target [25]. The data we have gathered from two individuals with Parkinson's disease, indicates that they do not have the same difficulties as children did near the target. When we compare some of the tasks performed by children with tasks performed by individuals with Parkinson's disease it is fairly easy to observe greater difference in movement control away from the target rather than close to the target.

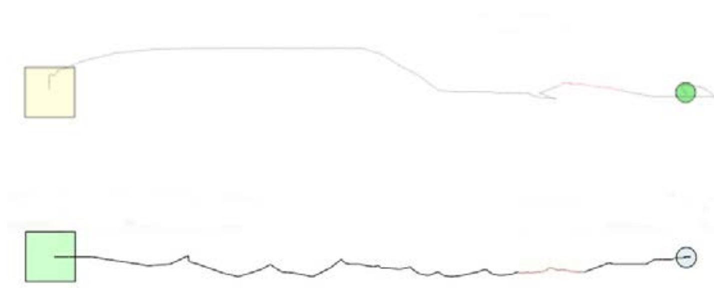


Figure 1.1: Task performed by a four year old (top) and an individual with Parkinson's disease (bottom)

Although speculative, we can assume that because the individual performing this task is a Parkinson's disease patient, the cause for the lack of control in the path towards the target is due to a fine motor skill impairment. Another explanation for this variability, observed in Parkinson's disease patients and other older adults, may be the use of different strategies for approaching a target [34]. In the case of some of our other participants this is particularly true. For example, one of our participants with Cerebral Palsy reported using a combination of a trackpad to approach the target and a mouse to click on the target. Another participant with Parkinson's disease used a modified mouse with a piece of paper blocking the right click to avoid clicking it by mistake. A participant with Carpal Tunnel Syndrome used a mouse with a single button, also to avoid clicking anything but the necessary click.

1.2 Our approach to studying and improving pointing performance

People of different age groups as well as people with disabilities have different skills, different levels of experience using a computer, and therefore different needs when it comes to designing an effective interaction model. In fact, populations of individuals with motor skills impairments show a great variability of pointing difficulties. This allows us to conjecture that the best approach to help individuals with motor impairments is via a personalized adaptation of the assistive technology in question. In the more severe cases, it may be necessary to develop new methods that are able to help with pointing tasks. Our objectives are to apply an existing computer assistive technology in a way that will adapt according to the each individual's difficulties and to develop a new method to help people that show more severe levels of fine motor skills impairments.

We started by analyzing the pointing performance of two individuals with Parkinson's disease. For Parkinson's disease there are effective symptomatic therapies that help improve some users' control over their movements [45]. Different stages of the disease also have different associated levels of motor control. This population is an example group that might benefit from an assistive technology that is specifically adapted to their distinct range of abilities. To identify the difficulties of these diverse populations of individuals, we must study the different ways in which they interact with computers, as well as all the motives that affect their range in motor abilities.

To determine the pointing abilities of individuals we use tools that assess the individual's characteristic movements. From their movement characteristics we can automatically adjust the assistance to the individual's needs, thus improving their overall interaction experience. The problem of improving the performance of an individual's pointing tasks has been addressed with several strategies. We will discuss some of the strategies we found in the literature in Chapter 3. Our first strategy consists of improving the performance by implementing a modified version of *PointAssist*. *PointAssist* works by slowing the speed of the cursor depending on the real-time analysis of the sub-movements of a task. A sub-movement is a smaller component of a complete rapid aimed movement from one point to another. *PointAssist* has been shown to help young children [25] and able-bodied older adults [23] with difficulties associated with target acquisition in terms of movement times and accuracy. One of the features that *PointAssist* implements is providing help to users only when they need it. *PointAssist* requires the user to make an effort before receiving help. Another feature of *PointAssist* is that it does not change the appearance of objects in the screen and thus has no negative impact on the user's overall perceptual experience. It does not require any special hardware, there is no learning curve, and it runs in the background independently of other software.

Most of the assistive technologies found in the literature rely on the precondition that the target is known. These are commonly referred to as target-aware techniques. Determining what a target is on the screen can prove to be challenging in some contexts. In contexts where there are many targets on the screen, having the program be aware of all the objects on the screen can hinder system performance.

However, pointing difficulties can be identified without knowledge of the target's location. We performed a real-time analysis of the pointing tasks by parsing the individual movements into sub-movements. We identified specific difficulties that an individual may have at a much more granular level than what is provided by a qualitative analysis of the overall target acquisition paths. It is possible to identify pointing difficulties by looking at properties of the sub-movements such as number of sub-movements, length, speed and direction. These can serve to develop methods of assistance such as those implemented by *PointAssist* to fit the type and severity of each difficulty.

Analysis of pointing tasks is done in the literature with an empirical approach that relies on Fitts' law. Fitts' law uses the relationship between the movement distance to a target and and the target's size, and it determines the time needed to complete the task. The analysis partaken by studies that rely on Fitts' law do not tend to consider tasks at the sub-movement level. Fitts' law provides tools for determining the index of difficulty of a task and the index of performance of the individual performing the task. However, to analyze tasks in more detail and to be able to personalize the assistance provided by *PointAssist* to each individual, we need to look at the sub-movement level of the pointing tasks. Speed and accuracy trade-off functions used to characterize tasks such as pointing tasks, assume that an individual will perform a number of sub-movements, a primary sub-movement followed by other corrective sub-movements, from an initial position to a target region [39, 3]. We can collect events in a computer system to account for sub-movements and determine when a sub-movement is initiated and when it ends by parsing the events in real-time. A complete movement task, performed by a user with the intent of reaching a target on the screen, will be a collection of sub-movements. The importance of studying sub-movements is that by looking at movement tasks at this level of granularity, the characteristics of the sub-movements provide a more accurate description of an individual's potential pointing behavior. Deterministic models that assume sub-movement parsing of an overall aimed movement can be proven to yield approximately the same movement times as Fitts' law [39].

1.3 Our contribution

One of our main contributions is to implement an effective way of personalizing the help provided by *PointAssist* for individuals with fine motor skills impairments. By analyzing the sub-movements of the tasks we can adjust *PointAssist* and personalize the help provided to meet each individual's characteristics. The results from the case study with two individuals with Parkinson's Disease led us to conclude that a personalized approach adjusting the parameters of the program based on the type and severity of a difficulty is needed due to the variability in performance. We then conducted a study with sixteen participants comprised of a pool of individuals with fine motor skills impairments due to Parkinson's disease, Cerebral Palsy, arthritis, Carpal Tunnel Syndrome, stroke and other motor impairments that affect fine motor skills in an effort to provide assistance using *PointAssist* and implementing a newly designed personalization heuristic. We found the designed heuristic to be statistically significant having an effect of $\chi^2 = 6.4$, df=1, p=.011 on accuracy. We then proceeded to study the long term effects of our difficulty identification methods via a longitudinal study of real-world computer use, conducted with two individuals with Cerebral Palsy and one individual with other physical impairments. Some participants showed marginally statistically significant results in the number of close clicks and the number of sub-movements. One participant had significantly lower sub-movement lengths and speeds with $p < .00001$ and $p < .0002$ respectively suggesting better motor control. Our final contribution is the design and development of a new assistive technology for individuals with severe motor skills impairments that we call the Reverse Funnel. From the informal inquiry on how individuals with severe motor impairments learn to use devices such as the wheelchair, and implementing similar sub-movement analysis as with the previous studies, we came up with the idea of restricting the movement of the cursor on the screen that would effectively "steady the mouse". Three participants were studied and we present an analysis of the effects that the Reverse Funnel had on the participants which yielded positive early results. We also suggest potential future developments.

CHAPTER 2
MODELING POINTING TASKS

2.1 Fitts' Law

It is a common trend in the literature that whenever one wishes to study the movement time of motor tasks, model the performance of an individual in a motor task or measure the motor task's difficulty, Fitts' law is always cited. The basis of the formulation in Fitts' law is the acceptance of information theory as a schema for modeling human behavior. This makes it a key model for interaction design. Fitts' law is derived from a formula that models the transmission of information, and is a consequence of Shannon's theorem on information transmission [37]. The formula for the transmission of information is translated into human transmission of information via movement. The original formulation of signal transmission through a channel looks like this:

$$C = B \log_2 \frac{S+N}{N} \tag{2.1}$$

Here C represents the information capacity, B is the bandwidth, S is the signal power and N is the noise power. The "capacity of the human motor system" in turn can be analogously analyzed using a derived formula from equation 2.1 [37]. Given a motor task, if we could measure the difficulty of the task ID and divide that by the time it takes to complete the task MT, we can obtain a measure of the index of performance IP (See equation 2.2). The index of performance will be equivalent to the information capacity C in formula 2.1.

$$IP = ID/MT \tag{2.2}$$

Comparing equations 2.1 and 2.2, and by appropriately renaming some of the variables the following index of difficulty formula is proposed:

$$ID = \log_2 \frac{A+W}{W} \tag{2.3}$$

Here A stands for the movement total distance and W stands for the “width of the region within which a move terminates”. By simple substitution we can see that

$$IP = (1/MT)\log_2 \frac{A+W}{W} \tag{2.4}$$

Using linear regression to model the relation between MT and ID, the desired index of performance can be obtained from the data. The resulting formula commonly known as Fitts’ law follows,

$$MT = a + b\log_2(A/W + 1) \tag{2.5}$$

It is a relationship that expresses some very intuitive interpretations of physical movements. In the context of pointing tasks, we can consider the target’s size W and distance A from an initial location. A simple hypothesis is that the smaller the target and the longer the distance traversed to reach a target, the harder the pointing task becomes. That is precisely the relation that equation 2.3 suggests. If the amplitude A increases, that is the distance to the target is larger, then the index of difficulty increases. If the size of the target increases, represented in this case by W, then the index of difficulty decreases. Thus, Fitts' law very conveniently models the motor tasks movement time in an empirical way based solely on the measures of distance and size of target.

As we examine people of diverse motor abilities, and we will try to predict their specific motor capacity in order to best deploy an adaptation, at which point the index of difficulty of the tasks they perform will be an important property to determine.

2.2 Measuring Accuracy

In a paper aimed at evaluating different computer pointing devices [31], several accuracy measures where proposed to expand on the traditional movement time and error rate reporting that comes from applying Fitts' law. These accuracy measures might prove useful in determining the pointing difficulties of different individuals. Here is a list of the proposed new measures:

- *Target Re-entry*: When the pointer enters the target region, leaves and re-enters the target region.
- *Task Axis Crossing*: If we imagine a perfect line between the starting position and target, anytime the real movement crosses this line a task axis crossing occurs.
- *Movement Direction Change*: Occurs when the movement path relative to the task axis changes direction. Similarly, if the tangent to the cursor's path is parallel to the task axis.

Figure 2.1: Participant showing Task Axis Crossing

Figure 2.2: Participant showing Target Re-entry

- *Orthogonal Direction Change*: Occurs when two direction changes are present along the axis orthogonal to the task axis. Similarly, the tangent to the cursor's path is perpendicular to the task axis.

In the tasks performed by two individuals with Parkinson's disease we could identify some of the difficulties described above (see figures 2.1, 2.2 and 2.3):

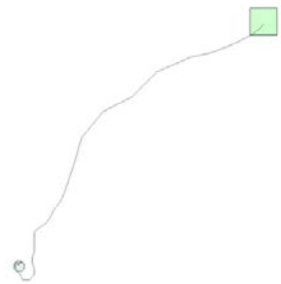


Figure 2.3: Participant showing Movement Direction Change

Other accuracy measures presented in [31] are:

- *Movement Variability*: A continuous measure from the x-y coordinates of the pointer during a movement task.

$$MV = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n-1}} \tag{2.6}$$

Where y_i is the distance from a sample point on the task path to the task axis and $\bar{y}$ is the mean distance of the n sample points. Equation 2.6 is the standard deviation in the distances of the sample points from the mean.

- *Movement Error*: The average deviation of the n sample points from the task axis (above or below the axis).

$$ME = \frac{\sum_{i=1}^{n}|y_i|}{n} \tag{2.7}$$

- *Movement Offset*: The mean deviation of sample points from the task axis. If the task axis is $y = 0$ in equation 2.6 then:

$$MO = \bar{y} \tag{2.8}$$

Two other accuracy measurements added in [34] are,

- *Missed click*: The mouse press and release pair occur outside the target, that is a failed clicked is registered.

- *Ratio of path length to task axis*: The fraction between the total path length and the shortest distance (straight-line) between the starting point of a movement and the center of a target. The closer it is to one, the better.

These measures are quantifications of some type of deviation from the "perfect" path between a starting and a finishing point, and are measurements that can be used to identify different kinds of users. For example in [30] a sub-movement analysis of motor-impaired users found that on average they made five times as many sub-movements as able-bodied users. Other studies like the one in [34], have been done using the previously described accuracy measures. The high variability of the impairments considered and the limited number of participants, only allowed them to describe the differences in cursor control in all user groups. Both of these studies suggest that new movement models are needed to accurately account for several important differences in the pointing behavior of physically impaired users.

CHAPTER 3
STRATEGIES FOR IMPROVING POINTING PERFORMANCE

3.1 Strategies Found in the Literature

As we saw in the previous chapter, Fitts' law is the main tool used to describe movement times of rapid aimed movements. The *optimized initial impulse model* for motor control of rapid aimed movements is used to better explain Fitts' law [3]. This model proposes that an initial impulse movement is made towards a target. The perfect initial movement consist of a single high-velocity movement that reaches the target. If the target is not reached with this movement then other corrective sub-movements are necessary. The approaches we are going to discuss in this chapter try to do one of two things, either decrease the movement distance to reach a target or increase the target's size. By doing either of these, the standard deviation of the endpoint of all movements can be minimized. This deviation is affected by time and movement distance as shown in the following equation,

$$S = k\left(\frac{D}{T}\right) \tag{3.1}$$

Where S is the standard deviation of the endpoints of sub-movements, D is sub-movement total distance, T is sub-movement total duration and k is a constant. By decreasing the movement distance, S is minimized. By increasing the target's size, the number of corrective sub-movements near the target can be decreased. In turn, the movement's total distance, viewed as the addition of all the distances of the sub-movements, is also minimized resulting in smaller S. In conclusion, an optimality

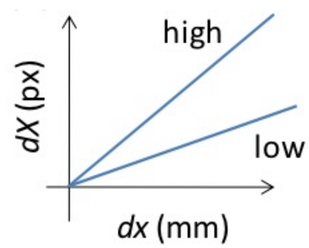


Figure 3.1: Taken from [56]

trade-off between distance and time needs to take place to attain high levels of accuracy.

Two important concepts to discuss are those of the control-display (C-D) ratio and the control-display (C-D) gain. Many of the presented performance improvements such as sticky icons, force enhanced targets and the angle mouse, depend on making changes in this C-D ratio or in the C-D gain. The C-D ratio is the relation dx/dX where dx is the change in distance measured in meters (for the physical interaction), and dX is the change in distance measured in pixels (for the screen interaction). The C-D gain on the other hand is the function used to translate the physical movement into a graphical movement.

A constant or linear C-D gain means that the physical move has a proportionally equivalent move on the screen. Whether this proportion is higher or lower depends on the slope of the gain. With a linear increase in acceleration the ratio dx/dX is also constant (see figure 3.2 (a)), and the C-D gain is $dx = k \cdot dX$ for some constant $k > 0$. If $k = 0$ there is no gain. When $k > 1$ there is a higher C-D gain, and for $0 < k < 1$ the C-D gain is lower (see figure 3.1). A low C-D gain has

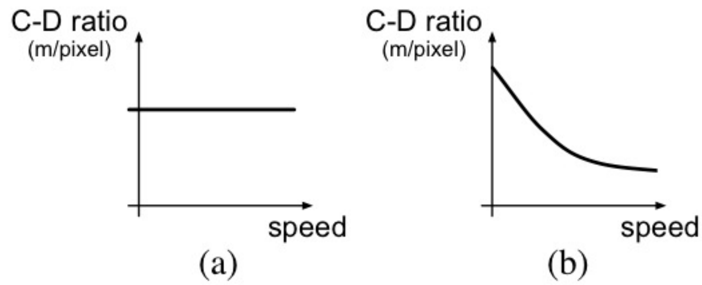


Figure 3.2: C-D ratio as a function of mouse speed (taken from [5])

the effect of making the targets larger in the physical plane because long physical movements translate to proportionally smaller movements on the screen [56]. A common movement adaptation is by adopting mouse acceleration. With this approach longer distances can be achieved with faster movements by decreasing the C-D ratio as acceleration increases (see figure 3.2 (b)). This is similar to say that the C-D gain is increasing, thus non-linear. Then, faster physical movements translate into an increasingly longer graphical movement.

3.1.1 Pointer Ballistics

We begin by discussing a common form of pointing performance improvement which is also included by default with modern Windows operating systems. The "Enhance pointer precision" option found in the mouse options of the Control Panel section (see figure 3.3), is based on the pointer ballistics analysis we found here [41].

The pointer precision feature of Windows relies on a transfer function that maps mouse velocities to pointer velocities. Figure 3.4 shows the inflection points under which the slope of the transfer function line becomes less steep. Having a lower

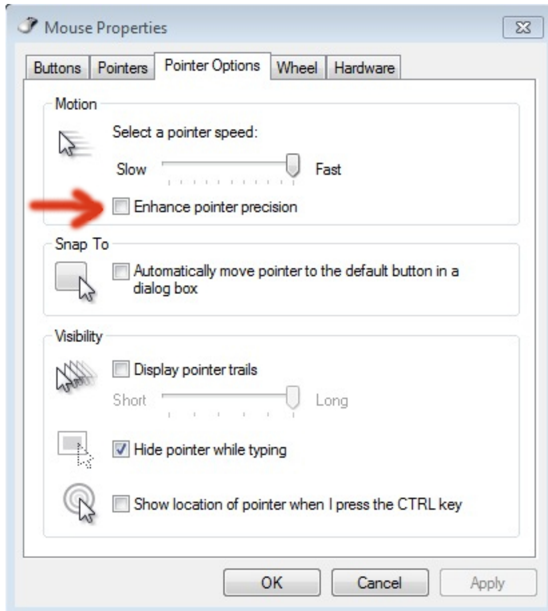


Figure 3.3: Enhance pointer precision in the Control Panel proposed by Microsoft

gain from the mouse velocity to the screen velocity allows for what they refer to as subpixilation which is when the user has to physically move the mouse further than the pointer moves on the screen. Microsoft claims that this achieves a high degree of precision at low velocities. Their acceleration gain algorithm consists of a lookup table that holds the values from the transfer function. Based on the incoming mouse vector magnitudes for X and Y, the translated values are looked up in the table. This is how the translation algorithm works with or without the "Enhance pointer precision" option. If the "Enhance pointer precision is selected" an acceleration multiplier based on the incoming X and Y magnitudes is then calculated and applied to translate the X and Y values.

Users can change the transfer function which is stored in the registry. Again

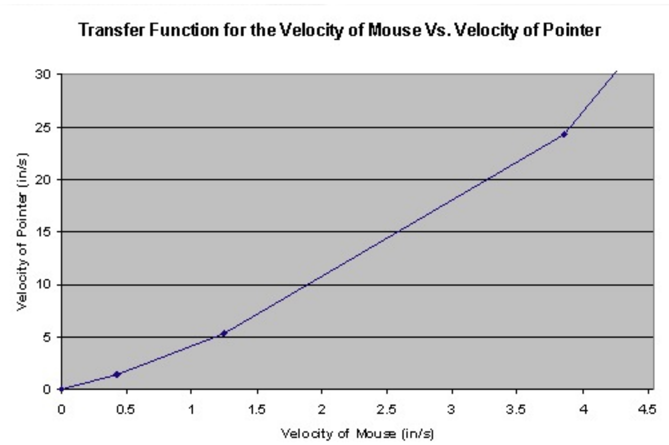


Figure 3.4: Detailed view of the transfer function graph for velocity values under 4 in/s

Microsoft makes the claim that this should allow users to control the ballistics to meet a variety of needs. However we found no research to support their implementation of pointer ballistics to aid people with disabilities. As we can see from figure 3.4, if a user moves with velocities under 4 in/s, the movement distance gain is greatly reduced. While studying one of our participants with Parkinson's disease we came across some interesting effects that the "Enhance pointer precision" may have with users that take more than the average number of sub-movements to reach a target. *PointAssist* works by reducing the speed of the cursor. Under the correct circumstances, an increased deceleration effect might be felt by a user. And in some cases it may result in more sub-movements with shorter distances and longer movement times, especially near a target where deceleration phase is expected.

The correct identification of sub-movement characteristics could be affected

Figure 3.5: a) Enabled “Enhanced pointer precision”; b) Disabled “Enhanced pointer precision”

by this feature. The feature’s impact on the user’s overall performance and the interference that this feature may have with *PointAssist* needs to be tested. As part of our preliminary analysis we have observed differences in the performance of similar tasks for the same individual with an without this feature (see figure 3.5).

3.1.2 Automatic Pointing Assistive Program

The paper on assisting people with an "Automatic Pointing Assistive Program" (APAP) [49], describes a method of assistance aimed at individuals with developmental disabilities. Their biggest claim is that the cursor-capturing functions are not capable of recognizing the desired target. Their solution then is a new mouse driver that intercepts the mouse click action in an expanded region around the target. Their study was conducted with two children aged 7 and 8 years old with mild cognitive impairments. In their action interception method they determined an activation area around the targets such that if a mouse click occurs within this area the cursor would automatically jump to the target inside the activation area and the action was sent to the system as a valid click (see figure 3.6).

Their experimentation methods consisted of three phases where the participants had to go through training, intervention and maintenance periods. Their conclusions were that both participants had improved pointing efficiency, where efficiency for them was defined as the number of successful pointing attempts. Both participants were also able to maintain their newly acquired skills. Our assessment of their implementation is that although efficient for these two young individuals, there are no reports on whether or not this method would be beneficial for older adults or people with motor impairments. Furthermore the idea of having a defined activation area around clickable object means that objects need to be spread out around the screen, otherwise there would be intersection among activation areas potentially creating confusing and annoying behavior of the cursor jumping to undesired objects.

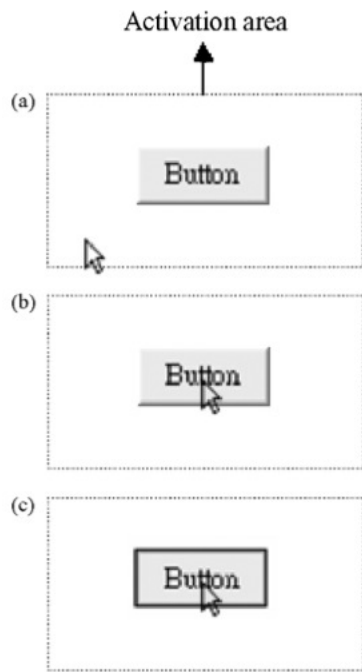


Figure 3.6: Figures show the operation flow of APAP when capturing a mouse click

Because the users needed to go through a training period this also creates a problem for user who do not wish or do not have the capabilities to learn the skills needed. Finally their implementation involves the installation of a new mouse driver which might not be well suited for able-body users.

3.1.3 Ability-Based Interfaces

Ability-Based Interfaces were discussed in [17], where three engines SUPPLE, ARNAULD and SUPPLE++ were studied. The first one generates user interfaces, based on device-specific constraints. It uses a cost function that determines the optimal interface based on the lowest cost determined from the constraints. The second

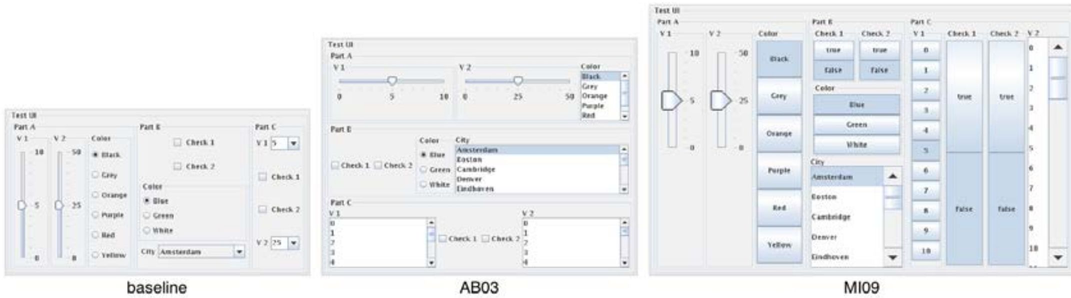


Figure 3.7: Baseline, able-body user (AB03) and motor impaired user (MI09) interfaces automatically generated by SUPPLE

takes as input the user's preferences to generate the interfaces. The third includes an activity modeler that obtains information from the user's motor capabilities based on an initial test, and generates an ability model which can be used as a cost function to generate an interface. Example interfaces generated by SUPPLE and taking into consideration the participants' preferences can be seen in figure 3.7.

Some of the advantages of generating these interfaces include that the user is permitted to input his preferences before an interface is generated. They reported that users preferred the appearance of the newly generated interfaces over the baseline interfaces. The results found by Gajos et al. (2008) were very convincing. Between 8.4% and 42.2% faster movements were reported over all users. Motor impaired users made 73% fewer errors. Users had to perform a preliminary motor assessment test to get the most benefits from the generated interfaces. This probably means that for users with degenerative motor impairments or with a high variability of movement behavior throughout the day, they would need to calibrate the system several times.

This method clearly affects the overall visual arrangement of the objects on the screen, but the fact is that users seemed to prefer the interfaces that the programs generated. There was no discussion about the possible underlying effect that these interfaces may have with the performance of the operating system or any application programs.

3.1.4 Area cursors and the Bubble cursor

The idea behind area cursors is to enlarge the cursor's effective activation area. Pointing cursors are generally single point cursors such as an arrow's tip or the crossing point of a cross-hair cursor. The hypothesis is that using an area cursor to select small targets can be used to slightly change Fitts' law width constraints to include the width of the cursor. Larger area cursors would then yield a lower index of difficulty. Area cursors are also known by the name the"Prince" technique. This follows the similarity of over-sized Prince tennis rackets to area cursors, as far as aiming a large area into a small point goes.

Experiments found that although subjects were slower using area cursor they had better aim [33]. Clear interaction impacts can be noticed from having a larger cursor on the screen. Most notably, large cursors can block objects on the screen. In [57], studies of translucent cursors that do not block the view on the screen did not have a negative effect on performance. When objects are clustered together or are in close proximity, the cursor has to discriminate between interfering objects and the desired target. This imposes that objects on the screen have to be sufficiently

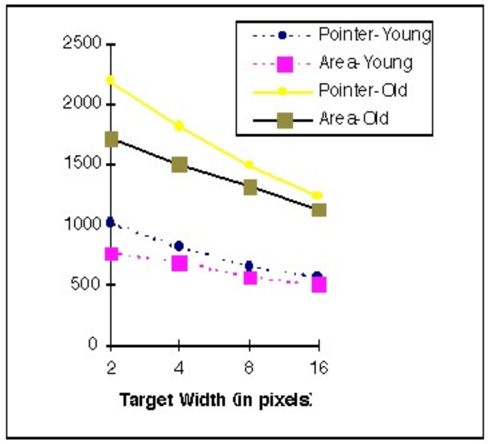


Figure 3.8: Mean positioning time by width of target icon for pointer and area cursor [57]

far apart not to cause confusion about object selection. Again in [57], they studied a slight variation of the cursor where for single isolated targets the cursor behaved as an area cursor and for clustered targets the cursor behaved as a usual. Figure 3.8 shows the advantage of area cursors in terms of movement times relative to the target sizes for old and young users.

Taking the idea from the prince technique, Grossman and Balakrishnan proposed semi-transparent dynamic cursor that changes its activation area depending on the proximity of surrounding targets, changing the effective width of targets in the process [19]. The shape of the cursor being a circle and the dynamic area resizing nature of the cursor gives this technique the "bubble" name (see figure 3.9).

A crucial characteristic of the bubble cursor is that in needs to know exactly

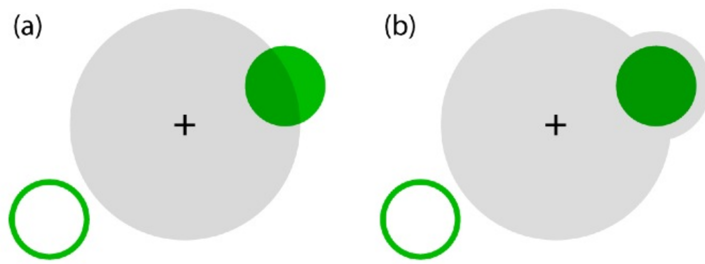


Figure 3.9: The Bubble cursor: The cursor in figure (b) resizes to acquire a single target at a time.

where the surrounding targets are located and what their size is in order to dynamically expand or contract. The effective width of targets is affected by the bubble cursor by dividing the space into regions. Every target resides in a single region where the target is the closest target to every point in the region. This region then becomes the effective width of the targets. The experiments showed that the bubble cursor reduces movement times and under experimental conditions it made selection easier even in clustered areas.

3.1.5 PowerCursor

PowerCursor [55] is a behavior generating toolkit based on the idea of improving cursor interaction by simulating haptic feedback. Haptic perception is the process of recognizing objects by touching them. Indirect pointing devices do not provide this touching effect but it can be simulated visually. The cursor can simulate force-feedback through visual feedback by having the computer displace the cursor position based on apparent surface characteristics on the screen. This method enriches the interaction by adding the visual effect of a tactile dimension. The toolkit facilitates the development of environments where cursor displacement can be applied. The cursor displacement techniques could serve to assist with some pointing problems. For example, if the user has the intent of clicking a radio button, a hole 'felt' in the form of visual feedback, could help the user reach the target more easily. PowerCursor was not necessarily designed for motor impaired users, and the benefits of its use are only suggested in the paper. In fact the developers point at some of the drawbacks of the toolkit in terms of performance and lack of full cursor control on the part of the user. Testing the software to see the effects on pointing performance is an open research question.

3.1.6 Object Pointing

Object pointing, described in [20], is referred to as the 'timorous' cursor because it avoids 'empty' spaces. In essence the new cursor proposed skips what they consider unnecessary empty space in the screen and automatically repositions the cursor for the user. The cursor reposition over a target not only requires knowledge of the targets along the traversed path, but it also requires a continuous analysis of the direction of the movement, the instantaneous velocity of the movement and the acceleration of the movement. One of the main objectives of the design was to avoid the waste of information processing that the system does while moving over so called empty spaces. This is an example of one of the extreme solutions to motion modeling where the movement time between objects is virtually zero being that the space

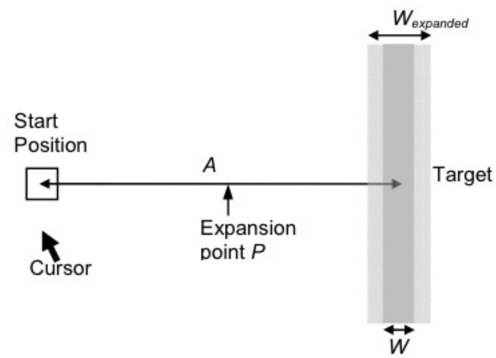


Figure 3.10: W and $W_{expanded}$ are the width and the expanded width respectively [38]

between objects is ignored. This method was proven to facilitate target acquisition. Also, they showed that as pointing becomes more difficult, object pointing becomes more beneficial. The observations in [3] point to some of the shortcomings of this method. First, object pointing is not applicable in all situations as users may want to manipulate the individual pixels. And second, the jumping motions that object pointing imposes on the interaction can prove to be annoying to some users. An alternative to these drawbacks is to switch on demand from regular to object pointing but then the interaction becomes more cumbersome.

3.1.7 Expanding Targets

A relatively common form of expanding targets exists in the MacOS X Dock. There, as the user enters the target's region it grows in size. An experiment that implemented the idea of expanding targets had an onset point in the path towards a target rather than having the target expand when it is entered (see figure 3.10).

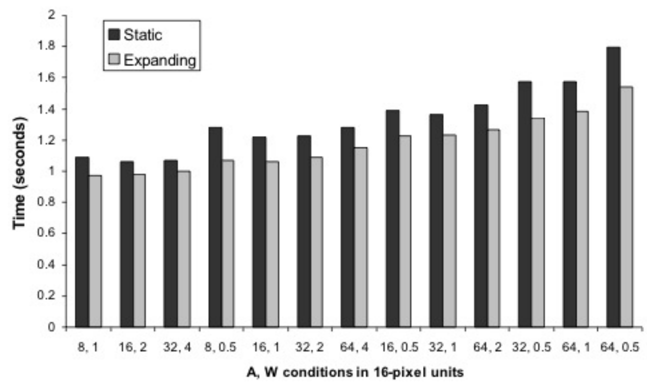


Figure 3.11: Comparison of movement times for static and expanding targets for all subjects

The experiments wanted to answer if the performance of selecting expanding targets could be modeled using Fitts' law and from the model determine the factors that affect this performance. They proved the hypothesis that the movement times would be dependent on the final width of the target and not the initial one at onset of movement [38]. Figure 3.11 shows some of the results comparing static and expanding targets with different combinations of movement length A and target width W with respect to movement times.

Consider that expanding the visual representation does not alter the motor space. Other experiments show improved target acquisition by visual expansion without enlarging the motor space [9]. This study also points to one of the major drawbacks with expanding targets, which is that objects in the screen can be blocked by the expanding targets preventing an effective interaction. This is similar to what happened with area cursors.

Figure 3.12: Image taken from [3]

3.1.8 Proxy Targets

Proxy targets reduce the distance between the cursor and the target bringing the target to the cursor. Thus, this is a method that looks to improve performance by reducing movement time as a result of having the targets closer to the cursor. Figure 3.12 illustrates more clearly this idea.

The study concluded that reducing the distance by means of proxy targets may not be effective to improve performance in terms of movement times for older adults and that further analyses are needed to confirm this [29]. There are benefits from using this approach and they are more apparent in interactions with large or multiple displays. In smaller screens, the visual feedback can clutter the screen causing undesired effects and thus subjective feedback needs to be analyzed as well.

3.1.9 Steady Clicks

Studies have pointed out certain characteristics in the pointing behavior of older adults and people with Parkinson's disease that are not appropriately modeled by the optimized sub-movement model. Some of the movement difficulties that have been suggested not to be appropriately modeled using a sub-movements analysis include slipping of the target, accidental clicks far from the target and accidental button presses other than the left button. The assistance provided by Steady Clicks is aimed at dealing with these issues. Steady Clicks freezes the cursor upon pressing the button and will unfreeze the cursor if either the button is released or the 100pixel threshold is surpassed. To determine when accidental clicks occur there is also .25 pixels per millisecond velocity threshold. Clicks are ignored when the movement occurs above this threshold. Two major assumptions were made to justify the development of Steady Clicks. The first assumed that users would prefer not to concentrate on clicking. The second is that by steadying the cursor users would be more accurate. Some experiments showed fewer mouse presses and improved performance times for some individuals [53, 54]. Participants seemed to prefer Steady Clicks to no assistance, however there were some negative results when it came to dragging.

3.1.10 The Angle Mouse

This is the first target-agnostic approach we encounter aimed at people with disabilities. Target-agnostic techniques are not aware of the targets' locations on the screen. The *Angle Mouse* relies on the analysis of angular deviation of movements. When a movement is "coherent", meaning that is as close to a straight line as possible, there is little angular deviation. The control-display (C-D) gain is adjusted based on the angular deviation of the movements. Essentially the higher the deviation the

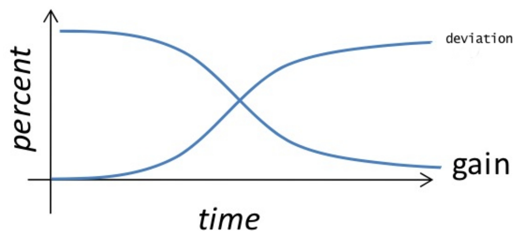


Figure 3.13: Conceptual relationship between angular deviation and C-D gain [56]

lower the gain (see figure 3.13). This C-D gain reduction makes targets larger in the motor-space since a long physical movement is translated into a proportionally shorter screen movement [56]. This is essentially the same effect that the “Enhance pointer precision” has for low mouse speeds. Their goal was to improve pointing performance of motor-impaired users while leaving able-bodied users unaffected.

The experiments conducted to test the *Angle Mouse* consisted in performance comparisons between the default mouse behavior, sticky icons (which we will discuss in a later section) and the *Angle Mouse* itself. Figure 3.14 shows some of the results of these comparisons looking at movement time, error rates, endpoint standard deviation and throughput. One of the main results found with the *Angle Mouse* was in terms of throughput where the throughput was significantly higher with the *Angle Mouse* than with the default mouse or sticky icons [56].

Motor-impaired group

Mouse Type	*MT (ms)*	*Errors (%)*	*SD (px)*	*TP (bits/s)*
Angle Mouse	2014	5.95^{d}	7.35^{s*}	$3.03^{d,s}$
default	2195	7.30	6.99^{s}	2.75
sticky icons	2041	5.85^{d*}	8.23	2.73

Able-bodied group

Mouse Type	*MT (ms)*	*Errors (%)*	*SD (px)*	*TP (bits/s)*
Angle Mouse	1155	9.95	5.64^{s}	4.26^{s*}
default	1146	10.59	5.50^{s}	4.31
sticky icons	1165	10.76	6.70	4.00

^{d}Better than the Windows default mouse (p<.05).
^{s}Better than sticky icons (p<.05).
*Marginal result (p<.10).

Figure 3.14: Results taken from [56] where they note that for all measures except throughput, lower is better.

3.1.11 Semantic Pointing, Sticky Icons and Force Enhanced Targets

Sticky icons and Force enhanced targets follow a similar idea; they both make changes to the C-D ratio. There are other ways mentioned in the literature to create sticky icons by using what they call warping algorithms, but we are going to center our discussion around the C-D gain method. What gives the icon its 'sticky' name is the local CD-gain decrement that can make the effective size of the target larger. Sticky icons automatically reduce the C-D gain once the cursor has entered the target [57]. Force enhanced targets are similar to sticky icons but they expand the effective 'sticky' area of the target beyond its border using a force field.

Figure 3.15 shows the underlying mechanisms of this approach. The idea

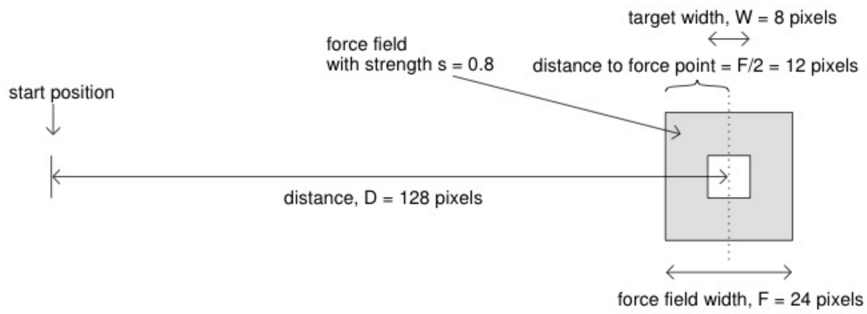


Figure 3.15: A force enhanced selection task (taken from [2])

with this target-aware approach is to create an area around the targets where the C-D ratio is increased gradually, thus lowering the CD-gain the closer you are to the target. This creates the effect of a force field making it harder to leave the target and easier to enter it.

The hypothesis behind semantic pointing is that difficulties in pointing tasks depend on the movement on the physical world and not on the screen representation of a task [5]. Semantic pointing relies on changes in the C-D ratio in a similar fashion as with sticky icons and force fields, except that changes in the C-D ratio are not gradually increased with target proximity. These changes are determined by interpreting C-D ratios as the relative sizes of objects in the physical and graphical spaces. The choice of C-D gain function will depend on the importance that areas of the screen have relative to the user. In this way clickable objects are more important than empty spaces, so for the latter the semantic interpretation, or the distorted motor space, is smaller. For targets on the screen the effective motor space will be larger. This method requires some knowledge of the objects on the screen and affects all objects on the screen.

Results show that sticky icons can be beneficial for older adults especially around smaller targets [57]. However in the presence of distractors, clickable objects in the path towards a desired target, sticky icons reported low improvement error rates and movement times. Force fields show an advantage over sticky icons for users with little experience using pointing devices [2], and they reported a 36% reduced error rate for weak force fields versus a 79% reduced error rates for stronger fields. Both of these methods can produce undesired behavior for some users since it is definitely harder to leave an icon after it is acquired, and in the case of force fields, when targets are in close proximity because of overlapping force fields. This points to the disadvantage that not all pointing tasks can take advantage of these methods.

One important result from the experiments on semantic pointing is that they showed the index of difficulty of a pointing task to be defined by the target size in the physical space rather than the size in the screen space [5]. As with sticky icons, the presence of distractors in semantic pointing increased the distance to the targets.

3.1.12 Adaptive Pointing for Absolute Pointing Devices

Absolute pointing devices rely on a mapping from the motor movement on the physical plane to the on-screen movement in a position-to-position way. This means that the on-screen movement space is proportional to the motor movement. Inherently there is no control-display (C-D) gain when using absolute pointing devices. However the approach taken in [35] is aimed precisely at testing the validity of incorporating a C-D gain mechanism that does not interfere with the visual perception of the absolute pointing device operation. This work is important to us because the approach relies on the optimized initial input model. We do not know to which extent *PointAssist* may work with absolute pointing devices, since typically there is no C-D gain. The results obtained in [35] report a mean reduction in error rate of 63% and a mean difference in movement time of 19% when comparing adaptive pointing to a filtered enhanced absolute pointing mechanism that smooths the pointing behavior.

CHAPTER 4
POINTASSIST

4.1 *PointAssist* as a helping tool

A study of the new techniques for helping with pointing tasks in a GUI argues that even if some of the techniques are promising, none has showed to be effective across all situations [3]. Most of the situations where these techniques fail involve particular aspects of each of the implementations that make the overall user experience less than ideal. Knowledge of the target locations, spatial arrangement of objects in the screen, blocked views of important content or unwanted mouse behavior, are examples of situations that are not properly addressed by most of the techniques. To propose a technique that can handle all possible interactions in a graphical user interface may be impossible but we can make a list of desirable properties that would at least avoid some of the pitfalls that prevent most of the techniques from being universally adopted. We would like for an assistive software to have the following properties.

1. The software should not interfere with the usual arrangement of objects in the screen nor interfere with the visual feedback that objects in the screen may provide. This includes no overlapping of objects in the screen that could cause confusion or annoyance and no new objects in the screen that can blur or block other objects in the screen.
2. The software should not interfere with the regular operation of the operating

system or the installed applications and it should not require any specialized drivers or hardware.

3. The software should not have to know where targets reside on the screen. This is commonly referred to as being target-agnostic. Among other things this avoids problems of determining which objects are clickable in contexts such as text editing where defining a target can be difficult and costly.
4. The assistance should not be continuous, but instead require the user to make an effort to receive help. Requiring the user to make an effort prevents users from approaching pointing tasks with passive cognitive strategies that rely solely on the help provided. With this approach we could study the benefits of motor ability maintenance and improvement.
5. There should not be model training associated with the use of the assistance. This is very important to guarantee the unrestricted adoption of the assistance. This also saves users and researchers from undertaking any training or maintenance phases.
6. The assistance should automatically adapt to every user. It should be able to determine a set of parameters that identify the pointing difficulties of the user and adapt accordingly.
7. The assistance should help a wide variety of users with a ample range of motor disabilities.

Apart from items 6 and 7, all other conditions are satisfied by the current implementation of *PointAssist*. Point 6 is partially addressed by the research done for this dissertation. An automatic implementation of the personalization procedure described in the coming sections could prove to be solution in a future development. Point 7 would allow the assistance to be universally applied. In the case of children, the accuracy achieved by the participants was comparable to the performance of 18 to 22 year olds [25]. The study with older adults helped determine how variable their performance range is compared to children and how different are their strategic approaches to point-and-click tasks. Children and older adults have shown increased accuracy in their point-and-click tasks when using *PointAssist*. In Chapter 7 we will see that the personalization methods we implemented helped a group of users with a wide variety of motor impairments. Considering the versatility *PointAssist* has shown in helping young children, older adults and individuals with disabilities, we argue that it is the closest technology to what we propose to be an ideal assistive technology.

All the studies that have been made with *PointAssist* rely on the underlying analysis of the sub-movements and their characteristics in real time. The study of sub-movements has its theoretical roots in the hypothesis made by Meyer et al. in [39]. Their hypothesis was that the noise associated with neuromotor responses to visuospatial feedback and the generation of rapid aimed movements, may be characterized by the durations of component sub-movements. Under their proposed model, certain characteristics of the sub-movements were not affected by the visual feedback during a user's movements. This suggests that the study of the characteristics of the sub-movements could potentially identify pointing difficulties across different populations regardless of their visuomotor adaptation skills. Other studies suggest that the optimized sub-movement model cannot properly model certain movement difficulties like pauses and slip-offs, and that new models for motor-impaired users need to be explored [30]. Nevertheless, we intend to further explore the applicability of the optimized sub-movement model to help alleviate some of the difficulties with target acquisition individuals with motor impairments might have by identifying different characteristics of the sub-movements of the users and adapting the help provided by *PointAssist* for each individual.

PointAssist works by parsing cursor motion into sub-movements as presented in [22]. A sub-movement potentially begins (also meaning that if movement has begun, a previous sub-movement ends) if either of these conditions holds,

* There is a change in direction. This is done by classifying mouse event endings in four different quadrants (right, left, up and down), where two consecutive mouse event endings in two different quadrants determine a directional change.
* There is a change in acceleration from negative to positive. A change in acceleration may indicate the proximity to a target. Alternatively, at the end of a sub-movement a deceleration phase is expected where the acceleration is negative and at the beginning of a new sub-movement a positive acceleration indicates a new sub-movement.
* A relative minimum in the absolute acceleration values while the acceleration is negative. The analysis of movement paths suggests that while moving along large distances there will be peak velocities achieved followed by a deceleration phase. Near targets the deceleration phase may be longer, and a new sub-

> movement near a target does not necessarily achieve positive acceleration. The relative minimum in the deceleration phase is a good indication that a new sub-movement is about to start.

Sub-movements are defined as movements of at least 4 pixels in length and a minimum speed of 0.02 pixels per millisecond. A 50 millisecond minimum is also required to take into account possible noise due to variability in the neuromotor adaptation. The proximity to a target is predicted by the changes in velocity and sub-movement lengths. It is assumed that near a target the sub-movements will be short and slow, according to the theory that a series of corrective sub-movements accompanied by a deceleration phase take place just before a click.

A real-time post-parsing of the sub-movements takes place to determine which sub-movements present characteristics consistent with difficulties near a target. Consecutive valid sub-movements are defined to be difficult depending on the length and speed parameters. These are the parameters that may need adjustment for different individuals. Determining the correct parameters for an individual allows for a more accurate identification of the difficult sub-movements that will trigger the assistance. Once a pair of consecutive difficult sub-movements is found *PointAssist* reduces the speed of the cursor by a factor of 2. This speed reduction mechanism is the assistance that *PointAssist* provides. The speed reduction mechanism is non-cumulative, it happens once a pair of difficult sub-movements is observed and is not triggered again until disabled by encountering a non-difficult sub-movement.

From the implementation of the sub-movement parsing algorithm we observe that since triggering the assistance should take place upon identification of pointing difficulties, able-bodied users should trigger it less than less able users. However, if the user is not trying to be accurate, *PointAssist* could fail to trigger the assistance. Thus, *PointAssist* emphasises that if a user needs help he or she needs to make an effort in their movements for the assistance to trigger.

4.1.1 Helping children point with ease

A study with 30 four year old children was conducted in [25] with the previously described mechanisms for identifying pointing difficulties. Tests were similar to those depicted in figures 1.1 and 3.5. The main goal was to assess the level of help that *PointAssist* could provide for these children while performing point-and-click tasks with a mouse. The target size affected accuracy, but distance to target did not. This was mainly because children were found to make more sub-movements near a target than away from the target. Fitts' law analysis confirmed that the movement behaviour of children was as expected by the model for movements away from the target. Figure 4.1 shows the overall movement behavior of children near target as reported on [25] .

Accuracy rates were increased by 12% in some of the trials, with ever higher percentages in other trials. Smaller targets of 16 pixels showed the most improvement, where *PointAssist* had an effect in target re-entry with $p < 0.01$, and an effect on accuracy with $p < 0.001$. Distance to target was varied between 128 and 512 pixels and no statistically significant results were found regarding distance variability with

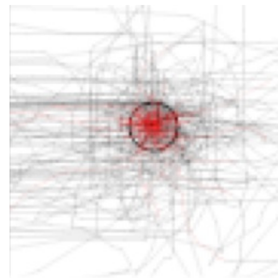


Figure 4.1: Participants' paths to a target of 16pixels in diameter. Red lines indicate sub-movements where the assistance was triggered. [25] (Red lines are only visible in color print. In the black and white version, a clustered gray area in the center of the target corresponds to the red lines in the color version.)

respect to either accuracy or target re-entry.

4.1.2 Helping older adults point with ease

Another study with *PointAssist* was conducted with twenty adults in the range of 66-88 years old. One difference between this study and the one with children was that the goal was to make a comparison between *PointAssist* and the "Enhance pointer precision" option discussed in chapter 3. There were indications of improved accuracy with *PointAssist* over the use of this option. The experimental setup and the parameter values used to parse the sub-movements were similar as those used in the study with children. The population sample was composed of highly educated older adults. Results obtained using repeated measures ANOVA as well as Friedman's test showed statistical significant effect of *PointAssist* on click accuracy ($F(1, 20) = .033, phi = .5$) but no significant results were found with respect to movement times, target re-entry or number of sub-movements [23].

CHAPTER 5
MOTOR IMPAIRMENTS

The study of the different tremors, motor symptoms and other motor disabilities is important if we want to identify and help with the distinct difficulties that individuals may present while performing pointing task. Identifying some difficulties will help us adjust and personalize our pointing improvement mechanisms for each individual. Let us then examine the diagnosis, care and symptoms associated with the different motor impairment conditions that we found in the population of individuals we studied.

5.1 Parkinson's Disease

5.1.1 Diagnosis and Care

Parkinson's disease is a brain disease that impairs motor control, speech, and other functions. It is known to be chronic, degenerating and progressive. Some of the signs and symptoms of Parkinson's Disease are proven to impair motor dexterity when performing actions that require a high degree of skill and hand coordination, thus making it of particular interest for our purposes. An individual with Parkinson's disease presents symptoms in a range of conditions called movement disorders, that are classified and identified as follows [46, 43, 51],

- *Distal resting tremor:* A resting tremor is the continuous involuntary movement that takes place when the individuals hands are at rest. The measured average range of the movement is known to be between 3-6 Hz. When the muscle is at rest, it exhibits the minimal frequency and higher frequencies are achieved when the muscle is voluntarily moved.

- *Bradykinesia/Akinesia:* Bradykinesia refers to the motor impairment characterized by slow movements. Because of the deteriorating nature of the disease, bradykinesia may progress into akinesia which is the failure to initiate movement or complete lack of movement. Rapid and sequential tapping of fingers becomes difficult. Rigidity is also a sign and it is observed when there is resistance while attempting to flex or extend arms at the elbow.
- *Postural instability and postural tremor:* Postural instability is reflected in the lack of balance while holding a posture. Similarly, postural tremor occurs when a limb is resting against gravity. Involuntary reflexes when trying to maintain balance or when trying to maintain a posture may cause the individual to fall. When an individual is unable to stay in a still position, or is unable to sit still or remain motionless, is also known as Akathisia.
- *Gait disturbance:* Defined as uncontrollable problems when walking. Characterized by small steps or shuffling steps that may make it hard for an individual to avoid obstacles, have poor balance and have difficulty turning.
- *Cognitive dysfunction:* The inability to process information may be reflected in problem when communicating, language problems, speech impediments and even depression in some cases.
- *Other motor symptoms:* A non-exhaustive list of other motor symptoms includes Dystonia which refers to twisting muscle contractions. It often affects

the feet and ankles and may interfere with gait; problems swallowing or Dysphagia; soft, monotonous speech or Hypophonia; rapid speech known as festinating speech; fatigue; infrequent blinking; and difficulty rising while seated among others.

The diagnosis of Parkinson's disease is often done clinically with patients presenting some of the symptoms discussed previously, together with shown positive response to levodopa treatment. Ledovopa is an antiparkinsonian drug used for symptomatic therapy [8]. Levodopa proves very effective against symptoms like bradykinesia and akinesia in early stages of the disease. Treatment with levodopa wears off in about 40 percent of patients after 5 years [46]. These patients have a characteristic effect of unpredictable variations in their motor skills while medication is effective, commonly referred to as the "on-off" effect. To counter the wearing effect from long term exposure to Levodopa, certain dopamine agonists are administered. Brain stimulation through surgery is another treatment for Parkinson's disease. Surgical procedures can improve motor control and functionality and reduce the need for antiparkinsonian medications [46].

Different kinds of tremors are classified by the cause and the clinical features. Notice that not all tremors are a result of Parkinson's disease. *Essential or benign tremor* seems to be the more common. It is benign and may have no progression or slow progression. Essential tremor can be characterized by constant noticeable postural or kinetic tremor. Bradikinesia, akinesia and postural instability are not part of the essential tremor [51]. It may become more severe and less frequent with time. Hands show the more visible effects of essential tremor. Emotional stress and physical exhaustion may heighten the tremor. Symptoms may start appearing at any age but it is most commonly seen past age 40. There is a 50% chance that essential tremor may be inherited [43]. *Parkinsonian or resting tremor* often precedes Parkinson's disease. The effect can also be observed in the hands as the patient attempts to lay them to rest. It is characterized by pill-rolling movements of the hands [46, 43] Stress and emotional responses can increase the tremor. The frequency of the essential tremor varies from 3 to 6 Hz [46]. It generally starts with patients over 60 years old. Like essential tremor it can affect both sides of the body. *Dystonic tremor* is seen in individuals regardless of age. Caused by *dystonia*, in which sustained muscle contractions cause repetitive movements, abnormal postures, involuntary twist or abnormal curving when an individual attempts to move or assume a posture. *Cerebellar tremor* is a tremor caused by damage to the cerebellum, and may be due to other conditions other than Parkinson's disease such as a stroke, a tumor, alcoholism or the abuse of certain medicines [43]. The damage may cause a mix of different kinds of tremors like resting, moving and postural tremors. Movement is most affected when an individual performs a directed voluntary movement. The side of the body that exhibits the tremor indicates the side of the brain that suffered the damage or lesion. For a more comprehensive list of tremors and their related symptoms refer to [43].

5.2 Other Physical Impairments

Thus far we have only discussed some of the issues related to individuals with Parkinson's disease. However, there are a series of health conditions that involve motor and or physical impairment of some kind that may interfere with the everyday use of computers. These are not necessarily related to Parkinson's disease (PD) and we need to consider them since our participants' impairments are not all due to PD. The term impairment is defined as the "loss or abnormality of body structure or of a physiological or psychological function" [52]. We have briefly mentioned that tremors, for example, need not be caused by Parkinson's disease. We can also mention other conditions such as arthritis, trauma, spinal chord injuries, carpal tunnel syndrome, brain damage due to stroke and Cerebral Palsy, that impair the user's ability to interact with the computer in an effective way. It is a fact that pointing and clicking are at the base of any modern computer interaction. People with physical impairments face a challenge in these kind of tasks, especially if they lack the ability or have a reduced ability to point and click.

Physical impairments that affect movement are classified as movement disorders. Those associated with Parkinson's disease are know as parkinsonism, although some parkinsonism symptoms can have other underlying causes. Some other common movement disorders are [52],

- *Ataxia*: lack of coordination resulting in unsteady and awkward motion. Caused mainly by damaged cells in the central nervous system.
- *Chorea or Choreia*: involuntary and irregularly quick movements. Caused

mainly by Huntington's disease.

- *Dystonia*: involuntary contractions of the muscles causing repetitive movements, abnormal postures and pain.
- *Myoclonus*: involuntary twitching of the muscles. Primary causes are brain lesions, spinal chord injuries, multiple sclerosis, Parkinson's disease and Alzheimer's disease.
- *Paresis or Paralysis*: partial or complete loss of muscle movement. Caused by brain lesions and spinal chord injuries.
- *Spasm*: sudden involuntary contraction of the muscles. Caused by mixed signals from the nervous system to the muscles. Other causes include cerebral palsy, brain lesions, spinal chord injuries and stroke.
- *Tremors*: unintentional oscillating muscle movements. Essential tremor of the hands has a prevalence of 4% in the United states. Tremors can be the cause of PD, brain injuries and stroke and can be triggered by medication.

Prevalence of movement disorders in the United Stated are reported in [52]. A 2.6% prevalence of strokes leading to a dexterity impairment is found in older adults. Between 1 to 1.5 million people suffer from Parkinson's disease, 400,00 from multiple sclerosis and 253,000 from spinal cord injuries. Reported worldwide prevalences of 9.3% and 26.9% represent cases of carpal tunnel syndrome and trauma disorders. These numbers make a compelling case for improving existing assistive technologies, like *PointAssist*, to aid these individuals in a personalized manner.

Let us ponder some of the potential difficulties in pointing and clicking for individuals suffering from these motor impairments. Again as mentioned in [52], people with arthritis may have a hard time performing certain movements. People with ataxia or tremors can have a difficult time following a precise path towards a target. People with akinesia or bradykinesia can have problems starting a movement as well as ending a movement, potentially affecting their overall time completion and their responsiveness to visual feedback. People with tremors have a tendency to slip away from the target because they click more slowly or may accidentally click an undesired target on the screen or an undesired button on the input device. Abnormal postures caused by dystonia may prevent the individual from using their hands at all, causing them to rely on knuckles and extensive arm movements to perform some of these tasks. For example, one of our participants employs only the index finger on the left hand to perform any sort of movement on the screen due to an abnormal posture. Carpal tunnel syndrome causes extreme pain, lack of sensation and grip force lack of coordination [36]. These are some of the challenges we face when trying to determine the characteristics of the movements of these individuals. Let us now discuss some of the research that has been conducted to develop assistive technologies for Parkinson's disease patients as well as for people with other fine motor skills impairments.

5.2.1 Visuomotor Adaptation and Cognitive Processes

Visuomotor adaptation refers to the process by which the visual input of the target's location is converted into a motor command. Performance can be affected by the motor space, the visual space and the control-display (C-D) function that controls the gain in transferring the motor into the visual space. Dexterous movements are required to perform certain tasks with input devices. For both disabled individuals and older adults it can be difficult to perform these kinds of movements [12]. According to [32], there are results that suggest impaired visuomotor adaptation in patients with Parkinson's Disease. In a study by Teulings, Contreras-Vidal, Stelmach and Adler [50], Parkinson's disease patients were compared to control groups of elderly and young adults in a series of handwriting exercises. These exercises involved perturbations on the visual feedback from normal sized to reduced size to enlarged size, and back to normal size. The study suggested that there was no visuomotor adaptation of Parkinson's disease patients as a result of the visual feedback adaptation. Cognitive function studies also propose that patients with fine motor skills disabilities exhibit 'spacial' deficits [7]. A study was conducted in [6], where a series of visuomotor, visuospacial and visuoperceptive tests concluded that Parkinson's disease patients not only had visuospatial impairments performing complex motor tasks, but also on visuoperceptual tasks where no motor tasks needed to be performed. Another experiment [10] was conducted where patients had to trace the paths towards carefully selected target circles on the screen using an infrared marker over the surface of a table in front of the computer. The experiment concluded that in comparison to similar age control groups, patients with motor skills impairments did indeed exhibit visuomotor adaptations.

A study on cognitive processes asserts that perception is subjective and context dependent [47]. Some authors suggest that impairments with operations that involve spacial perceptions are due to orientation shifts of mental perspective [7]. Brown and Marsden [7] found that mental shifting occurred in individuals with Parkinson's disease in tests that did not involve any spatial perceptive skills. A mental shift impairment involving a cognitive strategy is not necessarily a spatial impairment. These shifts may lead to perceived reduced motor performance, but should not be considered a spatial impairment. Even with the previously discussed experiments where it was suggested that individuals with fine motor skills impairments may exhibit visuomotor adaptation, there does not seem to be any absolute evidence of a spacial deficit [7] that could be generalized across the population of individuals with conditions that affect their fine motor skills. In the experiment Brown and Marsden conducted on mildly symptomatic patients with Parkinson's Disease, with no visible signs of other factors that could bias their results, they concluded that there was no apparent impairment related to the aspects of spatial functions. Because of the subjectivity of perception, we still need to be aware of the potential spatial impairments that the disease may impose on the individuals while performing the tasks in our experiments. Perhaps, as suggested by Brown and Marsden, simpler and easier to interpret experiments need to be conducted, if we are going to accurately and intelligently identify new pointing problems. Although we are not trying to prove or disprove whether individuals with disabilities have visuomotor impairment or spacial deficits, we are trying to identify the characteristics of the sub-movements of these individuals while performing tasks that involve visuomotor and spacial adaptations. If we can successfully identify these characteristics, we can differentiate this population from other known user groups, namely children and older adults. This could lead us in the correct direction to a better version of the identification of the pointing characteristics of any user and thus yield better results in the personalization process of the assistive technology. However we need to be careful not to select a set of evaluating measures that incorrectly identify difficulties in the pointing tasks of individuals with motor impairments, as they may or may not have visuomotor impairments. This concludes our brief overview of the symptoms, diagnostic, care, tremors and other motor impairments associated with the diseases and conditions of the participants we studied.

5.3 Further discussion on improving pointing performance for people with disabilities

One example to help users with the use of a mouse comes in the form of an adapter that intercepts signal sent from the mouse to the computer. It identifies tremors and attenuates the effect before the signal reaches the computer. It has been patented under United States Patent 6561993. A more detailed description can be found on the website at http://www.freepatentsonline.com/6561993.html.

> "A system and method for minimizing essential tremor effects while utilizing a pointing device on a computer system is disclosed . . . The system and method includes a software tuning algorithm used to obtain an individual's tremor characteristics . . . The modified device driver will filter the pointing device input data based on the filter coefficients and eliminate tremor effects from the on-screen pointer. Because the profile is transfer-

> able, if a device driver capable of accepting the profile plug-in were already installed on a computer, the profile could be loaded and used immediately on the computer without the need for re-calibration." [1]

Devices like these are not apt for all types of motor impairments, and even though the user profiles are transferable from one computer to another, the user has to manually adjust the level of movement "smoothness" that they wish to obtain from the device. This device includes a combination of hardware and software and in some cases, if the individual is not knowledgeable enough, it may lead to an eventual misuse of the device. It has been argued that some of the reasons why older adults under-use assistive devices are that they do not receive enough information on the use of the device, and that they do not know how to replace the devices in case they are damaged [18]. This kind of device assumes the user must learn to use and adapt to the device, if they are to maximize the benefits of its use. This is one reason why we are more inclined towards software based assistance to pointing performance, preferably if the assistance requires little or no knowledge of its use from the part of the user. We will discuss several such software based assistive technologies in Chapter 3. Although many of these are not specifically designed for individuals with Parkinson's disease, they are important because they illustrate the different design mechanisms used to improve the performance of pointing tasks.

Another study was conducted in [48] using a mouse with a wheel and specialized software. Their interest was to help people with multiple disabilities and minimal motor behavior. The software called Dynamic Pointing Assistive Program (DPAP), consisted of a redesigned mouse driver that would intercept the intent of clicking on a target based on mouse wheel movement. It would automatically reposition the cursor on the desired target based on the rotations of the mouse wheel. This approach was particularly effective for people with reduced movement capabilities, some which made use of body parts other than the hands to operate the mouse and the mouse wheel. This approach is another combination of hardware and specialized software that may not be suitable for all individuals and requires the individual to train before using the help. It also requires that the individual maintains the newly acquired skills via practicing. The question of whether or not their method can help individuals with other disabilities like tremors in the case of Parkinson's disease, is still unanswered. It might be very hard for an individual with a persistent tremor to effectively control the mouse wheel. But this is also true of the regular mouse. If an individual is completely incapable of moving the mouse, alternative hardware aides are needed. The effectiveness of these hardware-software combination approaches seems to depend on the type and severity of the condition where software alone might not be able to help the individual.

It should be clear by now that individuals with motor impairments present a particular challenge in identifying the characteristics of the pointing movements. Because of its degenerating nature we expect Parkinson's disease individuals to behave differently than individuals with Cerebral Palsy that is non-degenerating, or Carpal Tunnel Syndrome which may be temporary. Because of the wide range of movement disorders associated with these conditions and because of the variability in the severity of the symptoms that patients present, like the “on” or “off” periods in the case of Parkinson’s disease we present a method to personalize the assistance provided by *PointAssist* for all individuals studied.

CHAPTER 6
IDENTIFICATION OF POINTING DIFFICULTIES OF TWO INDIVIDUALS WITH PARKINSONS DISEASE VIA A SUB-MOVEMENT ANALYSIS

6.1 Preliminary study with two Parkinson's patients

We extended the pointing task testing software used in [25] and [23] to gather data remotely where participants could install the testing software into their own computers. The software was also modified to present tasks in eight different directions. We decided to gather data remotely because we want to assist PD patients in the computers they use every day, which in some cases may include customizations to address their needs, as was the case with the two participants we worked with. It proved very difficult to find participants, but we recruited two individuals and informed consent forms were obtained. To maintain anonymity we will refer to the participants as Bob and Dave.

Bob is a right handed 64-year-old male with PD who averages 6 hours per week of computer usage, and uses a touchpad. Dave is a 72-year-old male with PD who uses a two-button mouse an hour per day on average. We took several rounds of data to better account for individual variability. Bob ran the test 3 times with target diameter sizes of 16 and 32 pixels and distances to the target of 128 and 512 pixels. Each test had 4 practice tasks and two blocks of 32 tasks. The cursor speed was set to 8 (corresponds to the fifth tick from the left in Windows). He reported having Enhanced pointer precision (EPP) enabled [41]. Dave ran the test once for same target diameter sizes as Bob but with one distance to the target of 512 pixels and EPP disabled. He had 5 practice tasks and 4 blocks of 16 tasks. The participants ran the test one last time with target diameter sizes of 8 and 16 pixels, task lengths of 512 pixels and EPP disabled, for a total of 69 tasks.

Bob, being our first participant, initially ran the software with the same parameters used for children and older adults in [25] and [23]. Software modifications were required and we decided to ask Bob to run the software two more times. Upon Dave's recruitment, we already had an established experiment that did not need any modifications, which explains the difference in task collection between both participants. The final testing round was to have a more homogeneous set of data to compare the participants.

We characterize difficulties using accuracy measures and quantitative characteristics of the sub-movements of all tasks. We looked at sub-movement length (pixels), average sub-movement speeds (pixels/ms), average sub-movement maximum speeds, direction, number of sub-movements per task, target re-entry, and average task duration (milliseconds). We consider a sub-movement being near a target if it is less than 30 pixels away from its center and away from the target if it is more than 60 pixels away from its center.

Bob showed difficulties reaching the target. He had high movement times (Table 6.1) and averaged 12 sub-movements per task (Table 6.2) in tasks involving 512 pixel target distances, with a click success rate of 97% (Table 6.1). Thus, he seems

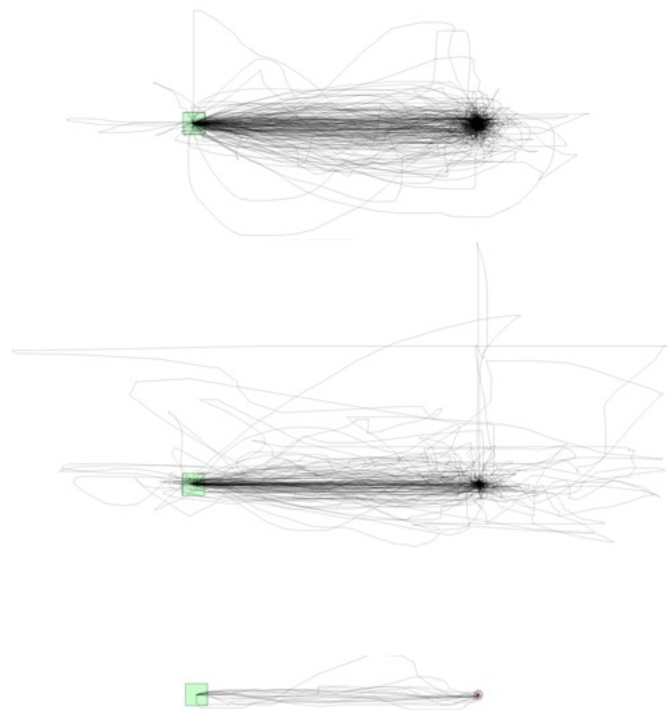


Figure 6.1: Paths taken by children (top), older adults (middle), and PD patients (bottom). Target diameter sizes of 16 pixels and movement distance of 512 pixels.

Table 6.1: Accuracy measures comparisons for 16 | 32 pixel target diameters.

	target re-entry	avg. task duration	click success
Children	2.8 | 1.7	6919 | 4542	80% | 88%
Older Adults	1.4 | 1.2	3149 | 2786	91% | 88%
Bob	1.7 | 1.6	8021 | 8014	97% | 100%
Dave	2.5 | 2.7	3835 | 2763	93% | 100%

Table 6.2: Number of sub-movements near and away from target center, task movement distances of 128 and 512 pixels.

	128 pixels tasks	512 pixels tasks
Children	4.2 | 2.3	5.1 | 3.8
Older Adults	2.8 | 2.2	3.1 | 4.7
Bob	2.5 | 3.1	3.0 | 12.2
Dave	no data	2.5 | 3.2

Table 6.3: Sub-movement characteristics near and away the target center.

	avg. length	avg. speed	avg. max speed
Children	28.5 | 95.3	.075 | .271	.188 | .509
Older Adults	38.6 | 82.6	.115 | .329	.255 | .620
Bob	28.7 | 158.7	.115 | .588	.212 | 1.01
Dave	45.5 | 128.9	.078 | .201	.172 | .358

Table 6.4: Accuracy measures comparisons for 8 | 16 pixels.

	target re-entry	avg. task duration	click success
Bob	2.1 | 2.0	3994 | 3990	75% | 100%
Dave	1.8 | 1.2	4209 | 3836	88% | 94%

Table 6.5: Number of sub-movements near and away the target center, 8 and 16 pixel target diameters, 512 pixels movement distance.

	8 pixel targets	16 pixel targets
Bob	3.8 | 2.0	4.3 | 1.9
Dave	3.0 | 2.1	2.9 | 2.2

accurate but slow. He had particular difficulty in the north and north-east task directions where there was a combination of high sub-movement count and high task duration. His high movement times (task duration) could be attributed to having EPP enabled, but two things suggest that the problem is elsewhere. First, his average speed away from the target was higher than that of older adults (Table 6.3), who also had EPP enabled [23]. Yet older adults had movement times of less than half as those of Bob (Table 6.1). Second, the combination of high average speed and average length (Table 6.3) of Bob's sub-movements suggest he is not really moving slowly but rather may have difficulty initiating his movements which could account for the high movement times. In fact we know he is not moving slowly since he had the highest average speed away from the target (Table 6.3). From this, we infer that Bob has some level of akinesia that is most prominent in some directions, affecting his movement time and speed, but not his accuracy.

Dave had no problems reaching the target. His average task durations are comparable to those of older adults (see 6.1). His high number of target re-entries (Table 6.1) shows his difficulties are near the target, and is comparable to that of young children that show more, less accurate sub-movements [25]. This is consistent with his average sub-movement speeds being close to that of children (Table 6.3), but not with the average sub-movement length. This is where Dave showed most difficulty, having the highest average sub-movement length of 45.5 pixels near a target. We attribute this lack of control near the target to tremors.

Pointing tasks examples show differences in pointing strategies employed by children, older adults and our two participants (see Figure 6.1). Children's movements tend to cluster around the target. Older adults tended to land short of the target, then slowly get closer, with some difficulties keeping a steady direction towards the target. Bob and Dave's movements are relatively controlled suggesting they are consciously making an effort to have aimed controlled movements.

Bob and Dave performed differently from each other. In the first rounds of testing Bob had high movement times, and Dave had lower movement times (Table 6.1); Dave re-entered targets almost twice as many times as Bob (Table 6.1); Bob had a high number of sub-movements away from target where Dave could complete a task in approximately 6 sub-movements. A high number of sub-movements were also noted in [30] where individuals with PD took five times as many sub-moves as able-bodied users. We see the variability in their performance in the second round of testing. Bob increased his target re-entry where Dave decreased it (Table 6.4). Bob took as many as 4 sub-movements to reach the target, which was 3 times less than in his first round (Table 6.5). Dave was consistent in his number of sub-movements yet his accuracy dropped by 4-6% (Table 6.4).

Our results show some of the difficulties as well as the variability between both participants. Differences between them can be attributed to different motor impairments and/or levels of motor control. Variability differences of each participant's performance were observed over time and can be attributed to unknown factors like on-off periods, strategies employed or habituation effects.

6.2 *PointAssist* and Parkinson's Disease: The need for personalization.

To make more accurate assessments we added the independent variable of direction. In neither of the previous studies with *PointAssist* was direction a variable under consideration. However, movements of the individuals with motor impairments are very different from the other individuals that have been studied using *PointAssist*, namely children and older adults. Therefore, direction might be an issue that affects performance as an individual with motor impairments may find it more difficult to move in certain directions and he/she may adopt different strategies depending on the direction of the movement. Figure 6.2 serves to illustrate that direction plays an important role in identifying difficulties. Notice that the paths away from the target are very different in each of the directions considered.

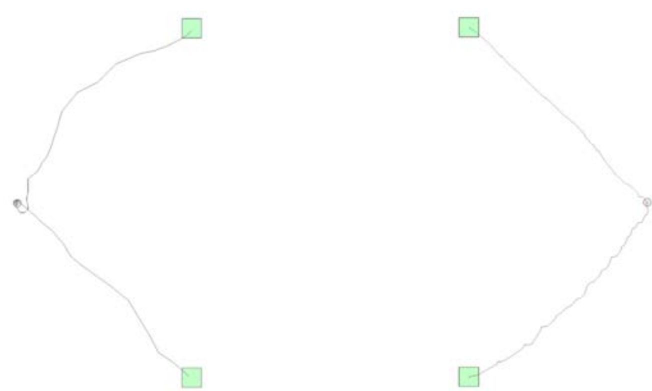


Figure 6.2: Task performed in different directions by a Parkinson's disease patient with 16 pixel targets (The starting points are the squares).

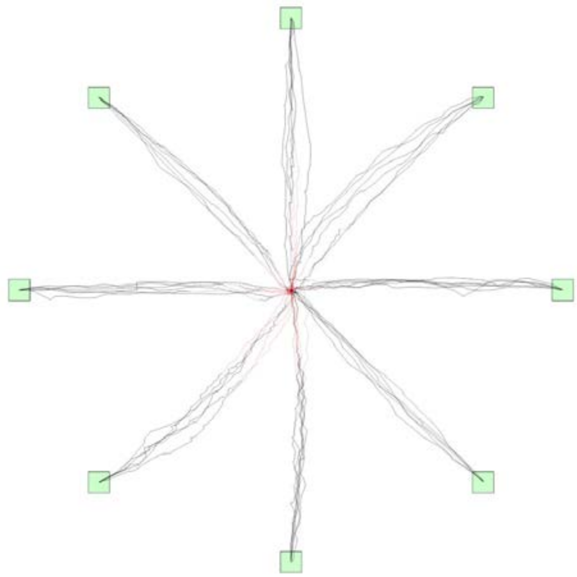


Figure 6.3: Paths taken by Bob in all directions with 16 pixel diameter targets and 512 pixels movement distance.

Motor impairments such as bradykinesia and akynisea can have effects in the direction of the movement, so an individual may find it easier to move in an east-west direction but have difficulties initiating or completing a west-east movement. Eight directions were considered and were implemented in the testing software, namely north, north-east, east, south-east, south, south-west, west and north-west.

Examples of pointing tasks help us identify pointing strategies employed by our participants (see Figure 6.3 and Figure 6.4). As we mentioned, Bob and Dave's movements were relatively controlled suggesting they were consciously making an effort to have aimed controlled movements. The pointing strategies employed by the two individuals, their performance difference in different directions and the variability showed in both testing rounds, are indicators of the need of a personalized method

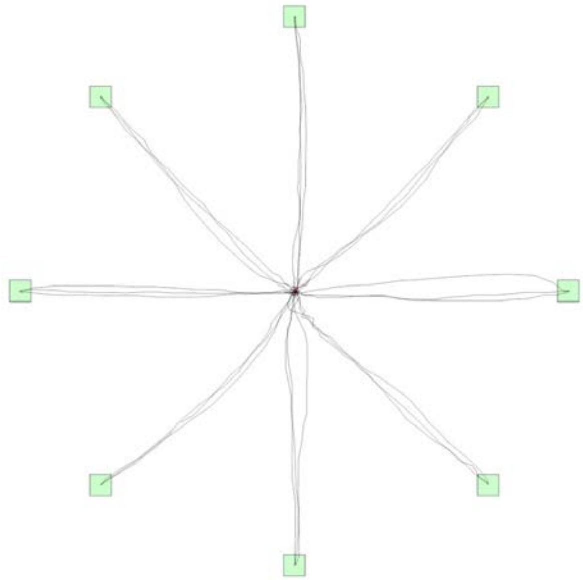


Figure 6.4: Paths taken by Dave in all directions with 16 pixel diameter targets and 512 pixels movement distance.

of assistance.

PointAssist fell short for what was needed by Parkinson's disease patients. The main reason being that it currently detects pointing difficulty using heuristics that are the same for all users. We must take into account the observed performace variability among individuals. A study by Hwang et al. [30] confirms that Parkinson's disease patients, as well as individuals with other severe motor impairments, show high variability in performance. If are going to provide assistance to patients with Parkinson's disease or other motor impairment conditions, we need the assistive technology to adapt to each individual and consider their changing needs and abilities

.

6.3 A First Look at Personalizing *PointAssist* for Individuals With Motor Impairments

The first step we took to provide a personalized method of assistance was to look at the distributions of those sub-movement characteristics that *PointAssist* uses to identify when a sub-movement is difficult. We consider a sub-movement to be near the target if it is less than 30 pixels from the target center. A sub-movement is away from the target if it is more than 60 pixels away from the target center. We consider only sub-movements from tasks that had *PointAssist* disabled. We identify false positives as being tasks where *PointAssist* would have triggered the assistance away from the target. An adjustment of the sub-movement length and sub-movement speed is done by comparing the distributions of the sub-movement lengths and sub-movement speeds near and away from the target center. Figures 6.5 and 6.6 represent the sub-movement distance distributions for Bob near and away from the target respectively. We see that if we selected an upper bound of 20 pixels to identify a sub-movement as difficult near the target we would be accounting for 80% of the sub-movements. At the same time we almost guarantee that most movements away from the target, 2.5% being of length below 20 pixels, will not trigger the assistance provided by *PointAssist*.

Looking at figures 6.7 and 6.8 we selected a maximum speed of 0.12 pxls/ms to account for 80% of sub-movements near target while with this choice only 7% of the sub-movements away from the target would be candidates for false positives. A similar analysis is done for Dave. Looking at figures 6.9 and 6.10 for Dave we se-

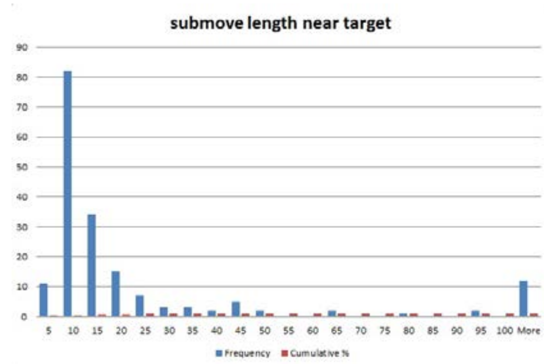


Figure 6.5: Sub-movement length distribution near target (Bob)

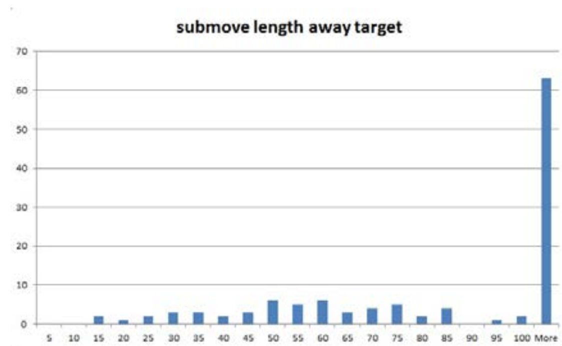


Figure 6.6: Sub-movement length distribution away from target (Bob)

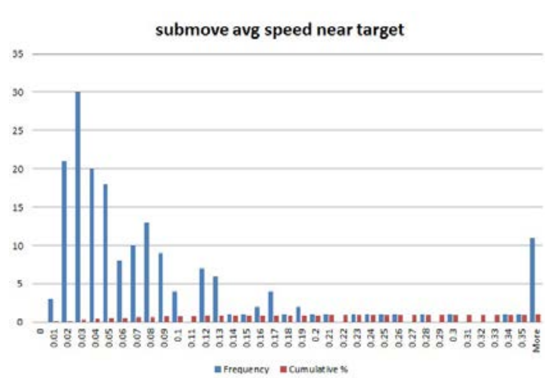


Figure 6.7: Sub-movement average speed distribution near target (Bob)

Figure 6.8: Sub-movement average speed distribution away from target (Bob)

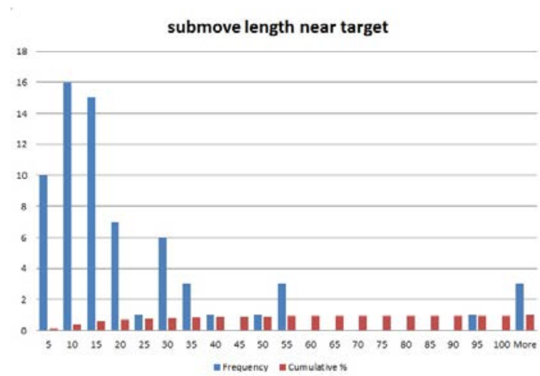


Figure 6.9: Sub-movement length distribution near target (Dave)

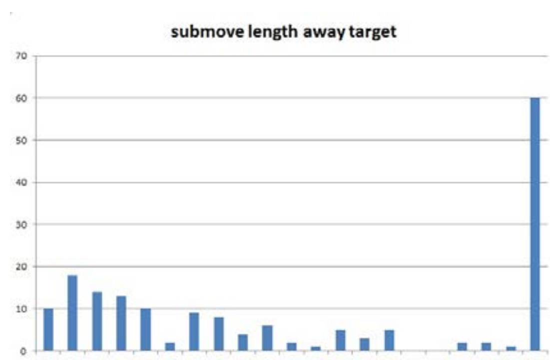


Figure 6.10: Sub-movement length distribution away from target (Dave)

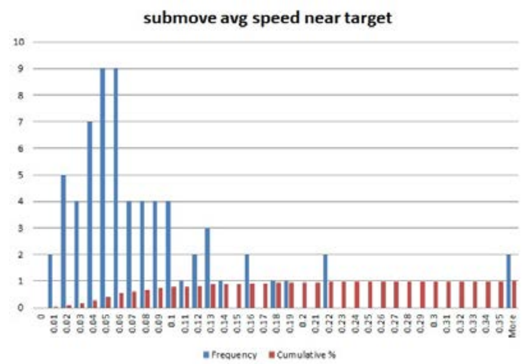


Figure 6.11: Sub-movement average speed distribution near target (Dave)

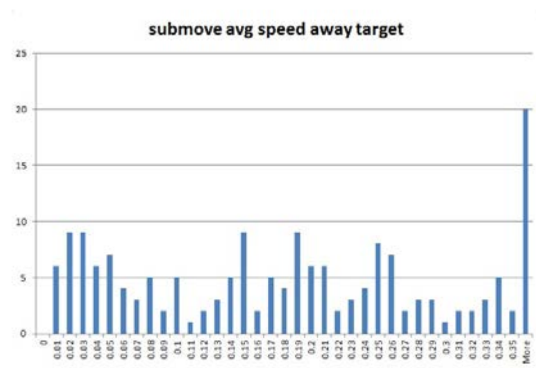


Figure 6.12: Sub-movement average speed distribution away from target (Dave)

lected a length upper bound of 25 pixels to account for 80% of the sub-movements near target. Though the distribution for sub-movements away from target indicates that 30% of sub-movements away from target are 25 pixels or less we would expect the sub-movement speed condition to rule out many of these cases that could yield a false positive result. Similarly, from figures 6.11 and 6.12 we selected a sub-movement speed of 0.14 pxls/ms

In the next chapter we describe and test the manual implementation of the parameter adjustment method described previously. Since we know that the parsing algorithm only triggers the assistance upon detecting a pair of difficult sub-movements, we performed a second analysis of the sub-movements of the participants in the next experiment to improve on the proposed parameter adjustment mechanism. We will describe in more detail the additions to the adjustment procedure and how the heuristic was implemented for each individual that participated in the study. The hypothesis is that this new mechanism of personalization will allow us to extend the help *PointAssist* provides to the population of individuals with motor impairments by yielding statistically significant results on accuracy measures.

CHAPTER 7
ASSISTING INDIVIDUALS WITH FINE MOTOR SKILLS IMPAIRMENTS VIA A SUB-MOVEMENT ANALYSIS

7.1 Research Question

The purpose of this study is to analyze the pointing performance of individuals with a range of motor impairments that affect their fine motor skills. We wish to test the effectiveness of *PointAssist* in detecting target acquisition difficulties amongst individuals with motor impairments and the effectiveness in providing help for these individuals with the mouse speed reduction mechanism that *PointAssist* implements. The previous case study with two individuals with Parkinson's disease informed us of the necessity of personalizing the helping strategy provided by *PointAssist*. This necessity arises from our expectation of a high variability in performance from all individuals due to their differences in motor control as a result of their fine motor skills impairments. In other words, it is hard to make generalizations on such a diverse population of individuals with a wide spectrum of conditions and disabilities that affect their pointing performance. A similar study with Parkinson's disease patients concluded that "Important differences in behaviour with respect to established models of movement indicate that new models are required when considering users with physical impairments." [34]. The variability in performance of each participant will prompt us to analyze the characteristics of the movements and sub-movements which will eventually lead to a heuristic to personalize the assistance provided by *PointAssist* for each individual. We apply the personalization heuristic to the first round of data collection that served as a baseline. We then test the effectiveness of the personalized help provided via *PointAssist* through a second round of data collection which yielded statistically significant differences for many of the dependent variables studied.

7.2 Data Collection

The test relied on exactly the same sub-movements parsing algorithm as in the case study with Parkinson's disease patients (see Chapter 4). We collected information remotely by deploying the software to individuals who were geographically in different places, so that they could install it in their personal computers. One of the major challenges faced was finding participants for the study. We partially addressed this issue by adopting an experiment that remotely tests the participants and that sends us the results automatically via the Internet. The main purpose of this approach was to partially test validity of the software in real-wold interactions by considering the actual setup that each individual had on their personal computers. We had no control over the overall settings on a hardware level. Even though the hardware setups were different, the test was generalized for all of the individuals studied. In the next section we will describe the population demographic data and the information collected regarding each individual's setup. All testing tools, as well as all analysis tools, were developed using C#.

Figure 7.1 Shows the basic demographic information collected for all participants. Once the participants completed the questionnaire the test would begin with a

Figure 7.1: Questionnaire that all participants completed at the beginning of each test (developed in C# with Visual Studio 2010).

series of 5 practice trials followed by the 64 trials that comprise the actual test. Participants had to move the mouse cursor from the center of a green square randomly placed on the screen and attempt to click in the red dot (circle). The red dot was selected from two different sizes and was randomly placed on the screen to represent eight different directions. Figure 7.2 shows different sample trials that the participants completed in two different directions with different target sizes. Participants received instant feedback during the test via sound, once a task was completed, and via the progress bar on the left shown on figure 7.2. Additional feedback was provided for both successful and unsuccessful clicks. A successful click would be recorded next to the progress bar as a greed dot and a missed click as a red dot.

Figure 7.2: Sample trials in the north-west direction with a 16 pixel target (left) and in the north-east direction with an 8 pixel target (right)

7.3 Independent and Dependent Variables

We had three independent variables: direction, target size and *PointAssist* on or off. A task direction was selected out of eight directions from north, south, east, west, north-east, north-west, south-east and south-west. We selected two target sizes, an 8 pixel and a 16 pixel target. Our control variable was the task length which was set at 512 pixels for all task.

The experiment consisted of two data collection rounds. For the first round of data collection, all participants performed a total of 69 tasks (8 directions x 2 target sizes x 4 blocks + 5 practice tasks) with *PointAssist* turned off. This data served as baseline to be analyzed using the personalization heuristic which we will describe later. This heuristic helped us adjust the parameters of the program that identify movement difficulties near a target for each participant.

A second round of testing took place where each participant performed a total of 69 tasks and where the *PointAssist* on/off variable was randomly assigned to all tasks. Half of the tasks had *PointAssist* enabled and half of the tasks had it disabled. The randomization procedure was to generate all tasks with the three independent variables in question equally distributed amongst all tasks. Then we would randomly select the order in which the tasks were presented to each participant, effectively randomizing all three variables from the participants' perspective. The data obtained was analyzed using PASW Statistics 18.0. We used Friedman's test for accuracy measures and repeated measures ANOVAs for the normally distributed data. The main dependent variables studied were click accuracy, press accuracy, release accuracy, movement time, target re-entry, and number of sub-movements.

7.4 Participant's demographics

The population of participants needed for the study is a very limited one. Though we did not require participants to be avid computer users, they should at least be knowledgeable enough to download and install the testing software. If not, they needed the assistance of a third party to help them setup their test. This was the case with 4 of the 16 participants recruited for this study. All other participants were able to follow the required steps in the testing process. We could argue then that the majority of the participants had computer usage knowledge beyond the novice level. This is also supported by the reported average number of hours of computer usage per week by all participants of 13.56 hours/week. An internet connection was necessary so that the results could be automatically sent at the end of each testing round.

Due to the difficulty finding participants, 14 participants were recruited during two recruiting rounds in a span of a little over a year. We reused the data from the two initial Parkinson's disease patients. Their data used the same independent variables as for participants recruited for this experiment and they underwent the personalization process that motivated this experiment. All 16 of the participants were in different geographical locations. Some participants were from Iowa, some from the Chicago area, some from Minnesota and the remaining were from Puerto Rico. We recruited 1 more Parkinson's disease (PD) patient for a total of three cases of PD; 3 Cerebral Palsy patients; 3 Carpal Tunnel Syndrome patients (of which one also suffered from Arthritis); 1 Stroke patient; 1 individual with Developmental Deficiencies; 1 participant with damage to the central nervous system; 1 case of Multiple Sclerosis; 1 individual with Spina Bifida; 2 other individuals were recruited but decided not to disclose their physical disabilities. Informed consents were obtained and most participants reported running the software on Windows 7 machines. Though the hardware specifics are not known to us we collected information on the input devices used and found that 14 out of 16 of the participants (88%) reported using a mouse, and the remaining two used a trackpad. Half of the participants are male, and the average age of the participants was 54 with the youngest being a 25 year old female and the oldest a 74 year old female. For a more detailed description of the participants recruited during this first phase please refer to Table 7.1.

Few of the participants were specific about their particular hardware setups and how their disabilities affected their computer usage performance. We describe

Table 7.1: Participant's demographic data.

ID	gender	age	hours/week	hand	impairment	device
1	male	55	20	right	Parkinson's Disease	mouse
4	female	37	40	right	Cerebral Palsy	mouse
7	male	29	30	right	Spina Bifida	mouse
8	male	48	4	right	physical disability	mouse
9	male	43	45	right	Cerebral Palsy	mouse
10	male	64	1	right	Parkinson's Disease	trackpad
11	female	72	1	right	Stroke	mouse
12	male	58	20	right	physical disability	mouse
13	female	53	5	right	Carpal Tunnel, Arthritis	mouse
14	female	70	12	right	Carpal Tunnel Syndrome	mouse
15	male	72	5	right	Parkinson's Disease	mouse
16	female	41	7	left	Carpal Tunnel Syndrome	mouse
18	female	25	10	left	developmental deficiencies	mouse
19	female	74	1	right	central nervous system damage	mouse
20	female	58	15	right	Multiple Sclerosis	trackpad
22	male	57	1	left	Cerebral Palsy	mouse

the ones we could gather to gain insight from the strategies employed and from the effect the participants think their disability has on their computer usage skills.

One participant reported using a modified mouse with a piece of paper that would prevent accidentally clicking the right button which he reported was a result of his tremors. Another participant reported having issues of a delayed response in releasing the mouse button. Having multiple buttons caused a confusion in the hand coordination that resulted in either an accidental right click while the left click was pressed, or a delay in the release of the left click. The participant reported being frustrated because the delay resulting from the confusion often caused slips and misses on the targets on the screen. This was all reported prior to the experiment, thus my suggestion was to simplify the tasks by removing the confusion of the multiple mouse buttons using a single button mouse that I provided to the participant.

Participants 4, 7 and 8 are of particular interest because they are the only three cases of participants that I was able to personally meet and their willingness to continue participating in my research endeavors, allowed me to recruit them for the longitudinal study that followed. I informally interviewed participants 4 and 7 regarding their computer usage strategies and was able to assess more in detail how their respective disabilities affected their computer usage abilities. They were very different from each other. Participant 4 has Cerebral Palsy that affects her movements on her right hand side. She did not require special accommodations but she did complain that the test was too long at first. She was very excited to see her performance change after the first round of testing took place and she saw the results from the second round with the implemented personalization. Participant 7, being bound to a wheelchair, required a setting that was lower than normal so that he could reach the mouse. The major effect his disability had on his performance was his constant tiredness felt on his arms, slight pain if tasks were long and a minor loss of sensibility on his arms which I attribute to the abnormal position with his arms raised to reach the mouse and the keyboard. Participant 8 is deaf and had cognitive disabilities so we were unable to interview him. However, we gained some insight from his undisclosed disability by watching him work with the computer. The number of computer usage hours per week for Participant 8 is very limited since his cognitive disability requires him to have a proctor by his side to guide him through all his computing tasks.

7.5 First Round: Adjusting Parameters Manually to Accommodate to Individual Needs

The first round consisted of data that did not provide any help to the participants. It served to pinpoint the parameters of the program that could later be used to personalize the asssitance for each individual.

Before introducing the steps used to analyze and adjust the parameters of the program that trigger help, we need to define a few concepts. When *PointAssist* detects two consecutive "difficult" sub-movements, it triggers help via a speed reduction mechanism that we refer to as precision-mode. Difficult sub-movements satisfy two properties: a maximum speed, and a maximum distance. In previous studies a top speed of 0.08 pixels and a maximum sub-movement length of 24 pixels were considered to account for a difficult sub-movement. Because we want to identify when an individual with motor impairments is having difficulty near a target, we need to redefine those parameters for that individual. That is the goal of the procedure we describe next.

When we look at sub-movement characteristic distributions near a target we refer to the characteristics of the sub-movements that took place in a radius of 30 pixels from the target center. Similarly, a sub-movement away from the target will be a sub-movement that is more than 60 pixels from the center of the target. Phase I of the personalization heuristic looks at length and speed distributions of all sub-movements near and away from the target. If precision-mode triggers away from the target, we consider it a false positive. By selecting values that would account for a large number of potentially difficult sub-movements near target we would be reducing the number of false positives.

After selecting values for the sub-movement length and speed parameters, we conduct a simulation on the data using the parsing algorithm that determines when precision-mode would be triggered. This is the analysis that takes place on Phase II. The main idea of Phase II is that from the simulation we look at the difficult tasks and determine which tasks triggered help near the target and which tasks did not trigger help. Difficult tasks near the target were identified by either being tasks where the click was inaccurate, tasks with target re-entry instances and tasks with more than 2 sub-movements near the target. By iterating Phase II we can reduce the number of tasks that will not trigger precision mode near the target. Through our two phases of sub-movement parameter analysis we achieve a naive suboptimal method of personalization. From this method we can learn about strategies that may later be automated and optimized to personalize detection and help.

Now that we have defined all the appropriate concepts we can describe the procedure to personalize the assistance in two phases. This procedure is presented as a minimal approach to individualization and we describe it below:

Phase I - For each participant:

I.1 From the baseline data, determine the distributions of the average sub-movement speed and the distribution of sub-movement length near target.

I.2 Select parameter values that account for at least 80% of the cumulative distribution for each of the two variables of sub-movement average speed and sub-movement length near target.

I.3 Determine the distributions of the average sub-movement speed and the distribution of the sub-movement length away from target.

I.4 Check that the values selected in the second step do not exceed 30% of the cumulative distribution for both variables in the distributions away from the target. Else, select new values from the distributions near target.

I.5 Use the values selected in the previous steps as input to the consecutive sub-movement analysis which takes place in phase 2.

Phase II - For each parcticipant:

II.1 Input the selected parameters into the consecutive sub-movements analysis tool. (the sub-movement analysis tool simulates the behavior of *PointAssist* for tasks that do not have the assistance enabled)

II.2 Determine the number of times the assistance would have triggered for difficult tasks near a target.

II.3 If the number of difficult tasks near target that triggered help is larger than the number of difficult tasks near target that did not trigger help, keep the parameters as the potential sub-optimal settings for that participant. (These are the values used in the second round of data collection where assistance will be enabled).

II.4 Else, adjust the length and speed parameters using the sub-movement characteristics of the difficult tasks near target that did not trigger help. From the resulting adjusted parameters, repeat Phase 2.

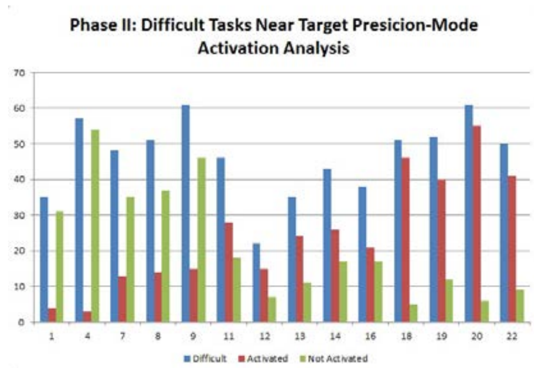


Figure 7.3: First iteration of Phase 2.

In Chapter 6 we described Phase I for participants 10 (Bob) and 15 (Dave). These participants were not subject to the second phase of the analysis because we could not recruit them for further testing. The remaining 14 participants underwent

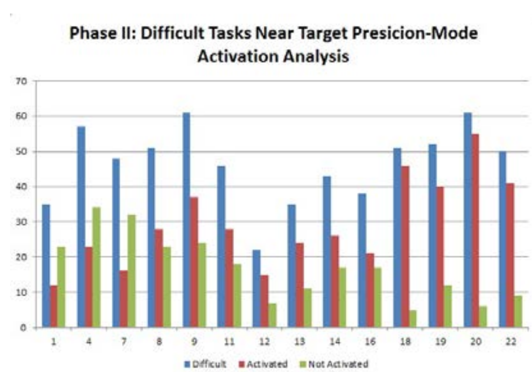


Figure 7.4: Second iteration of Phase 2.

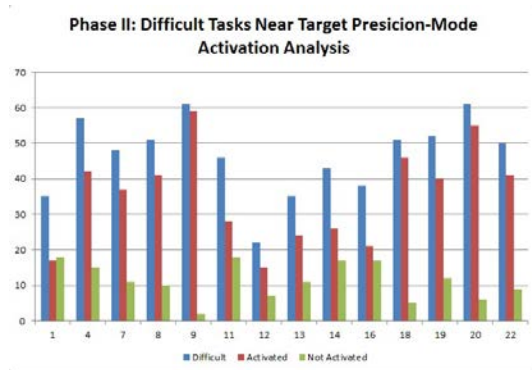


Figure 7.5: Third iteration of Phase 2.

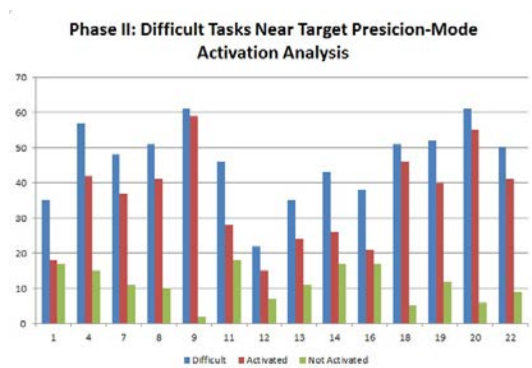


Figure 7.6: Fourth iteration of Phase 2.

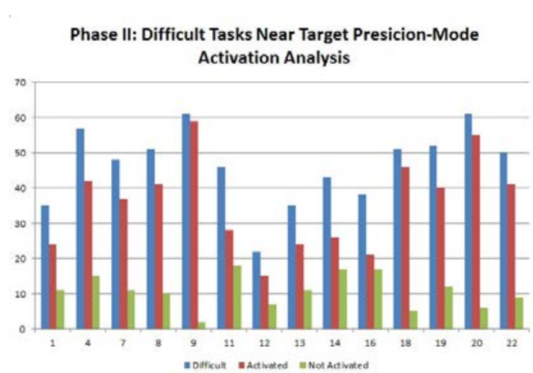


Figure 7.7: Fifth iteration of Phase 2.

both phases of personalization. Please refer to section A in the Appendix for all distance distribution histograms that where used for Phase I. Participant 1 went through five iterations of Phase II; participants 4, 7, 8 and 9 took three iterations, and the rest of the participants only took one iteration before we found values that would maximize the number of difficult tasks near target that would trigger precision-mode. Figures 7.3, 7.4, 7.5, 7.6 and 7.7 show the different iterations of Phase II and the resulting number of difficult tasks that activated and that did not activate precision-mode from the simulation.

The resulting parameters of personalization that will help us identify difficult sub-movements for each participant and that were the result of Phase II of the analysis are summarized in Table 7.2.

Table 7.2: Participant's selected personalization parameters for sub-movement characteristics that determine difficult sub-movements. († Phase I only)

	Participant ID							
	1	4	7	8	9	10†	11	12
length (pixels)	30	35	25	30	45	20	30	25
speed (pixels/ms)	.15	.14	.14	.14	.19	.12	.12	.29

	Participant ID							
	13	14	15†	16	18	19	20	22
length (pixels)	24	45	25	30	40	60	25	25
speed (pixels/ms)	.16	.18	.14	.15	.24	.25	.10	.07

7.6 Second Round: Testing the Adjustment in a Controlled Experimental Setting.

A second round of testing took place with a test that was exactly the same as the one described in section 7.2, except this time randomly enabling *PointAssist* on all tasks. Recall we had participants perform a total of 69 tasks, 5 of which were practice tasks. With 2 target sizes, 8 directions and *PointAssist* on or off, we generated all the tasks half with *PointAssist* enabled, half with *PointAssist* disabled, and we randomly ordered the tasks for the participants to complete. No two participants performed the tasks in the same order resulting in them not being able to figure out when they were being helped or not. Figures 7.8, 7.9, 7.10 and 7.11 show the overall results from all participants in all directions for the cases with *PointAssist* on and *PointAssist* off. In red you can see the paths where precision-mode activated. As desired, the concentration of red is at the center of the images where the target is located. Thus, in general, *PointAssist* behaves as expected. From a qualitative perspective, performance with 8 pixel targets seems to be improved with *PointAssist* enabled.

.

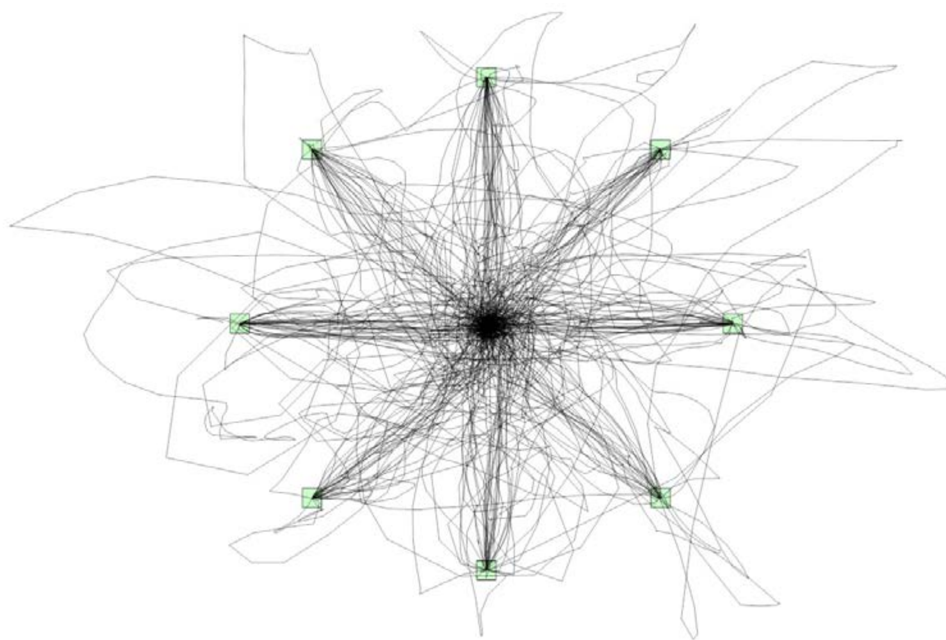


Figure 7.8: All tasks performed by 16 participants on 8 pixel targets with *PointAssist* off.

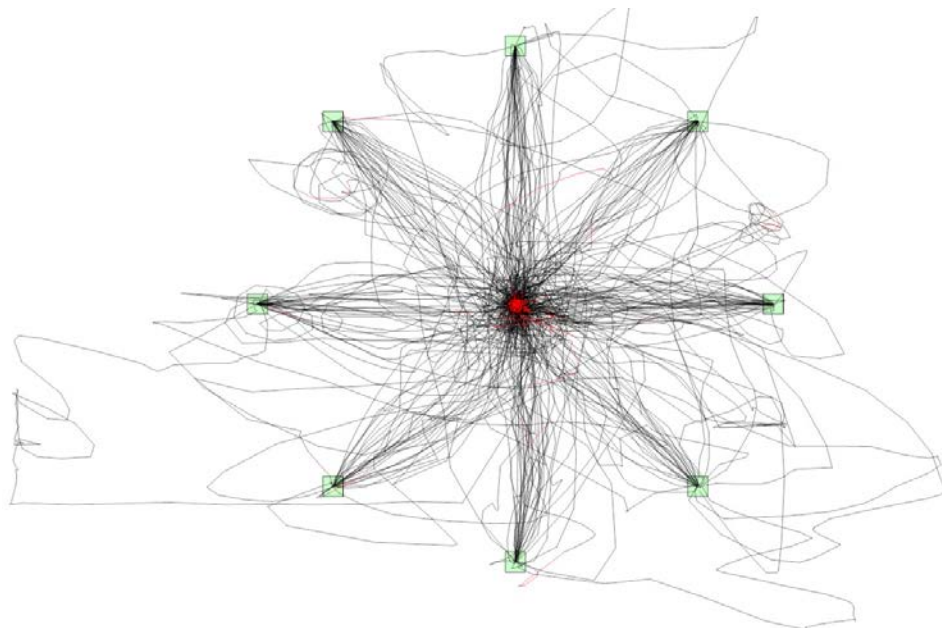
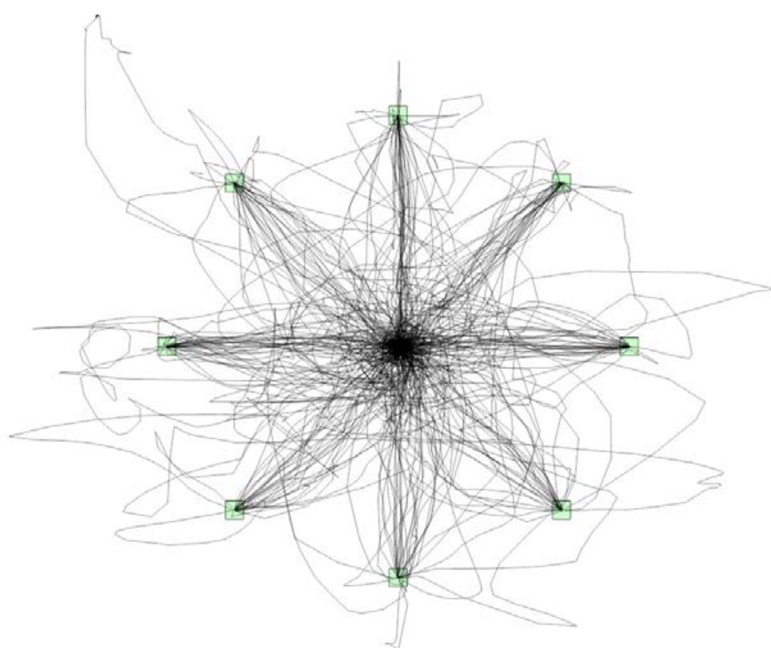


Figure 7.9: All tasks performed by 16 participants on 8 pixel targets with *PointAssist* on. Paths in red indicate precision-mode activated on the path. (Red lines are only visible in color print. In the black and white version, a clustered gray area in the center of the target corresponds to the red lines in the color version.)



F

Figure 7.10: All tasks performed by 16 participants on 16 pixel targets with *PointAssist* off.

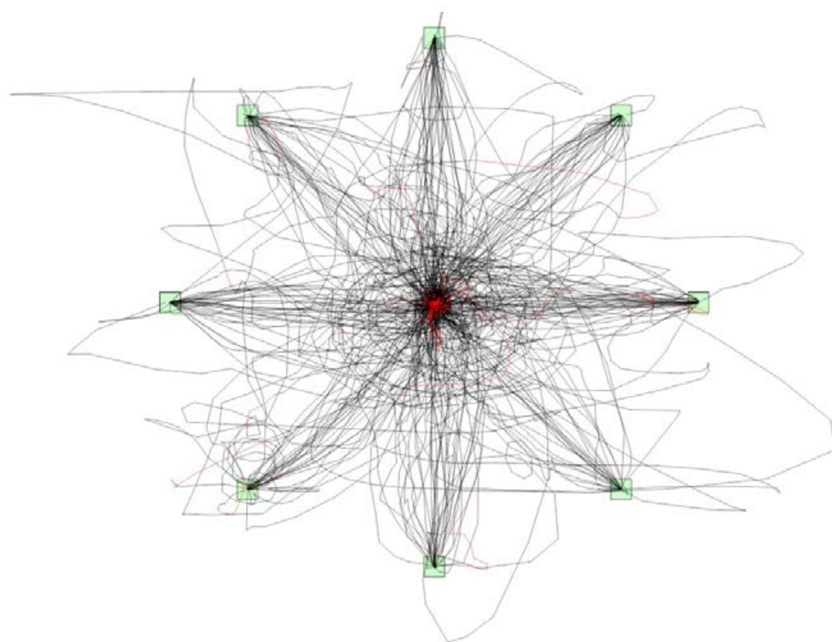


Figure 7.11: All tasks performed by 16 participants on 16 pixel targets with *PointAssist* on. Paths in red indicate precision-mode activated on the path. (Red lines are only visible in color print. In the black and white version, a clustered gray area in the center of the target corresponds to the red lines in the color version.)

PointAssist proved to be statistically significant on click accuracy (χ^2 = 6.4,df=1,p=.011). We see from 7.12 that 9 out of 16 participants improved with *PointAssist*, that is 56% of the participants improved. 6 out of 16 participants did not show any improvement, while one participant did better without assistance. This is a very important indication that the personalization heuristic works. Participant 10 was the only participant that did not improve which we attribute to the fact that he only underwent Phase I of the personalization heuristic. We calculated the effect size using Cohen's d value which we found to be d=.78. This indicates a large effect size of help vs. no help using *PointAssist*. Cohen's d shows how big is the difference between the two means compared to the variability in the sample which we infer is high since participants have a wide range of disabilities and conditions that affect their pointing performance.

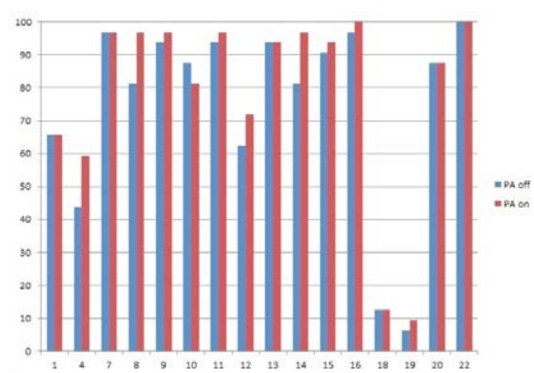


Figure 7.12: Click accuracy results on all tasks performed by all participants.

Figure 7.13 shows the click accuracy distribution for all tasks of the 16 par-

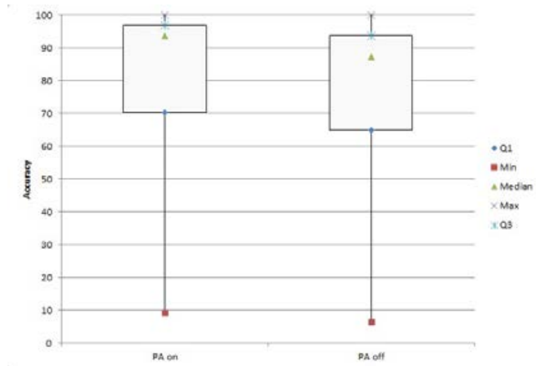


Figure 7.13: Click accuracy results distribution for *PointAssist* on/off for all tasks.

ticipants. The result is further confirmed by a significance effect on press accuracy as well as on release accuracy from Friedman's test with $\chi^2 = 8.0$, df=1, p=.005 and $\chi^2 = 4.45$, df=1, p=.035 respectively. We found a Cohen's d=.70 for press accuracy was and Cohen's d=.71 for release accuracy.

We can see the significance is more pronounced in the press accuracy, and the effect was slightly higher with release accuracy. Since we selected target sizes 50 to 75 percent smaller than a common 32 pixel icon size we can expect users to be within the boundaries of a common target. This means that we are indeed helping individuals be more accurate on the initial response of clicking on a target and we are helping them avoid slipping off the target. This is an important result since other studies have found slipping off target to be a difficulty that most individuals with motor impairments encounter [28]. Figures 7.14 and 7.15 show the distributions of press and release accuracy.

We found eight participants had better press accuracy while nine participants

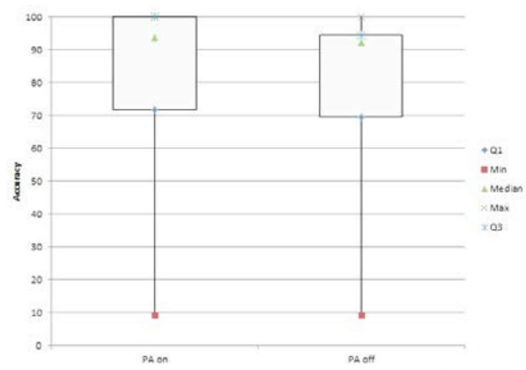


Figure 7.14: Press accuracy results distribution for *PointAssist* on/off for all tasks.

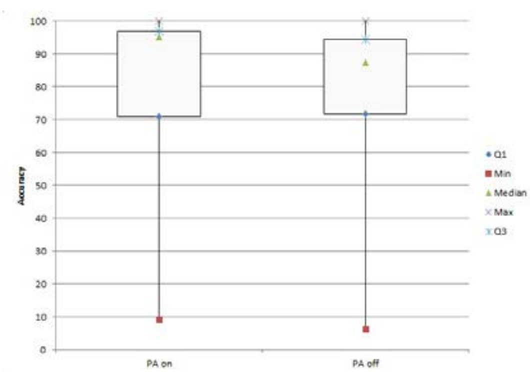


Figure 7.15: Release accuracy results distribution for *PointAssist* on/off for all tasks.

did not see any change on press accuracy (figure 7.16). Nine participants had better release accuracy, 5 had no release accuracy change and two actually had some problems slipping away from the target with *PointAssist* on (figure 7.16). Though in general no significance was found with respect to target sizes, we did find marginally significant results on two movement directions, north and south. Click accuracy moving north or south had the same near significant effect of χ^2 = 3.6, df=1, p=.058. Figures 7.18 and 7.19 show the distributions for the north and south directions re-

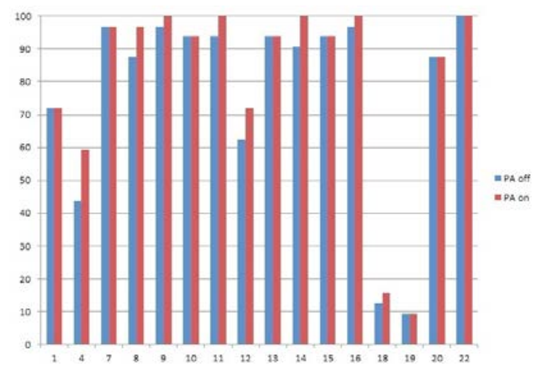


Figure 7.16: Press accuracy results on all tasks performed by 16 participants.

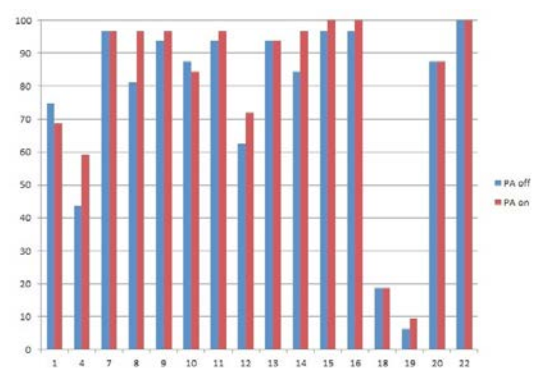


Figure 7.17: Release accuracy results on all tasks performed by 16 participants.

spectively. More specifically, we did find significance for 16 pixel size targets in the north direction with $\chi^2 = 5.0$, df=1, p=.025, and marginal significance in the south direction with 8 pixel targets with $\chi^2 = 3.58$, df=1, p=.059.

Figure 7.20 shows the distribution of all pointing tasks with and without *PointAssist* in the north direction with target size of 16 pixels. Similarly, figure 7.21 shows the distributions for all pointing tasks in the south direction with target size of 8 pixels. No significance was found in terms of movement time, target re-entry

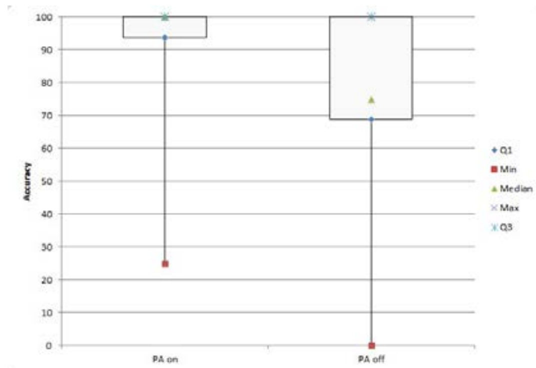


Figure 7.18: Click accuracy results distribution for *PointAssist* on/off for all tasks going in the north direction.

or average click distance to target center.

A within-subjects ANOVA resulted in marginal significance of $F(7, 105) =$ $, p = .091$ on the effect that direction and *PointAssist* had on press duration. Press duration is defined as the time from the initial press of the mouse button to the time of button release. This means that the effect of help on press duration changes depending on the direction that you are going. This marginally significant result indicates a trend which prompts us to look at future research with more participants that would indicate the benefit *PointAssist* may have on certain directions. Indicators of this trend can be seen in some participants that show movement difficulties more pronounced in some directions than others.

Performance comparisons between two Cerebral Palsy participants can be seen on figure 7.22. Notice how participant 9 shows distinct patterns in different directions, seemingly having not much trouble going in the south direction while struggling in

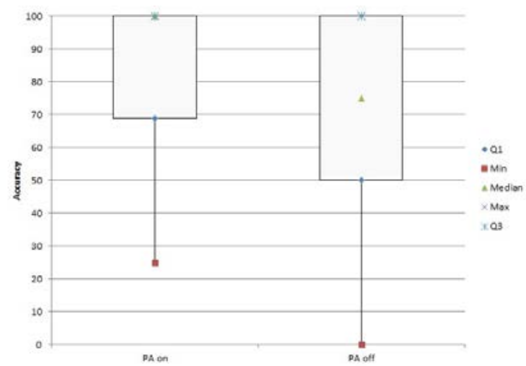


Figure 7.19: Click distribution for *PointAssist* on/off for all tasks going in the south direction.

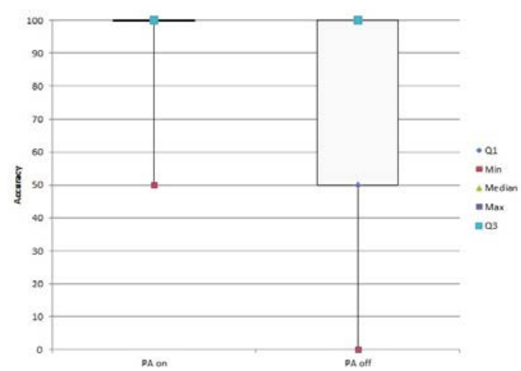


Figure 7.20: Click distribution for *PointAssist* on/off for all tasks going in the north direction for 16 pixel targets.

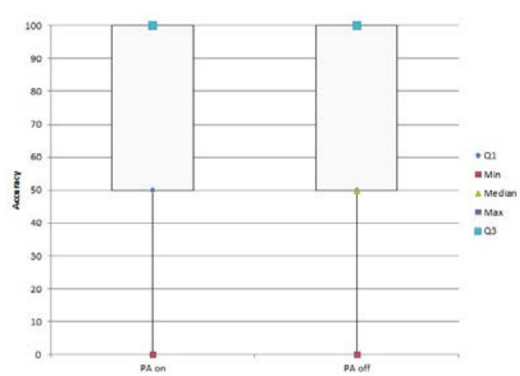


Figure 7.21: Click distribution for *PointAssist* on/off for all tasks going in the south direction for 8 pixel targets.

all other directions. Participant 4 has a more consistent pattern and no particular direction seems to be more affected than any other.

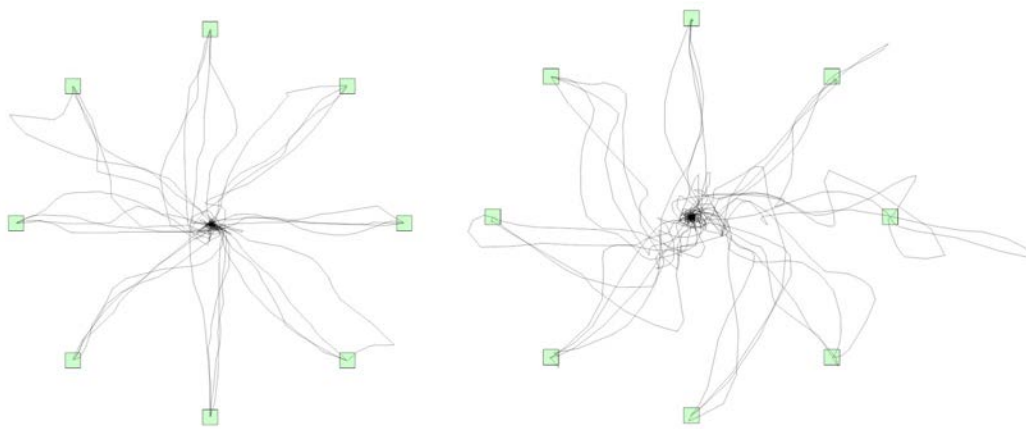


Figure 7.22: Sample tasks for participant 4 on the left, and participant 9 on the right.

Figure 7.23 shows comparisons between two individuals with Carpal Tunnel Syndrome. Interestingly, participant 13 seems to show a pattern of movement slightly skewed to the right, while participant 14 seems slightly skewed to the left. Skewness in the movement patterns may be due to strategies that participants adopt to avoid pain associated with Carpal Tunnel Syndrome. Differences in difficulty can also be seen in different directions. For example, participant 13 had more difficulty initiating movement from starting points in the corners where participant 14 showed more difficulties initiating movement on the east and west bound directions.

Another sample comparison between individuals with distinct impairments is shown on figure 7.24. Participant 11 being a stroke patient and participant 20 having

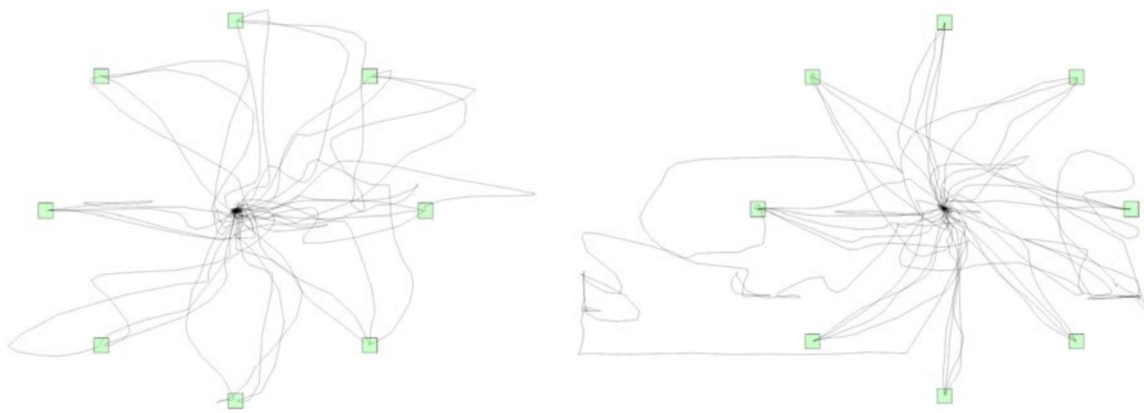


Figure 7.23: Sample tasks for participant 13 on the left, and participant 14 on the right.

Multiple Sclerosis are undoubtedly in distinct categories of performance difficulties that may very well be products of the same motor impairing condition. Ataxia which is found on both individuals with Multiple Sclerosis and individuals that suffered a Stroke may be the cause of the fluctuating and erratic motion we see from both participants in different degrees. However participant 20 has a distinct repetitive pattern behavior which points more towards Myoclonus (involuntary twitching of the muscles) than Ataxia. Participant 11 would be the more likely case of Ataxis since this individual has an extremely erratic yet fluctuating behavior giving rise to the spiral looking picture in figure 7.24.

Thus we see a trend indicating how direction plays an important role in differentiating when an individual is having difficulties.

We found marginally significant differences of $F(1, 15) = 3.44, p = .083$ on the effect that target size and *PointAssist* had on the average number of sub-movements.

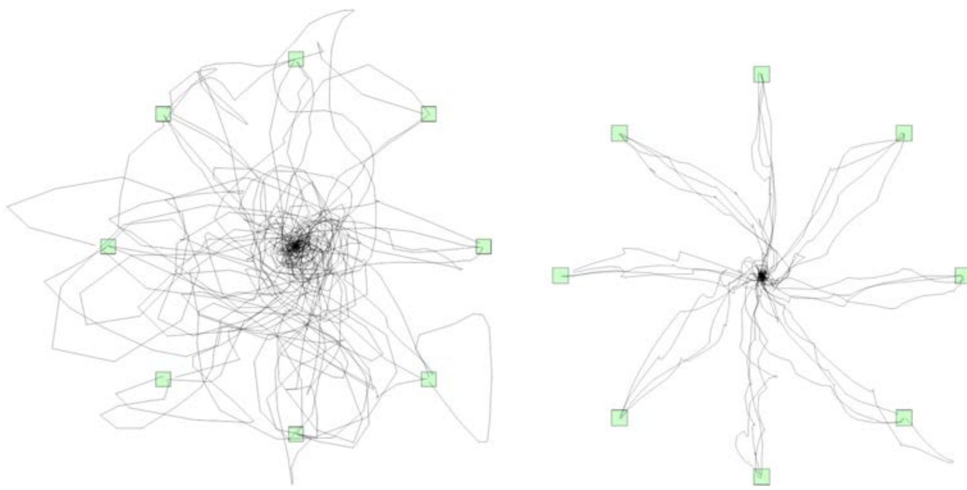


Figure 7.24: Sample tasks for participant 11 on the left, and participant 20 on the right.

This means that the effect of help on the number of sub-movements, changes depending on the target size. This may be indicative of a pattern of different strategic approaches to pointing tasks due to perceptual feedback that the different target sizes provide. It would be logical to think that a participant would feel more confident to reach a larger target thus prompting him or her to perform less sub-movements. More tests need to be done to reach this conclusion but at least we see from the results that some pattern arises which is worth investigating. A stronger indicator of a similar pattern is given when we look at the combined effect of target, direction and *PointAssist* on the average number of sub-movements per task. We found the combined effect to be marginally statistically significant with $F(7, 105) = 2.08, p = .052$.

7.7 Discussion

7.7.1 False Positives

As we discussed before, we consider sub-movements to be away from the target if they happen more than 60 pixels away from the target center. It is possible that *PointAssist* may trigger precision-mode away from the target and we would consider those instances as false positive results. The point of Phase I of the personalization process was to try to improve on the false positives by trying to predict their occurrence making a parameter selection that would effectively identify more sub-movements as difficult near the target than sub-movements away from the target. To show that Phase I worked as intended we should expect a low false positive rate. There were a total of 553 tasks with *PointAssist* enabled and only 43 out of those triggered precision-mode more than 60 pixels away from the target center. This yields a 7.6% false positive rate. This rate is .7% lower than the false positive rate reported for young children [25] and 2.4% lower than the rate reported for older able-body adults [23], showing that Phase I of the personalization heuristic yields positive results and it suggests a future re-evaluation of the parameters used to identify difficulties for young children and older adults that may take on account similar personalization mechanisms.

7.7.2 False Negatives

Phase II of the personalization heuristic was to reduce the number of tasks we identified as being difficult near the target that triggered precision-mode. Thus, we may define a false negative as being a task that was difficult near target but did not trigger precision-mode. Following the analysis done in previous studies with *PointAssist* (see [23, 25]) we looked for tasks where there was target re-entry, participants did not click successfully and precision-mode did not trigger less than 64 pixels from the target center. With only 5 such tasks out of 212 difficult tasks we identified, this yields a false negative rate of 2.4%.

7.7.3 True Positives

With the same criteria for difficult tasks used in previous studies we found that from 212 tasks, *PointAssist* triggered precision-mode in 148 of them within 64 pixels of the target for a 69.8% true positive rate. However, considering that during Phase II we also considered difficult tasks that had more than two sub-movements within 30 pixels of the target center, we also calculated the tasks that triggered precision-mode within 30 pixels. Surprisingly, even though the range was reduced to less than half the distance to trigger precision-mode, we found that 144 tasks triggered precision-mode within 30 pixels of the target for a rate of 67.9% for true positives. Considering how diverse are the 16 participants in terms of their motor impairments and how the results obtained for false negatives and true positives compare to what was found in previous studies of *PointAssist*, we can confidently conclude that Phase II effectively worked and that the personalization heuristic yielded satisfactory results.

From the first seven participants that were initially recruited, three were recruited to participate in further studies. Participants 4, 7 and 9 were compensated for their participation in the longitudinal study that followed and that we will discuss in Chapter 8. By the time we recruited the rest of the participants, there was not enough time to conduct a similar longitudinal experiment which explains why no other participants could be included in the month and a half long longitudinal study that followed.

CHAPTER 8
A CASE STUDY OF THREE INDIVIDUALS USING ***POINTASSIST*** IN A SINGLE-SUBJECT DESIGN LONGITUDINAL EXPERIMENT

8.1 Research Question and Demographics

Our objective with this experiment is to test the validity of *PointAssist* in real-world interactions and the effect that it would have long term. We conducted a single-subject longitudinal experiment with three of the participants from the previous experiment. We recruited six participants in a personal visit we payed to their place of work. All individuals worked at Company A in the United States. Company A is a company that employs only individuals with motor impairments. The company accepts donations of electronic equipment and their employees refurbish and resell these items on Ebay. Interacting with the computer is an essential part of everyday tasks for the employees at Company A, which made them perfect candidates to conduct a longitudinal experiment to test if the assistance provided by *PointAssist* would have a significant effect on their performance over time.

For the longitudinal experiment we were granted IRB permission to compensate the participants in an effort to retain them throughout the length of the study. We paid $25 to each participant. Out of six individuals that we recruited at Company A, three participated in the previous experiment and in the longitudinal study. The other three individuals from Company A, being that their fine motor skills were so severely impaired, were recruited for a separate experiment that we will discuss in Chapter 9.

Table 8.1: Participant's demographic data (longitudinal study).

ID	gender	age	hours/week	hand	impairment	device
Alice	female	37	40	right	Cerebral Palsy	mouse
Karl	male	29	30	right	Spina Bifida	mouse
Joe	male	48	4	right	physical disability	mouse

Table 8.1 summarizes the participants' demographic data. Participant 4 is a right handed female with Cerebral Palsy who has more difficulties with her right hand side. We will call her Alice. Alice did not display any cognitive disabilities and was very eager to participate since she felt she needed help improving her skills using the mouse. Participant 7, whom we call Karl, has Spina Bifida which forces him to use a wheelchair. He has posture issues which affect his stamina when performing skilled tasks. Karl suffered from occasional pain and fatigue on his arms. No cognitive disabilities were reported either. Participant 8 from our previous experiment was our third participant and we will refer to him as Joe. Though his physical impairment was undisclosed, his mobility is restricted by clutches. He is deaf and has some cognitive disability that required a proctor to be by his side while performing all computer tasks so that he would not lose focus. Joe's hand coordination was affected and he also experienced fatigue. All three participants use the mouse as their input device of preference. Their supervisor explained that because of Joe's cognitive disabilities he was rarely assigned tasks that involved computer use. This explains in table 8.1 we see that Joe uses the mouse a lot less than the other two participants.

Some of the accuracy measures from the previous experiment are summarized in table 8.2. We expect that since the click accuracy of both Alice and Joe went up by more than 10%, *PointAssist* should show a positive long term effect for them. In Karl's case, he did not improve on his accuracy, though he was very accurate to begin with. However, he did improve his target re-entry and his completion time with *PointAssist*, so we could expect to see some long term positive effect from the assistance provided.

Table 8.2: Accuracy results for all longitudinal study participants from the previous experiment. Numbers reported are averages of the respective categories for precision-mode on and precision-mode off.

Participant	Target Re-entry	Task Duration	Click Success
Alice (on)	1.34	3461 ms	59.4%
Alice (off)	1.25	3211 ms	43.8%
Karl (on)	1.25	2146 ms	96.9%
Karl (off)	1.34	2279 ms	96.9%
Joe (on)	2.03	6169 ms	96.88%
Joe (off)	1.98	7614 ms	81.25%

8.2 Methodology

Single-subject longitudinal experiments are not common practice in HCI. Yet their application is extremely useful in experimental designs where treatments are introduced randomly over the course of a sampling process. We considered an experimental design where we introduce *PointAssist* as a "treatment" over randomly selected treatment times. Treatment times are defined as blocks of time over which a random assignment of treatment occurs [14]. We can use the same definition of treatment times and talk instead of assistance times since we do not intend to treat any conditions but rather to assist with the pointing tasks on the screen. For the rest of the study we will refer to random assistance times as the randomly selected time intervals in which *PointAssist* was introduced.

Single-subject longitudinal experiments have been shown to provide "significance statements about the effect of experimental treatments on a particular individual when he is the only subject" [13]. The null hypothesis in these type of experiments is that the treatments will show no effect difference over the measurement times [44]. Thus, we wish to test whether we will obtain any significant difference that will indicate that *PointAssist* has a positive effect over time on some measurement of skill improvement on the cursor control on the screen.

We defined time intervals to be 15 minute blocks, and we conducted an experiment with a total of 20 blocks for a total testing time of 5 hours. Each week we tested four blocks for a total span of 5 weeks. Assistace was provided via the speed reduction mechanism of *PointAssist* that relies on the analysis of the sub-movements. Blocks labeled with a letter A correspond to no assistance and blocks labeled with a letter B correspond to assistance. Half of the blocks were A blocks and the other half B blocks, and their order was randomly assigned. This random assignment of letters to blocks would then correspond to the random assignment of assistance times to time intervals. We summarize the obtained randomized block sequence for each participant in table 8.3. The result is an ABAB type of assignment that is also common in the literature. With this design we would satisfy our objective of testing the effect *PointAssist* had over time.

Table 8.3: Random assignment of assistance times to time intervals. A means no assistance was provided. B means assistance was provided. Assistance was provided via *PointAssist*

Participant	Assignment
Alice	A B A B A A B B B B B A A A B A A A B B
Karl	B A B B B B A A B A A B A A A B B A A B
Joe	A A A B A B B A B A B B A A B A B B A B

To test the effectiveness of *PointAssist* in real-world interactions we designed software that would collect data from regular computer use using C# in Visual Studio 2010. That is, we did not give the participants a controlled set of actions to perform on the computer screen. Instead they were asked to do their regular work on Ebay while our program collected cursor movement data that would be imported later into a database using Microsoft Access 2010.

Fearing that most of the time would be spent typing rather than clicking we did suggest that all participants spend some time playing a game. We took screenshots at 5 minute intervals to keep track of the type of activities that the participants performed during each 15 minute block and we found that most of the time they either performed tasks on the Ebay web environment (see figure 8.1) or they played games in one of two websites from *http://www.goobix.com/* or *http://www.setgame.com/*, which were the two gaming sites we suggested initially. Common tasks that the participants performed while playing games on these websites can be seen on figure 8.2

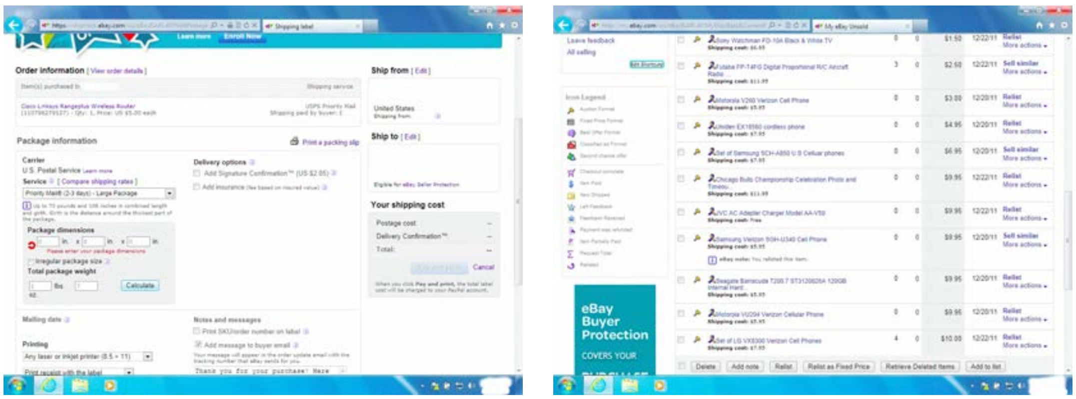


Figure 8.1: Common tasks for Alice, Karl and Joe while working on the Ebay web environment.

Data collection took place with the help of an assistant, who happened to be the participants' supervisor at Company A. Our assistant would submit the output comma-separated value file from each session into a shared dropbox folder (see www.dropbox.com), after each 15 minute testing session ended. The software would collect all the data from the mouse movements as a background process but it did not interfere with any of the actions that the participants performed on the screen. Whenever a participant was in a B block, the software would trigger precision-mode

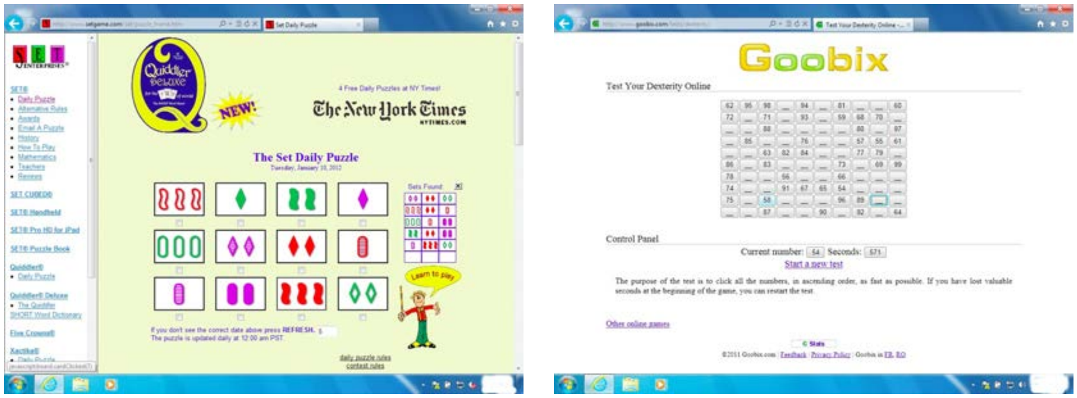


Figure 8.2: Games played by Alice, Karl and Joe on setgame.com (left) and goobix.com (right).

providing assistance when it determined that difficulty was detected. We used exactly the same parsing algorithm as in the experiment discussed in chapter 7 and we also used the same personalized parameters we obtained for these participants in the previous experiment. These parameters were constant throughout all the testing blocks.

We provided the participants with a laptop computer to work with during each block and the computer had the exact same setup for each participant. Participants used a Dell XPS M1330 with Windows 7 Professional installed. The laptop had 15 inch screen, an Intel Core (TM) 2 Duo CPU that ran at a clock speed of 2.00 GHz and 3.00GB of memory available. Participants reported using a generic Dell USB mouse instead of the laptop's trackpad because they found the latter to be too small and difficult to use.

8.3 Measuring performance:
Dependent and Independent variables

To measure performance we needed to think of values that would somewhat resemble the accuracy measures that have been gathered in previous experiments with *PointAssist*. The difficulty here is that our primary concern in previous experiments was accuracy. However we cannot define accuracy in the same terms because we do not know exactly what the targets are nor their location, so talking about click accuracy would make little sense in this context. However as we will see, there are measurements that can give us an idea of how accurate participants were in their tasks.

The only thing we had control over, other than the hardware setup that we explained in the previous section, was the introduction of assistance via *PointAssist*. Thus, our only independent variable is the assistance provided which we randomized over the time intervals as we explained previously. To account for the equivalent of a task in the controlled experimental setting, we considered a task to be the collection of sub-movements between each click, where each click would be a succession of a mouse press and a mouse release. Using this definition of what a task is in our context we studied a number of dependent variables.

In a study about how children and young adults conduct pointing tasks, the characteristics of the sub-movement played a major role in determining how accurate and how controlled their movements were [22]. The study found that children's number of sub-movements was significantly larger than that of young adults and that there were a number of overshoots and undershoots in sub-movements close to the target that would cause them to inaccurately click on a target. More importantly, the cause for these results was the "inaccurate sub-movement lengths and directions" [22] that the participants exhibited. This study made an important point of looking at performance in terms of the sub-movement characteristics.

We looked at the characteristics of the sub-movements as well to see if participants had more or less control over their movements with and without assistance. By looking at the average sub-movement length, the average sub-movement duration and the average sub-movement speed we can determine if a user is able to perform more precise rapid aimed sub-movements. The sub-movement characteristics will help us identify more in detail if *PointAssist* is providing assistance with fine motor skills in the form of better control over the cursor on the screen.

The distance traveled and the total number of clicks in a block are measurements of the amount of activity that each participant had and the efficiency of their paths. If we take the ratio of the total distance between clicks and the total number of clicks we get a description of the performance of each individual per path traveled on average. The lower this ratio is, the better the overall path performance. We will call this the path performance ratio. When we compare blocks with and without assistance we would like blocks with assistance to have a lower path performance ratio.

The sub-movement count per task near a click and away from a click is another variable we studied. Using our definition of a task, we consider a sub-movement as being near a click if it occurs less than 64 pixels away from the click, and away from the click if it occurs more than 64 pixels away from the click. This is in accordance to the way sub-movements near click and away from the click were considered in [25] and in [23]. We looked at the average number of sub-movements for tasks of 128 pixels in length, tasks between 128 and 256 pixels, tasks between 256 and 384 pixels, tasks between 384 and 512 pixels and tasks longer than 512 pixels. This would give us a better idea about how effective is the assistance in a range of tasks from relatively short to relatively long tasks. Again, the choice of short length as 128 and long length as 512 comes from the choices made in previous experiments with *PointAssist*.

We defined the number of slips as any combination of press-release where the distance from press to release is larger than 16 pixels. This could be an indicator that a click was missed. Since each 15 minute session was different participants clicked more in some sessions than others regardless of whether they received assistance or not. So we took the ratio of slips over the total number of clicks per session to account for the variable of slips ratio.

The average distance of mouse press to mouse release in each task could also be an indicator of accuracy, so we considered this variable as well. If the distance from press to release is too large, we can also say that the participants are slipping away from the targets. Ideally we would like for the average distance from press to release of B blocks to be smaller than that of A blocks to account for the effect that *PointAssist* may have.

The number of close clicks over the total number of clicks in a 15 minute session which we call the close clicks ratio, is a measurement we took which would account for the possible number of times that an individual attempts to click at a target. If two clicks are "sufficiently close" together we consider this attempts to click at the same target. We define a click as sufficiently close to the previous click if it is within a radius of 16 pixels from the previous click and it occurs less than 2 seconds after the previous click. In [40], the smallest Windows icon size is said to be 16x16 pixels. Thus it makes sense for us to assume 16 pixels as a radius to check for clicks that are attempts at the same object.

Following the procedures we found in [14] and in [13] we can draw a parallel from the random assignment of assistance to times that in an experiment with multiple subjects is equivalent to a random assignment of subjects to treatments. In this case an independent t-test is used for normally distributed variables, and we ranked the rest of the variables and applied Mann-Whitney U test that is commonly used for two-treatment designs. All data was analyzed using PASW Statistics 18.0.

8.4 Results

We begin our analysis by illustrating some example images from the participants' performance that will help us assess the quality of the study, the contrast between sessions with and without assistance and how participants interacted with the computer in their respective sessions.

If we look at a sample A test from one participant at specific time intervals we can get an idea of the things that the participant was working on during that session. In figure 8.3 we see three cumulative stages of a sample 15 minute A block. We can see from these pictures the instances where the participant either clicked or attempted to click on an object on the screen.

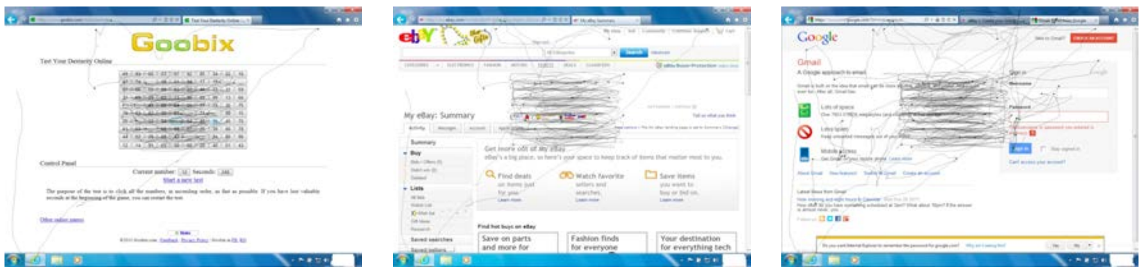


Figure 8.3: From left to right: sample block A at 5, 10 and 15 minute intervals for Alice. Green dots represent a mouse press.

We can obtain a similar set of pictures from a B block as we see in figure 8.4. We highlight in red the paths that triggered assistance for this participant. Participants did not know and could not tell if the assistance was enabled or not. This is because the order of blocks was randomly selected and we did not provide any feedback to the participants if they were being assisted during a session to prevent biasing their strategies or our results. It should be noted that many instances where we see the red paths, indicating that the assistance was triggered, occurred near a press instance. From a qualitative standpoint this is exactly where we want our assistance to trigger.

It is somewhat difficult to assess qualitatively what sort of movement difficul-

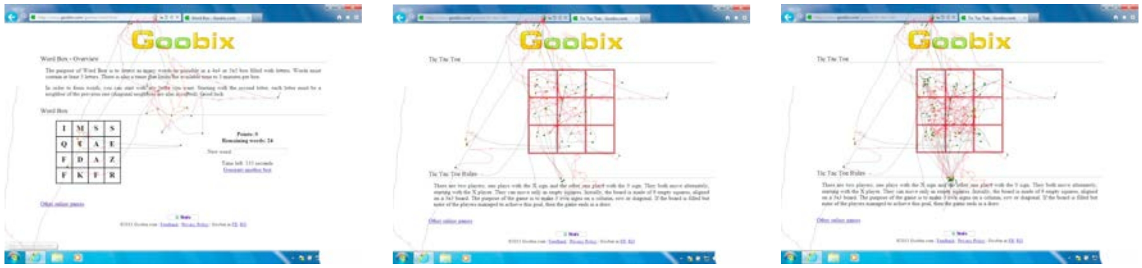


Figure 8.4: From left to right: sample block B at 5, 10 and 15 minute intervals for Alice. Green dots represent a mouse press.

ties a participant may have from the previous pictures. We took each 15 minute block and we normalized all tasks within the block as if all clicks occurred at the center of the screen (recall that we defined tasks as paths between two clicks). This would allow us to see movement difficulties that a participant may have in a way that resembles the kind of diagrams that we have in previous studies using *PointAssist*. With this normalization approach we get a picture like 8.5, where we can more clearly see the accumulation of red paths towards the center for the B block instance showed. This is an indicator that the test worked properly and perhaps a preliminary testament that *PointAssist* works as intended in real-world interactions. We now proceed to state the results obtained from the experiment. Please refer to the appendix B for all normalized diagrams of all three participants.

Figure 8.5: Normalized sample blocks A (left) and B (right), corresponding to blocks in 8.3 and 8.4 respectively for Alice. Paths in red represent instances where precision-mode activated.

8.4.1 Sub-movement characteristics

A summary of the sub-movement characteristics can be found in table 8.4. Alice showed significant differences in all the sub-movement characteristics that we collected in favor of the assistance. The length of her sub-movements were significantly shorter with $t(17) = 6.21$, $p < 0.00001$ with $M = 12.8, SD = .49$) for no assistance and ($M = 11.3, SD = .59$) for assistance (see figure 8.6).

Table 8.4: Means and standard deviations for each participant's sub-movement characteristics

	Alice			Karl			Joe		
Variable	mean	sd	t stat	mean	sd	t stat	mean	sd	t stat
length A	12.8	.49	6.2****	10.9	.54	-2.4*	12.4	.36	.41
length B	11.3	.59		11.4	.44		12.1	.57	
duration A	774	303	2.9*	828	177	.43	1127	632	.64
duration B	480	107		957	470		934	710	
avg. speed A	.053	.006	4.9***	.043	.003	-3.2**	.050	.004	-.24
avg. speed B	.043	.004		.048	.003		.050	.002	
max. speed A	.13	.008	10.6****	.13	.008	.98	.12	.01	2.46*
max. speed B	.09	.006		.13	.005		.10	.01	

$^{*}p < .05$, $^{**}p < .01$, $^{***}p < .001$, $^{****}p < .0001$

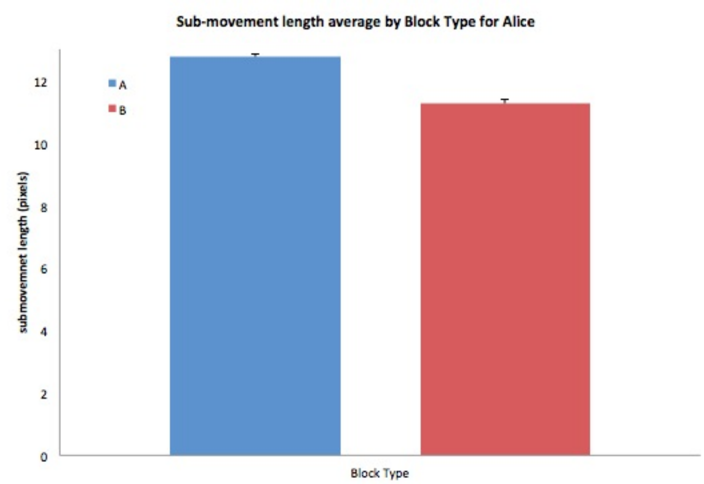


Figure 8.6: Average sub-movement length (mean ± SEM) for tasks performed by Alice. Blocks of type A mean no assistance was provided (left blue bar) and blocks of type B mean assistance was provided (right red bar).

She showed significant differences in the sub-movement duration in favor of the assistance with $t(11) = 2.89$, $p < 0.015$ with $(M = 773.9, SD = 302.6)$ for no assistance and $(M = 479.8, SD = 107.2)$ for assistance (see figure 8.7). The Kolmogorov-Smirnov test of normality indicated that A blocks had $p = .2$ and B blocks have $p = .355$ which helps verify the data is normally distributed.

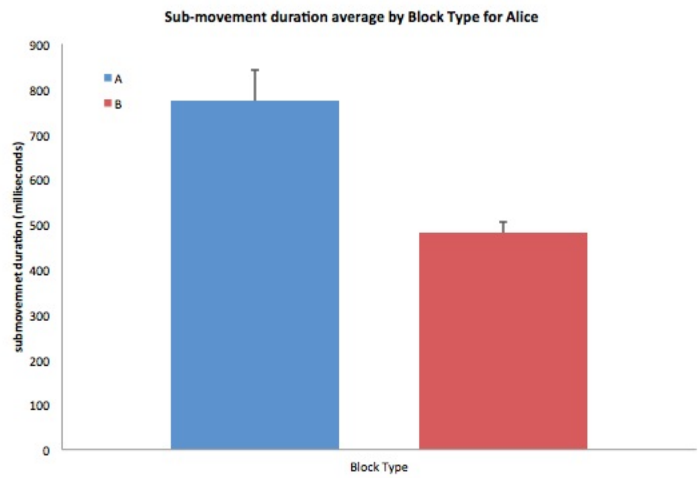


Figure 8.7: Average sub-movement duration (mean ± SEM) for tasks performed by Alice. Blocks of type A mean no assistance was provided (left blue bar) and blocks of type B mean assistance was provided (right red bar).

Finally her sub-movement average speed and sub-movement maximum speeds were significantly lowered with assistance. Results for the average sub-movement speeds were $t(15) = 4.9$, $p < 0.0002$ with $(M = .053, SD = .0055)$ for no assistance and $(M = .043, SD = .0035)$ for assistance (see figure 8.8).

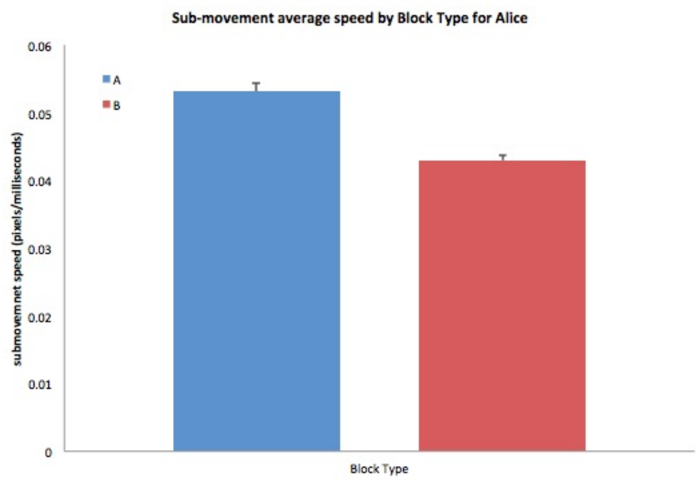


Figure 8.8: Sub-movement average speed (mean ± SEM) for tasks performed by Alice. Blocks of type A mean no assistance was provided (left blue bar) and blocks of type B mean assistance was provided (right red bar).

Her maximum sub-movements speed results were $t(17) = 10.6$, $p < 0.00001$ with $(M = .13, SD = .008)$ for no assistance and $(M = .09, SD = .006)$ for assistance (see figure 8.9).

The tests of normality for the variables of sub-movement length, sub-movement duration, sub-movement average speed and sub-movement maximum speed are summarized in table 8.5. The test of normality remained inconclusive for the length but none of the other variables had results were significant indicating that the data is normally distributed.

The results for Karl and Joe sub-movement were inconclusive but for a more detailed analysis please refer to the appendix C.

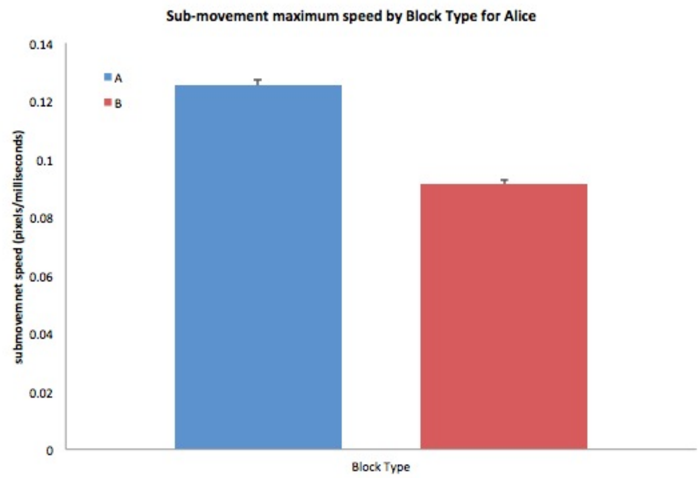


Figure 8.9: Sub-movement maximum speed (mean ± SEM) for tasks performed by Alice. Blocks of type A mean no assistance was provided (left blue bar) and blocks of type B mean assistance was provided (right red bar).

Table 8.5: Kolmogorov-Smirnov tests of normality for variables of sub-movement length, duration, sub-movement average speed and sub-movement maximum speed.

	A blocks			B blocks		
	Statistic	df	Sig.	Statistic	df	Sig.
sub-movement length	.275	10	.031	.127	10	.200
sub-movement duration	.167	10	.200	.214	10	.200
sub-movement avg. speed	.233	10	.132	.196	10	.200
sub-movement max speed	.207	10	.200	.158	10	.200

Joe's sub-movements only saw an effect in favor of the assistance in the sub-movement maximum speed with $t(18) = 2.46$, $p < 0.024$ with $(M = .121, SD = .01)$ for no assistance and $(M = .109, SD = .01)$ for assistance (see figure 8.10).

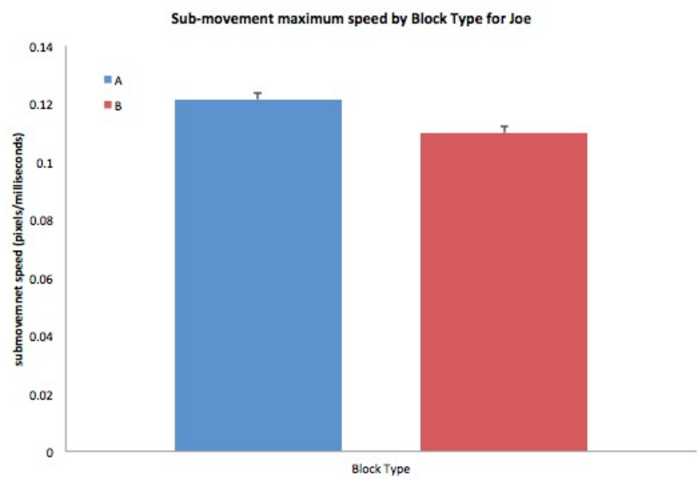


Figure 8.10: Sub-movement maximum speed (mean ± SEM) for tasks performed by Joe. Blocks of type A mean no assistance was provided (left blue bar) and blocks of type B mean assistance was provided (right red bar).

8.4.2 Path performance ratio

Alice performed significantly better with assistance. The average performance over all paths was significantly improved with $t(15) = 2.4$, $p < 0.03$ with $M = 495.7, SD = 45.7)$ for no assistance and $(M = 315, 1, SD = 27.3)$ for assistance (see figure 8.11).

Table 8.6: Means and standard deviations for each participant's path performance ratio by block

	Alice			Karl			Joe		
Variable	mean	sd	t stat	mean	sd	t stat	mean	sd	t stat
ratio A	496	204	2.4*	489	196	-1.1	378	140	.05
ratio B	315	122		639	383		376	75	

$^{*}p < .05$

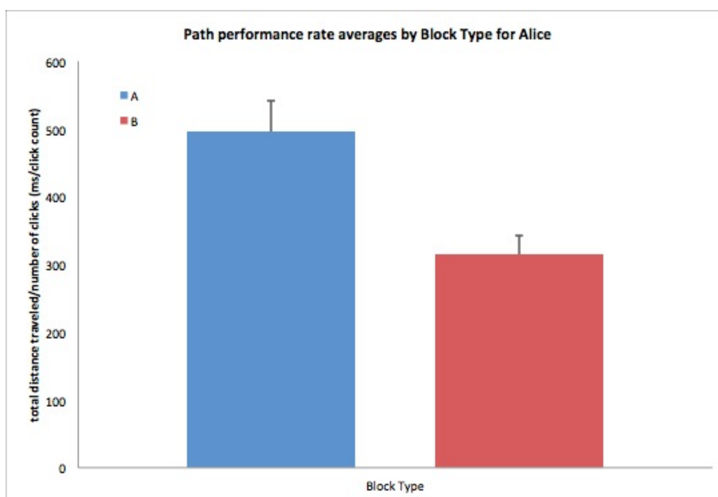


Figure 8.11: Average ratio of total path distance and total number of clicks (mean ± SEM) for blocks performed by Alice. Blocks of type A mean no assistance was provided (left blue bar) and blocks of type B mean assistance was provided (right red bar).

As we see from table 8.6 the test results for Karl and for Joe did not yield any statistically significant results. For the remaining variables of slip rates, average distance from press to release and the close clicks ratio we did not find any conclusive results but a discussion is included in the appendix C as well as the results for the sub-movement count variable.

8.5 Discussion

We saw in section 8.4.1 that the sub-movement characteristics of length and speed were significantly lower for Alice. On average over all tasks, Alice performed 3.94 sub-movements near a target with assistance and 4.18 sub-movements without assistance. In addition, Alice's sub-movements with *PointAssist* took significantly less time.

The sub-movement characteristic results suggest that Alice had better control of her fine motor skills with shorter and slower sub-movements. We argue that the effect of *PointAssist* in her sub-movement characteristics improved her fine motor skills by allowing her to control the cursor better with slower and shorter movements that took less time. As an analogy, think of taking a right turn on a car. You slow down before the curb and you turn little by little. If you are able to perform this move in a shorter amount of time it shows a high level of skill controlling the car on a right turn. So, for Alice a rapid aimed sub-movement is improved if she slows down a bit, and we prove this on the basis that her "rapid" aimed movements took less time. The only missing piece of the puzzle is that we have to remember that Alice is an individual with Cerebral Palsy and what is "rapid" for Alice may not be as rapid as one might think. To finalize the argument, we must recall that Alice was one of many individuals we were able to help with accuracy in chapter 7. Via the analysis of the sub-movement characteristics we found what it takes for her to be accurate using *PointAssist* as the assistive technology.

We also showed how her path performance measured as the total distance traveled over the total number of clicks was significantly better with assistance. Thus we have evidence to conclude that *PointAssist* helped Alice's overall performance.

Karl's results were largely inconclusive. His performance was not consistent and we could not draw any conclusions other than the discussion we have added to the appendix.

We saw that Joe's maximum sub-movement speeds were significantly better with assistance. That is, he was also able to achieve lower speeds with assistance. Overall we think the assistance had the effect of better control when he is about to click but further studies are needed because of his inconsistent performance.

Finally, we can find a parallel from the results for each participant of the single-subject longitudinal experiment with what we found in chapter 7. Recall that in the figure 7.12 we saw an improvement in the accuracy results for Alice (participant 4) and for Joe (participant 8), but not for Karl (participant 7) who was very accurate to begin with. This was also confirmed in the results for press and release accuracy in figures 7.14 and 7.15 respectively. In conclusion, it is not surprising that Alice and Joe received help from *PointAssist* while Karl did not.

CHAPTER 9
THE REVERSE FUNNEL: DEVELOPING FOR INDIVIDUALS WITH SEVERE FINE MOTOR SKILLS IMPAIRMENTS

9.1 Motivation and Demographics

The purpose of this experiment is to introduce and test the Reverse Funnel. The Reverse Funnel is an idea we propose to restrict the movement for individuals with severe fine motor skill impairments in a way that will help them orient themselves in the direction of the desired target on the computer screen.

We began this experiment by recruiting three individuals whose demographic data is summarized in table 9.1. Participant 3, whom we call Fred for anonymity purposes, is a 26 year old individual with Cerebral Palsy that reported using a trackball about 12 hours per week. Fred has a severe mobility impairment that allows him only to move his index finger on his left arm. Because of this the use of the mouse is out of the question. Instead his initial choice of input device is the trackball taped to a table to prevent it from moving that we see in figure 9.1. His limited movement also means that each pointing task is very demanding and as a result he gets tired quickly. He has particular difficulty moving in the upward direction.

We will call participant 5 Ted. Ted is a 26 year old male with Cerebral Palsy that reports using the arrow keys on a special keyboard with large keys to guide the cursor on the screen. Ted reported that his main issue is not being able to control the cursor enough to stay away from the screen boundaries and as we will see his accuracy is fundamentally null.

Participant 6 is another 26 year old with an undisclosed disability that affects his posture and his mobility with both arms. We will call him Ned. Ned reported using a regular mouse only 5 hours a week. His use of the computer is limited and he reported having very poor control and easily getting tired of repetitive tasks because of his abnormal posture.

Figure 9.1: Device used initially by one of the participants with Cerebral Palsy.

We performed initial tests using the data collection method used in chapter 7. The results revealed that the participants were candidates that would not benefit from the help provided by *PointAssist* because their issues go beyond what precision-mode can do for them. Accuracy results are summarized in table 9.2. Notice the high average task completion times for each participant. The sixteen participants that we studied in chapter 7 had an average completion time of 4603.02 ms, an average target re-entry rate of 1.63 and a 74% click success rate for tasks without assistance.

Table 9.1: Participant's initial demographic data pre-Reverse Funnel test.

ID	gender	age	hours/week	hand	impairment	device
3	male	26	12	right	Cerebral Palsy	trackball
5	female	26	30	left	Cerebral Palsy	keyboard arrow keys
6	male	26	5	right	Physical Disability	mouse

Table 9.2: Preliminary accuracy results for all participants. Numbers reported are averages of the respective categories.

Participant	Target Re-entry	Task Duration	Click Success
Fred	0.6	15188 ms	5%
Ted	0.094	12316 ms	0%
Ned	0.42	87523 ms	0%

Compared to these participants Fred, Ted and Ned had 5% or less click accuracy rates, and yet with such a low click success rate their target re-entry rates are non-zero. This translates to very poor control over a target as well as very poor control over the devices at the time of performing a click. In addition, Fred's task duration was 3.3 times worse than other individuals with motor impairments, Ted's task duration was 2.7 times worse and Ned was 19 times worse. That is, it took Ned more than a minute to complete a movement across the screen of 512 pixels when the average individual with motor impairments that we studied only took 4 seconds.

Numbers are one way to show the severity of these participants fine motor skills. But if we look at their actual tasks from a qualitative standpoint, the picture tells an even more compelling story. We can see Fred's overall performance in figure 9.2. Fred's difficulties range from the inability to maintain a steady path to the difficulty in initiating movement in the correct direction.

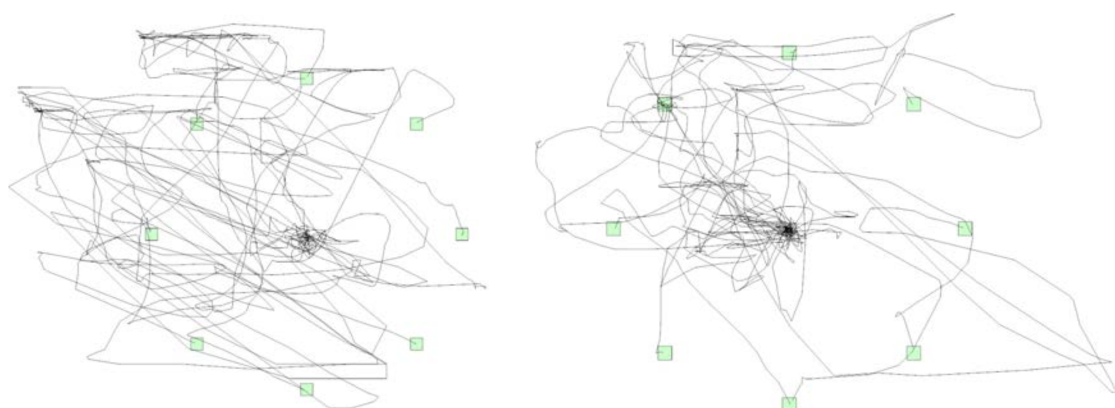


Figure 9.2: All tasks for Fred on 8 pixel (left) and 16 pixel (right) targets.

We can see Ted's overall performance in figure 9.3. Ted also showed difficulties keeping on a steady path, and relatively speaking we can argue that the level of severity in his fine motor skill control is similar to that of Fred's and that makes sense since they are both Cerebral Palsy patients. However, we know a priori that Fred's peculiar situation is that of being able to move only the left index finger which says perhaps that Ted's difficulties are worse than that of Fred's.

Ned's performance is altogether different and far more chaotic patterns arise from his movement behavior. We see in figure 9.4 what is an indiscernible movement

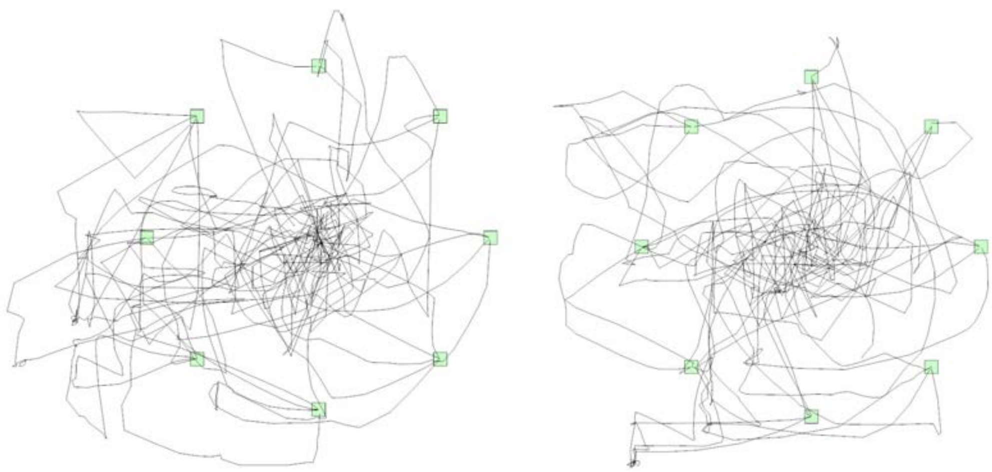


Figure 9.3: All tasks for Ted on 8 pixel (left) and 16 pixel (right) targets.

pattern that tell's us that any approach to help him steady the cursor on the screen needs to have a mechanism that restraints his inability to steady the mouse, it needs to help him slow down his movements and needs to keep him away from the screen edge which seems to be a major hindrance of his performance.

However from amidst the chaos, we rescued some instances from all three participants that will help us motivate the development of a new method of assistance. Figures 9.5, 9.6 and 9.7 represent single tasks from each participant that illustrate a common occurring pattern. The pattern we are referring to is the zig-zagging pattern that in fact is ideally characterized by Ted's east-bound task on the 16 pixel target of figure 9.6. Fred's patterns in figure 9.5 also show a similar behavior and in both instances we see how the direction towards the target is hardly ever maintained. In few instances, Ned showed that his performance suffered from this zig-zag pattern

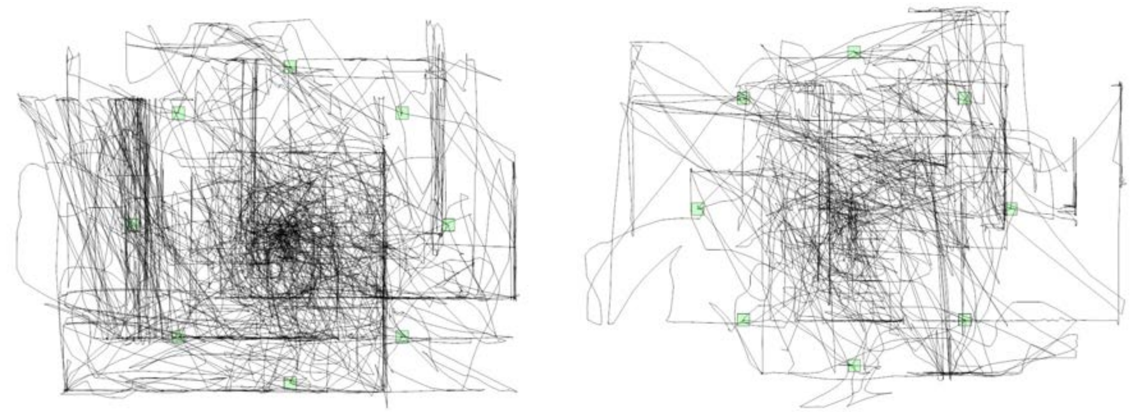


Figure 9.4: All tasks for Ned on 8 pixel (left) and 16 pixel (right) targets.

(see figure 9.7).

Another important observation is that in all the examples we have seen of the three participants, none of the tasks were accurate, that is, all paths end outside of the target area. In fact, in an informal interview with the participants after this round of data collection we were able to gather a series of comments and suggestions that we took into account in the design of the algorithm that we test in this chapter.

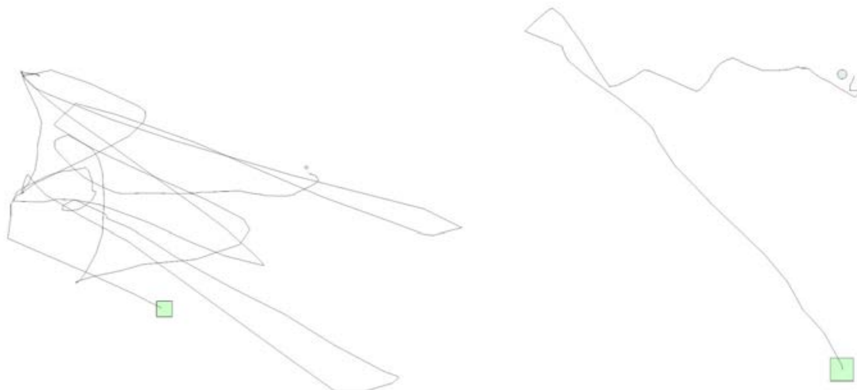


Figure 9.5: Sample tasks for Fred on 8 pixel (left) and 16 pixel (right) targets.

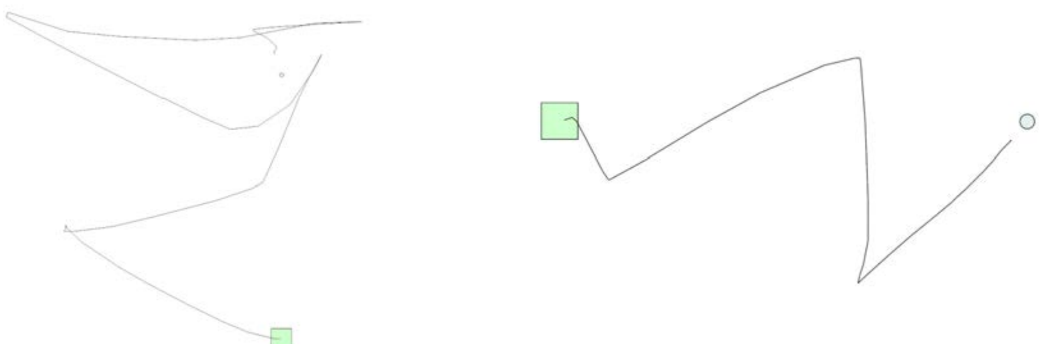


Figure 9.6: All tasks for Ted on 8 pixel (left) and 16 pixel (right) targets.

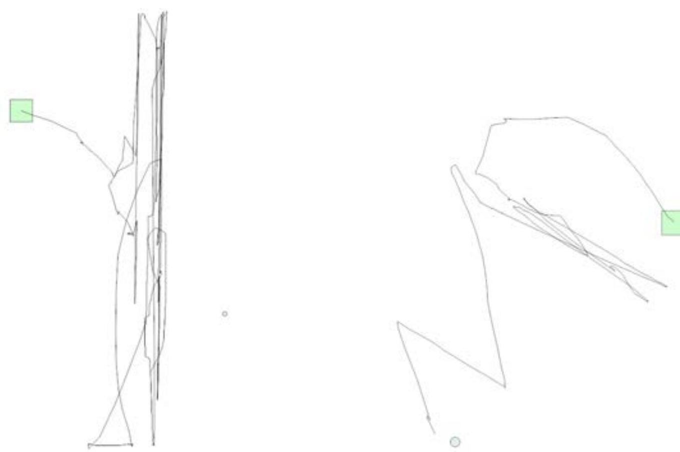


Figure 9.7: All tasks for Ned on 8 pixel (left) and 16 pixel (right) targets.

9.2 Designing an assistive technology based on the participants comments

First and foremost, the participants complained about the length of the test. Having too many repetitive tasks with little or no feedback of what they were supposed to do was a major concern for all three participants. Second, the length of the tasks as well as the the size of the targets were described as being difficult. The length, because it took them too long to reach the target. The sizes of the targets were too small and they could hardly ever click accurately on them. They reported that they sometimes gave up by clicking before reaching the target to finish the task earlier because they were tired. Third, the lack of feedback gave them a hard time. In some instances they required assistance from a third party to help them with positive reinforcement through the test when a task was completed successfully. In others, the third party would help them stay on task and remind them of the tasks' objective. These interactions we experienced first hand while visiting the participants. From this interview we gathered information that was valuable to the design of the experiment by drawing parallels from what the participants did to learn to use other assistive devices such as the wheelchair. We must point out that the main objective of our experiment was not to teach them how to use the mouse nor to teach them how to point on the screen. Yet, the experiences from other contexts where they learned to use a device became extremely important in the design of the Reverse Funnel that we describe later.

The strategy employed by Ted in the process of adapting to the use of the motorized wheelchair was to practice in a closed corridor. The feedback he received from bumping into the walls helped him keep a steadier straighter movement. So, we figured that we should develop a method that would mimic this "wall" if it would help ameliorate the arising zig-zag pattern we saw in some instances of their movement behavior. However, the question still remained about how much we could improve their performance using software alone.

To ease some of the frustration inherent to the use of their current input devices, we took the liberty of suggesting alternative methods of input. We acquired a trackpad that could be connected via Bluetooth to Fred's computer. It made sense that if his mobility was limited to a single finger, a device such as a trackpad would be a great improvement over the trackball he had used up until the time of this experiment. Ted mentioned that his preferred method of input was the mouse and we suggested that in the coming experiment he should employ the use of the mouse rather than the keyboard arrow keys he had used in the initial data collection round. Upon hearing about both Fred's and Ted's alternative input methods, Ned himself suggested he should use a combination of both. He would approach the target with a trackpad and once on a target he would perform the click with a mouse. This, he believed, could greatly improve his ability to stay away from the screen edges and should help him be more accurate when clicking on a target.

As we will see, from the test of the Reverse Funnel that we developed, their performance was greatly improved just on the adoption of the alternative input methods. However, both the numbers and the images will show that indeed, the fundamental idea behind the Reverse Funnel of restricting the movement to a region on the screen, provided the final touch in what we consider to be an incredible increase in performance for all three participants.

9.3 The Reverse Funnel

The idea behind the Reverse Funnel was a conjecture from the initial observation of the zig-zagging pattern we saw in the first data collection round. It later confirmed to be a sensible approach in light of the comments from the participants regarding their learning strategies employed for assistive devices such as the motorized wheelchair. What we did was create a reversed funnel that would open up in the direction in which the user moves (see figure 9.8) based on the sub-movement parsing analysis we have used in previous experiments. Thus the funnel does not funnel in the movements but rather funnels out the movement. Therein the name, Reverse Funnel.

Inside the funnel the cursor moves freely according to the experimental settings which were constant for all participants. Outside the funnel cursor moves very slowly. In fact the cursor speed was slowed to a minimum while outside the funnel. The logic was that rather than restrict the movement completely by creating a barrier that would prevent access to some elements on the screen, slower movement should prevent erratic movement behavior would encourage user to get back inside the funnel area which was in the predicted movement direction.

We managed this prediction by using the cumulative sub-movement direction

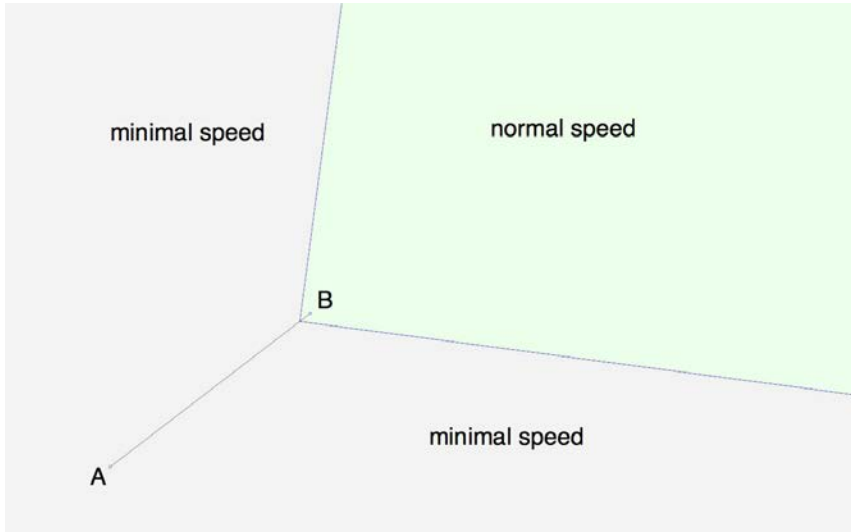


Figure 9.8: Funnel sample opening in the direction of $\overline{AB}$. Area inside of the funnel where the cursor moves freely is colored in green. Area in gray would cause the cursor to move at minimum speed.

average observed from each 5 consecutive sub-movements collected. We predict the direction based on the last point of the oldest sub-movement and the last point of the newest sub-movement (see figure 9.9.

However some unexpected behavior could occur in the case that a user is genuinely trying to move in a direction other than the direction the funnel predicts. This could cause some frustration because of the decreased cursor speed outside the funnel area. We resolve this by creating a timer that would reset the funnel if the user is outside the funnel area for too long. The actual algorithm implemented in C# follows.

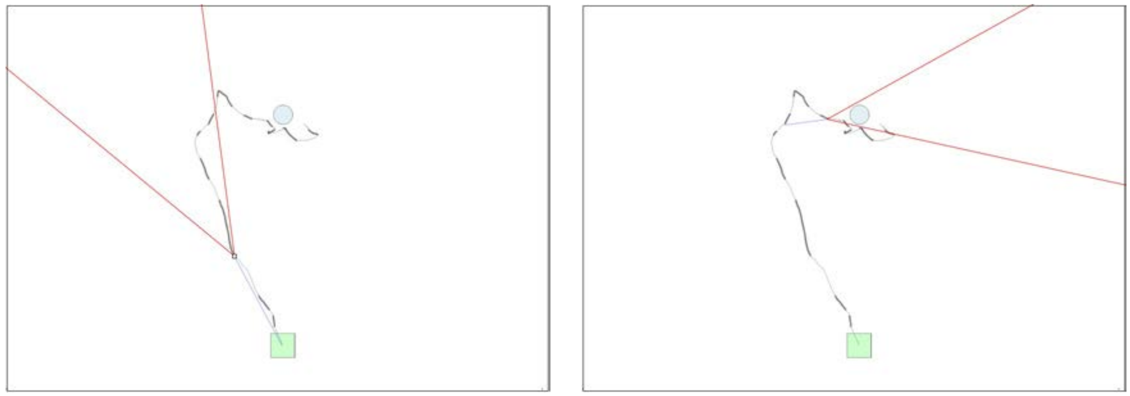


Figure 9.9: Sample task in the north direction showing the Reverse Funnel in different stages. The blue line illustrates the points used to calculate the opening direction of the Reverse Funnel. The red lines are the funnel boundaries and what the participants actually see. The sub-movements are illustrated with alternating grey line thickness.

Reverse Funnel Algorithm:

While(a new sub-movement is identified)

1. Collect sub-movement on a list
2. If the list contains 5 sub-movements:
   - 2.1 Calculate the vector that points from the last point of the oldest sub-movement in the direction of the last point of the newest sub-movement on the list.
   - 2.2 Use the last point of the last sub-movement on the list as the vertex and open a reversed funnel with a 45° opening in the direction of the calculated vector in step 1. Let the sides of the reversed funnel span until the edge of the screen.

3. Alternate the following steps until a new sub-movement is observed:

   3.1 If cursor is inside funnel wait for the next sub-movement and delete oldest sub-movement from the list once the new sub-movement has been recorded on the list. Continue with the next loop iteration.

   3.2 Else if cursor is outside of the funnel reduce cursor speed to a minimum and start a 6 second counter. After 6 seconds have passed, if cursor is still outside of the funnel calculate a new vector from the last point of the oldest sub-movement to the last point observed outside of the funnel and go to step 2.2 using the last point observed as the vertex. If cursor returns inside the funnel before the 6 seconds have passed restore the cursor speed and go to step 3.

In a sub-movement analysis of motion impaired users, 90% of the tasks from able-bodied users required less than seven sub-movements, while motor impaired users required seven sub-movements or more for the same percentage of tasks [30]. The experiment in [30] had tasks lengths of 574 pixels. Our reasoning for selecting 5 sub-movements as the threshold for predicting movement direction stems from the fact that our experiment has tasks of distance at most 44% shorter than the tasks from the experiment previously mentioned. Furthermore we found that participants averaged 1.42 to 1.62 sub-movements before a change in direction was detected, where a change in direction is defined as in Chapter 4 where we described the sub-movement parsing algorithm. So we could expect a clear change in direction in less than two sub-movements which is why we update the Funnel after a new sub-movement is detected. That is the main reason we rely on the spatial average of 5 consecutive sub-movements to predict movement direction for all three participants.

9.4 Experiment setup

We provided the participants with a laptop computer to test the funnel algorithm. The laptop specifications are the same as those described in Chapter 8. The participants were asked complete a questionnaire that can be seen on figure 9.10.

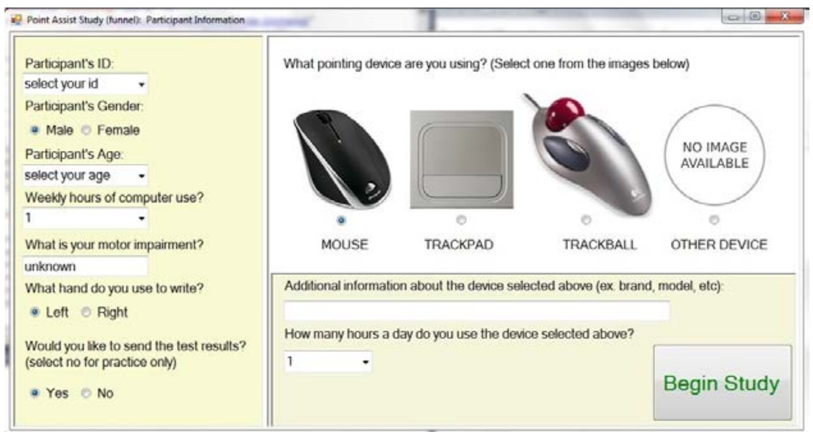


Figure 9.10: Questionnaire that Reverse Funnel participants completed at the beginning of each test (developed in C# with Visual Studio 2010).

Participants ended up using the suggested new input devices and we can see their setups in table 9.3. Fred used an Apple Magic Trackpad that connected to the laptop via Bluetooth. Ted used a standard optical Dell mouse. Ned used a combination of both the trackpad to approach a target and the mouse to click on the target.

Table 9.3: Reverse Funnel participants' data.

Participant	gender	age	hand	impairment	device
Fred	male	26	right	Cerebral Palsy	trackpad
Ted	female	26	left	Cerebral Palsy	mouse
Ned	male	26	right	Physical Disability	tackpad/mouse

To accommodate to the participant's comments of the first round of data collection, we chose the following independent variables: target sizes were enlarged to 32 and 64 pixels in diameter; task lengths where reduced to 128, 256 and 384 pixels; number of directions was reduced to north, south, east and west only. Thus, 4 directions, 3 tasks lengths and 2 target sizes yield 24 tasks per block. We asked the participants to complete 4 blocks plus a block of 2 practice tasks for a grand total of 98 tasks. To test the effectiveness of the Reverse Funnel, half of the tasks had the Reverse Funnel enabled and half had it disabled. We then randomized all 98 tasks, which would account for the randomization of all the independent variables of target size, direction, task length and Reverse Funnel on or off.

Furthermore, we decided to conduct a test where the funnel was always visible (see figure 9.11). We made this decision based on two reasons: first, the Reverse Funnel is meant to assist and guide participants in their paths towards a target on the screen; second, we took into consideration that the participants mentioned it was hard for them to focus on the tasks at hand if the feedback provided was insufficient.

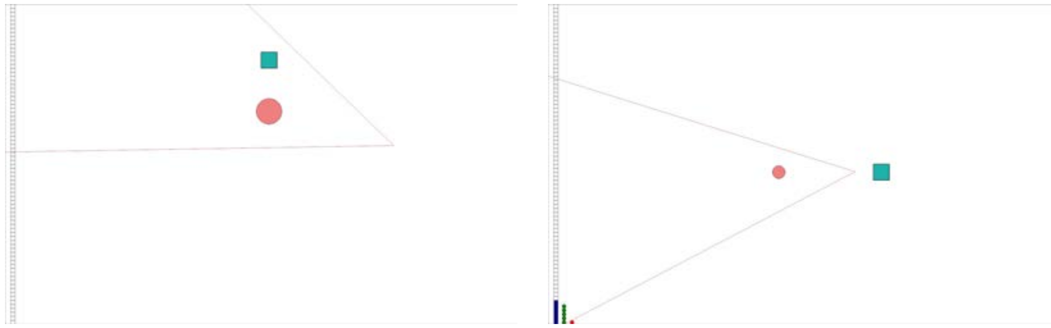


Figure 9.11: Sample Funnel Tests with visible funnel (developed in C# with Visual Studio 2010). Test with a 64 pixel target, 128 pixels distance southbound on the left. Test with a 32 pixel target, 256 distance westbound on the right.

9.5 Results

Table 9.4 summarizes some accuracy measures from the tests performed by the participants with the Reverse Funnel. The participants explained to us some of the strategies they used while performing the tasks with the Reverse Funnel.

Table 9.4: Reverse Funnel accuracy measures for all participants with funnel-on and with funnel-off.

Participant	click success	target re-entry	task duration
Fred (on)	.73	1.33	5601
Fred (off)	.52	1.45	4462
Ted (on)	.52	.85	7616
Ted (off)	.45	1.33	6554
Ned (on)	.35	1.02	17116
Ned (off)	.19	1.14	15833

Fred's main concern was that he felt that the funnel slowed him down too much, though he did not mind the lines on the screen. The numbers confirm this fact since it took him longer to complete a task on average with the Reverse Funnel (see table 9.4). He did not perceive any accuracy improvement however we see from table 9.4 that he did almost 20% better with the Reverse Funnel that without it.

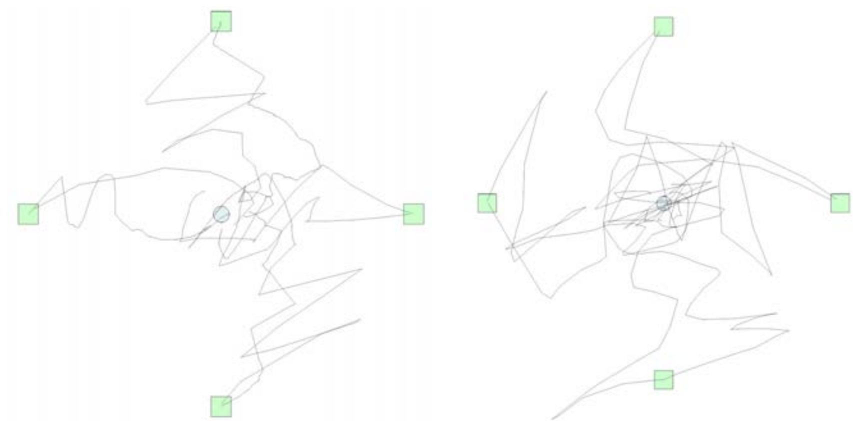


Figure 9.12: All tasks for Fred on 32 pixel targets and 384 distances, with Reverse Funnel (left) and without Reverse Funnel (right).

Ted commented that the lines were distracting. His strategy was affected by the expectation of the Funnel turning on. He enjoyed the colors though and he felt the funnel gave him some encouragement which seemed sort of contradictory, but we interpret that he felt encouraged because he could actually see the help even if it was sometimes distracting.

Ned liked the lines on the screen and he felt they helped him move in the right

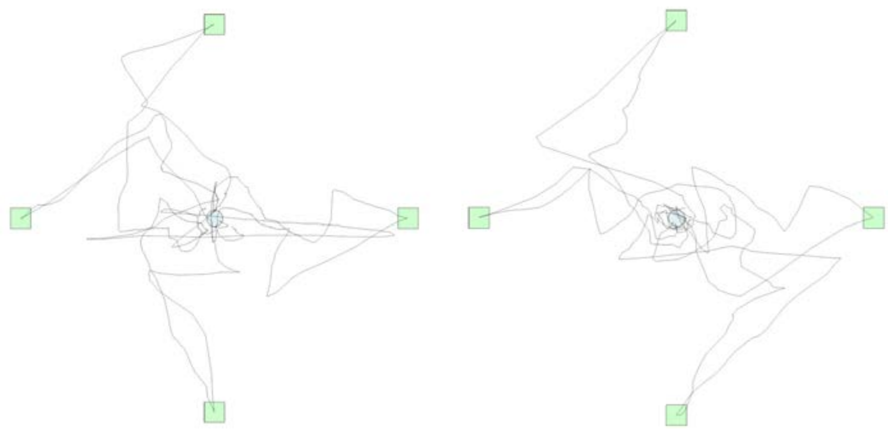


Figure 9.13: All tasks for Ted on 32 pixel targets and 384 distances, with Reverse Funnel (left) and without Reverse Funnel (right).

direction. His subjective appreciation of the funnel was that it did encourage him to do better. Overall, all participants commented that they enjoyed the test, they liked the larger targets and experienced less fatigue with shorter distances. Figures 9.12, 9.13 and 9.14 represent examples all the tasks from all participants with and without the Reverse Funnel with the same target sizes and with the longest task movement distance.

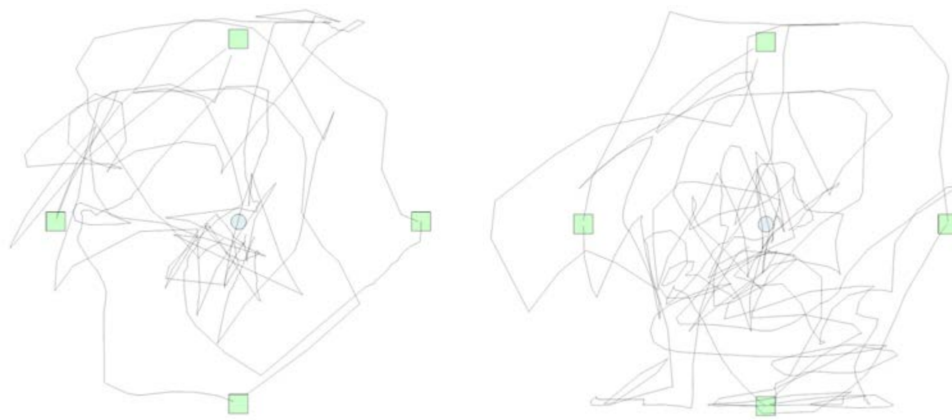


Figure 9.14: All tasks for Ted on 32 pixel targets and 384 distances, with Reverse Funnel (left) and without Revese Funnel (right).

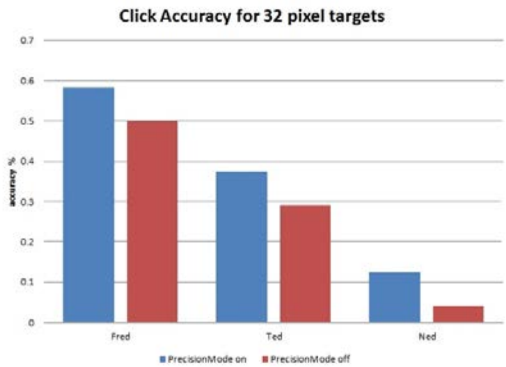
The comments from the participants after they performed the test were taken into consideration in the next list of advantages and disadvantages that the Reverse Funnel may provide.

Main advantages:

- may provide better predictability
- helps keep user on the “right” course since slower movement outside of the funnel encourages returning to the area inside the funnel (We say “right” because we are predicting the intended movement direction.)
- test is not blind, users see the assistance provided and can adjust their strategies based on the visual feedback

Potential disadvantages:

- maybe distracting since the funnel is always showing when enabled.



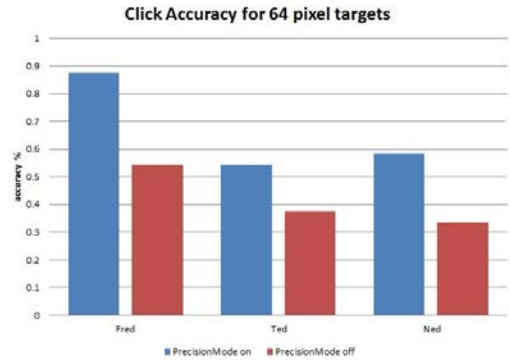


Figure 9.15: Click accuracy by target. 32 pixel targets on the left and 64 pixel targets on the right.

- obstruction of screen objects may occur in real-world interactions
- slowing outside of the funnel can be annoying

An important effect we must discuss is that of the change of input devices and the changes in task lengths and target sizes that clearly affected the performance of all three participants. Yet in almost all instances, though all participants saw an improvement due to the changes in hardware and changes in the test, the Reverse Funnel still had a positive impact on most accuracy measures we collected even by target size. Figure 9.15 shows the accuracy results from 9.4 separated by target sizes. The rest of the results by target are shown in table 9.5.

Table 9.5: Reverse Funnel accuracy measures for all participants with funnel-on and with funnel-off by target size.

	32 pixels		64 pixels	
Participant	target re-entry	task duration	target re-entry	task duration
Fred (on)	1.2	5794	1.5	5407
Fred (off)	1.8	4788	1.2	4136.7
Ted (on)	.7	7775	1	7459
Ted (off)	1.3	6903	1.4	6206
Ned (on)	.5	15465	1.6	18768
Ned (off)	.8	14926	1.5	16740

9.6 Discussion

Recall that Fred changed his trackball for a trackpad. We can see the overall effect of the average task completion time from the trackball's 15188 ms to the trackpad's 5031 ms. His accuracy was greatly improved with the Reverse Funnel. Certainly the choice of input device had a great impact on his accuracy, but he did perform 21% better with the Reverse Funnel regardless of the input device (see table 9.4). He also did better with the Reverse Funnel in target re-entry. Figure 9.12 shows how the Reverse Funnel had a slight effect in some of the paths approaching the target. Our subjective appreciation of the paths is that they look as if the zig-zag behavior was lessened.

Ted's performance saw an improvement as well. His paths were still erratic with a similar zig-zagging behavior, yet his completion times were also reduced to an average 7085 ms per task. The Reverse Funnel did not seem to have an impact on his time completion but it helped with his click accuracy by 7% (see table 9.4). The choice of larger targets as well as the change in input device also had an impact on his accuracy. His target re-entry is probably the best indicator that indeed we provided some help with the Reverse Funnel. We saw that Ted had almost twice as many target re-entries on average without the Reverse Funnel (see table 9.4). Figure 9.13 does not indicate that the Reverse Funnel had any effect from a qualitative standpoint.

Finally, Ned's new strategy of combining a trackpad and a mouse to click payed off with an substantial increase in accuracy. Here we also see that the Reverse Funnel increased Ned's accuracy by 16% (see table 9.4). His target re-entry also saw some improvement though perhaps not very significant and his completion time did not see any improvement at all. In fact, the Reverse Funnel affected his completion time by a full two seconds for each task on average. Figure 9.14 shows how qualitatively the Funnel had some impact in the overall chaotic pattern which seemed to be higher without the Funnel.

The clear indicator of the potential the Reverse Funnel has in performance is the accuracy rate. This is an unexpected result since we hypothesized that the Funnel would have a stronger effect in the path towards, but instead found it made a bigger difference within the target's proximity. We conclude that since the Reverse Funnel relies on a cursor speed reduction mechanism, it benefited the participants' movements while in close proximity to a target, thus affecting the click accuracy the most.

CHAPTER 10
CONCLUSION

At the verge of the most current innovations in the design of assistive technologies, we encounter what has been described as the four pillars of the future developments in personalized dynamic accessibility. These are described on a recent paper published in the bi-monthly publication of the ACM called Interactions[16]. They are the following:

- Adaptation should be shared between the user and the interface: designing an adaptive system should not be done with the premise that only the user is responsible of adapting him or helself to the system.
- Personalized accessibility should take into account each user's necessities and range of abilities: individuals will not have the same needs, will not have the same abilities and so, why should a system be designed without taking each individual into account?
- A system that will adapt to an individual should be dynamic: this will take care of individuals who have a changing or variable range of performance.
- A system should be scalable: a system that provides accessibility should adapt and evolve with changes in resources. Again, individuals will not have the same needs and the system should scale to the requirements of each individual, to the number of individuals each with his or her own unique range of abilities and to other innovations that may come along the way.

Let us take a look at our contributions and how they may reflect the four pillars mentioned before.

- We were able to show via a case study that individuals will indeed have a wide range of abilities and that personalization is, as is stated in the pillars, an essential component of any new assistive technology.
- By extending *PointAssist* to individuals with Motor Impairments we currently have the only proven target-agnostic assistive technology that works with a variety of users, namely children, able-bodied older adults and individuals with disabilities. Through our personalization mechanism we managed to adapt the system to the user, thus sharing the burden of system adaptation with the user. These two reasons make *PointAssist* adaptable to many users and thus comply with the first pillar.
- We tested *PointAssist* and proved we could improve click success rates of individuals that undoubtedly had a wide range of motor skills. Furthermore, through personalization we took each individual's needs and abilities into account. This satisfies the second of the pillars mentioned before.
- Remote testing was effectively implemented for all the experiments we conducted. This is a new trend in the field that points towards feasibility in data collection. Participants with a certain set of abilities are difficult to find and remote testing breaks the geographical boundaries that may prevent effective and feasible data collection.

- We have performed an evaluation of *PointAssist* in a real-world setting. Testing the validity of an assistive technology in a real-world setting has the potential of bridging the gap between development and the adoption of the technology by the users. We have contributed by defining new ways in which we can measure performance in real-world interactions. Real-world experimentation is also very important if we want to take care of scalability issues in our designed assistive technologies.
- With the same study we also tested the value of longitudinal studies in HCI. With the longitudinal study we contributed to the research methods in this type of study. More research needs to be done testing the long term effects of assistive techonologies. This is even more relevant if the assistive technology is going to be adapted to the users' abilities. As we saw from our results, a more careful evaluation of the adaptation procedures is needed since the users' abilities may change over time and the assistive technology has no way of adapting to those changes.
- Through single-subject longitudinal experimentation we provided evidence that *PointAssist* works in real-world environments for some users with Motor Impairments. These results shows promise for future research with individuals with motor impairments that combine longitudinal experimentation with personalization and automatic adaptation.
- Being that longitudinal studies are relatively unexplored experimental alternatives in HCI, little is known regarding how to make assessments on accuracy

and performance, especially if these are conducted as we did in real-world environments. We defined and studied a variety of dependent variables that were able to describe the performance of the individuals in real-wold settings.

- We proposed and tested the Reverse Funnel, a new and novel method of assisting individuals with severe motor impairments. We obtained positive average results that indicate the potential of the assistance as an alternative to be tested with more individuals with motor impairments. This study suggests that software assistive technologies are feasible and promising for individuals with severe motor impairments and it may allow them to effectively use common input devices.

As we look into future research we want to look at an automatic implementation of the proposed sub-optimal procedure for personalizing the assistance provided by *PointAssist*. Individuals with motor impairments that vary over time would greatly benefit from a system that can adapt to their changing and/or variable range in performance. The longitudinal study gave us insight into the long term effects of a system that, though personalized for each individual, was not adapted to the potential changes and variability of individuals with motor impairments over time. We did however furnish results that showed that *PointAssist* works in real-world environments. To scale the assistance for other individuals with motor impairments we suggest that frequent adjustment intervention is needed. Instead of manual adjustments to the assistance, we will further explore implementing an automatic engine that would take care of the periodically and automatically adjusting the assistance as the user's performance varies or as users with different motor impairments are encountered. To increase the validity of the results we would test it in real-world interactions.

Learned statistical models have been used to identify users with physical impairments with 92.7% accuracy [27] as well as novice and skilled users with 91% accuracy [26]. By looking at the characteristics of the sub-movements and implementing similar models as those used to distinguish between able-bodied and impaired users we can identify difficulties and variabilities within those difficulties that would help our personalization mechanism and will make an assistive technology such as *PointAssist* comply with the third pillar of dynamic accessibility.

We also would like to explore more in depth the proposed Reverse Funnel. We need to expand our results to a significant number of individuals with motor impairments. We can also explore modifications to the Reverse Funnel that would help us adapt and optimize the assistance for individuals with varying range of abilities. Recall that the Reverse Funnel had very specific parameters such as a fixed opening of 45°. Also, we only accumulated 5 sub-movements in the spacial average procedure we implemented to predict movement direction. Neither the angle opening nor the number of sub-movements to predict direction were tested to be optimal. Optimal solutions need to be found and alternative solutions need to be explored that make better predictions of the direction where the Reverse Funnel should open.

APPENDIX A
CONTROLLED EXPERIMENT: DISTRIBUTIONS USED DURING PHASE I.

This section contains distributions for sub-movement length and sub-movement average speed from all the participants of the experiment conducted in Chapter 7. Recall that near target means less than 30 pixels from the target center and away from target means more than 60 pixels from the target center. Distributions are organized so that near target are next to away from target for both sub-movement length and sub-movement average speed in order to better visualize the process of choosing parameters for personalization during Phase I.

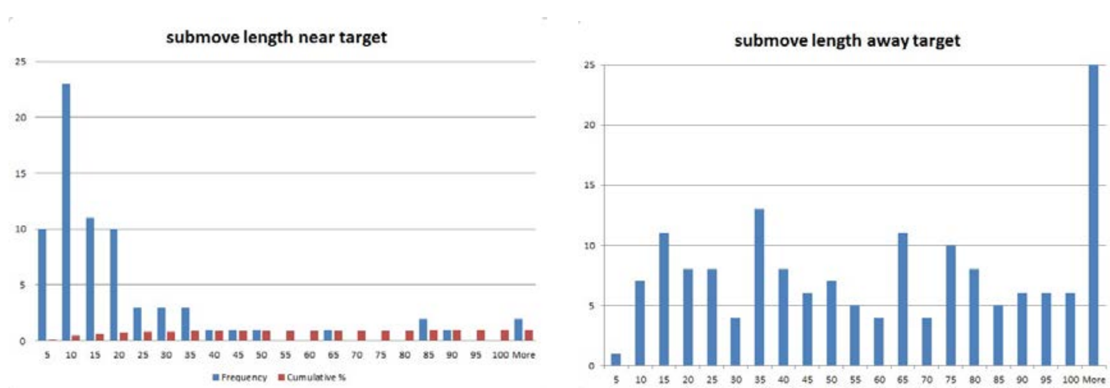


Figure A.1: (Participant 1) Sub-movement length distribution near target (left) and away from target (right).

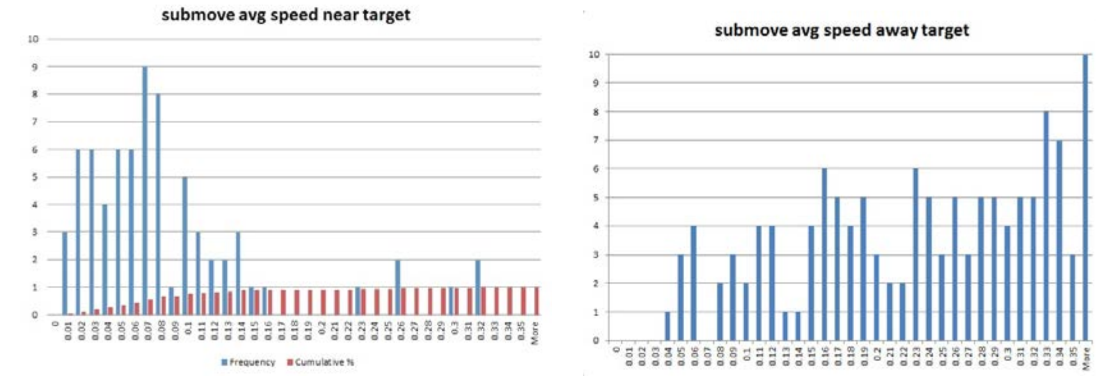


Figure A.2: (Participant 1) Sub-movement average speed distribution near target (left) and away from target (right).

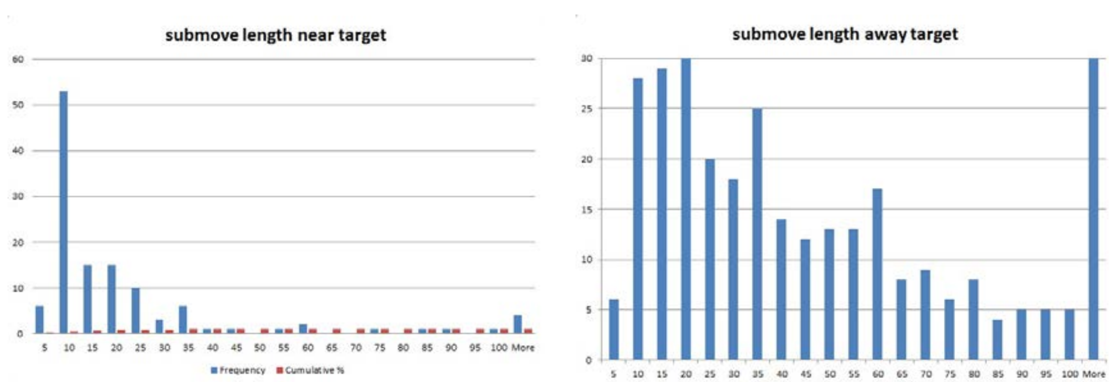


Figure A.3: (Participant 4) Sub-movement length distribution near target (left) and away from target (right).

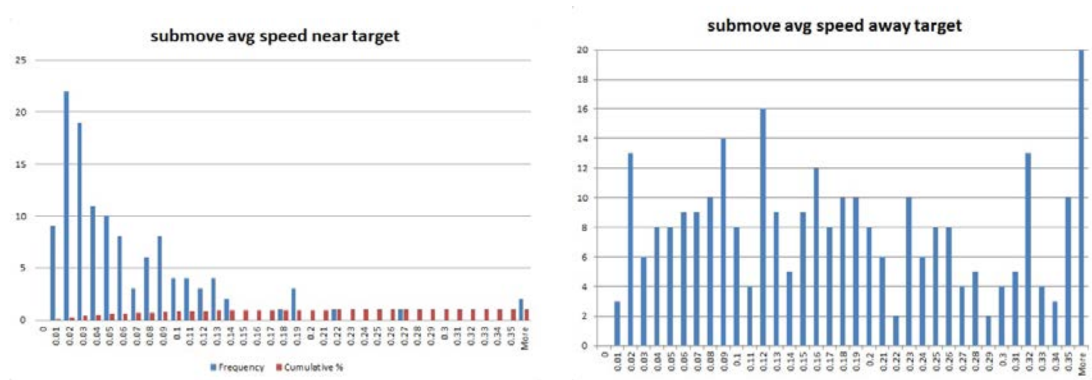


Figure A.4: (Participant 4) Sub-movement average speed distribution near target (left) and away from target (right).

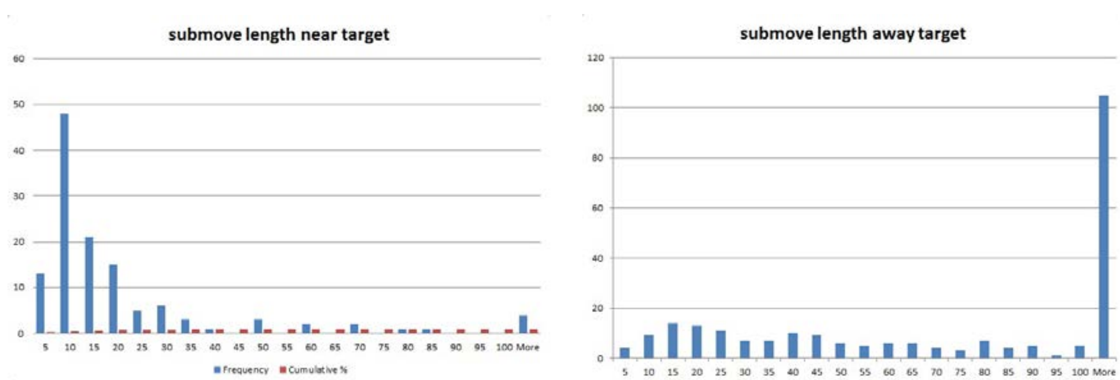


Figure A.5: (Participant 7) Sub-movement length distribution near target (left) and away from target (right).

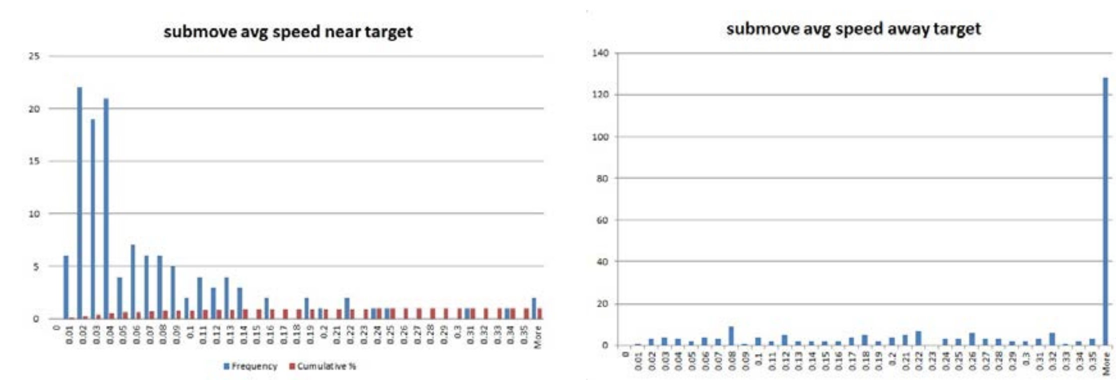


Figure A.6: (Participant 7) Sub-movement average speed distribution near target (left) and away from target (right).

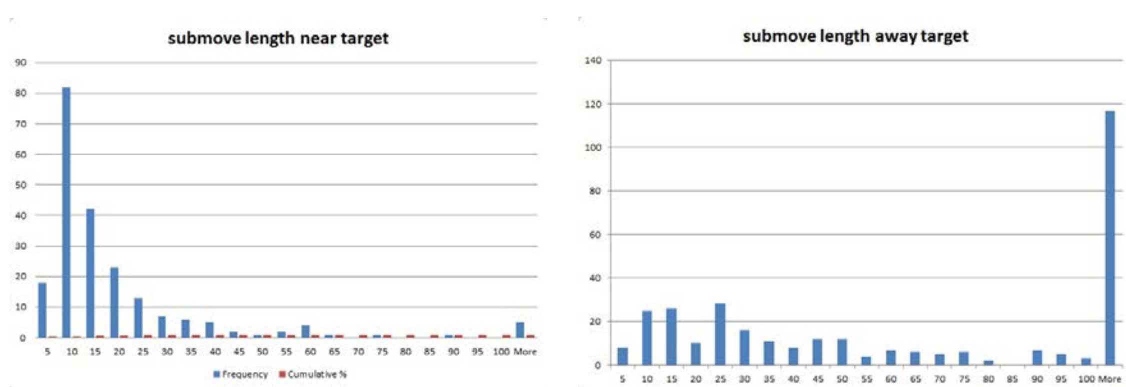


Figure A.7: (Participant 8) Sub-movement length distribution near target (left) and away from target (right).

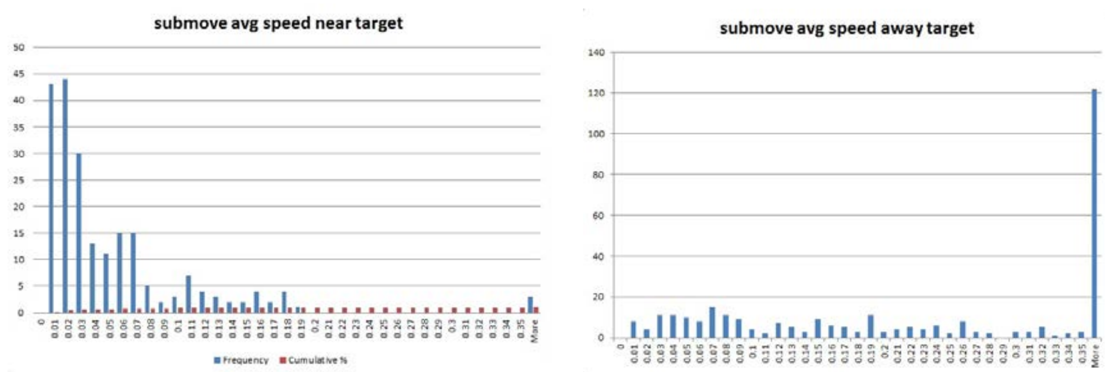


Figure A.8: (Participant 8) Sub-movement average speed distribution near target (left) and away from target (right).

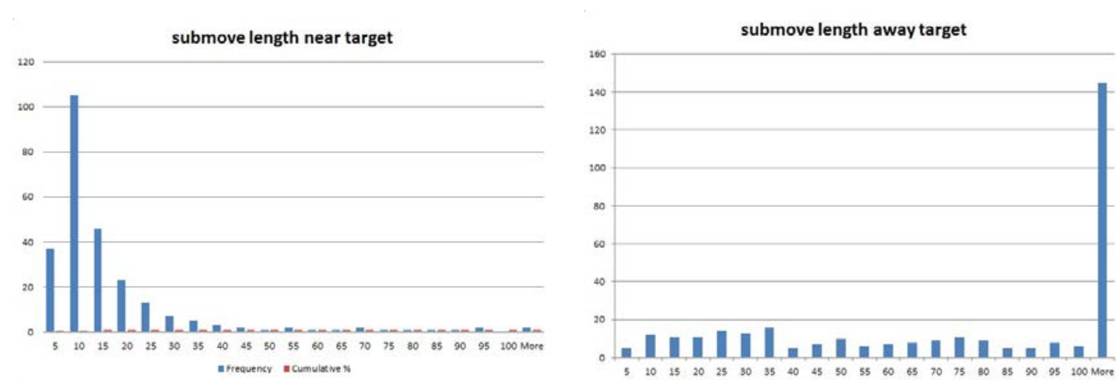


Figure A.9: (Participant 9) Sub-movement length distribution near target (left) and away from target (right).

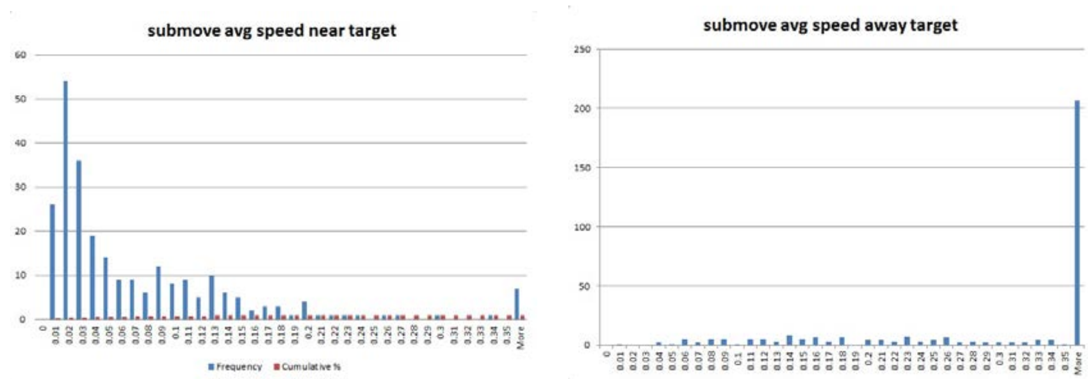


Figure A.10: (Participant 9) Sub-movement average speed distribution near target (left) and away from target (right).

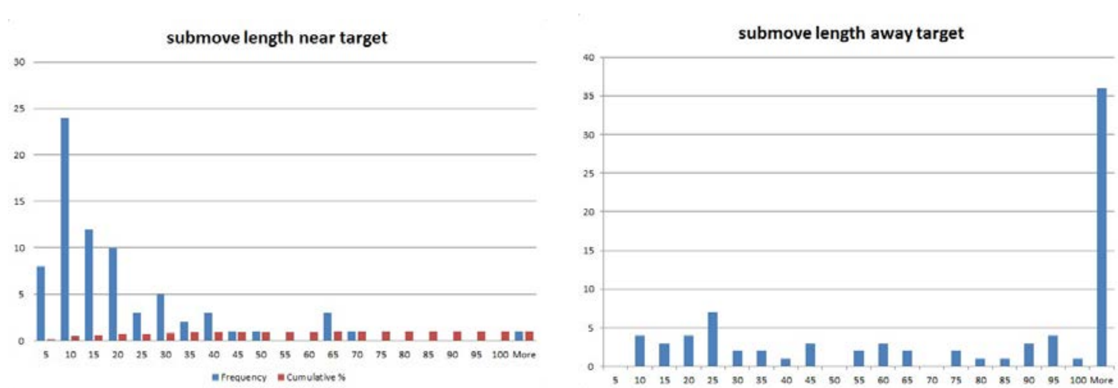


Figure A.11: (Participant 11) Sub-movement length distribution near target (left) and away from target (right).

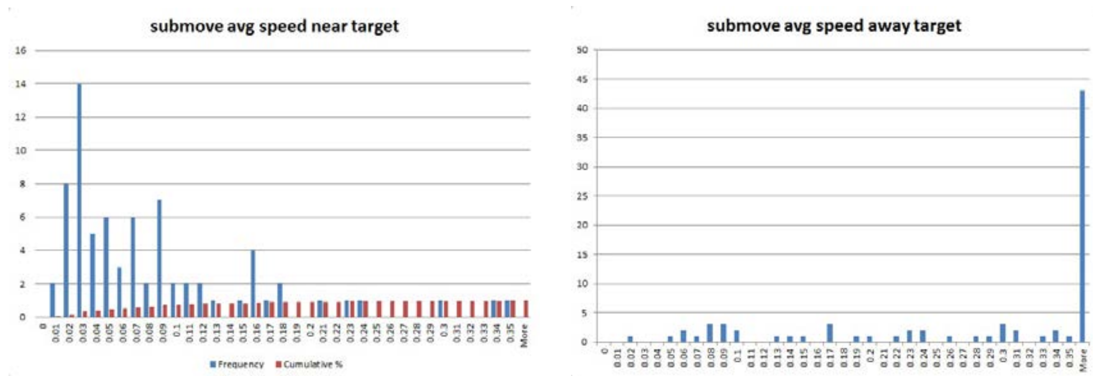


Figure A.12: (Participant 11) Sub-movement average speed distribution near target (left) and away from target (right).

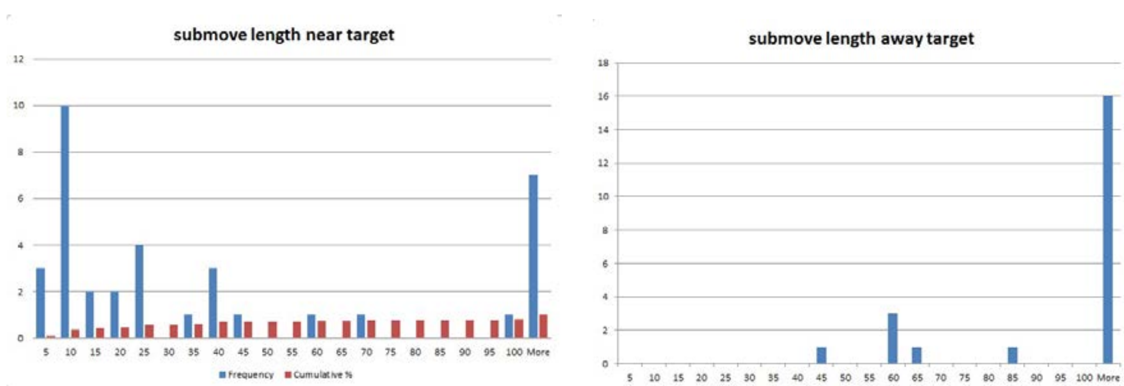


Figure A.13: (Participant 12) Sub-movement length distribution near target (left) and away from target (right).

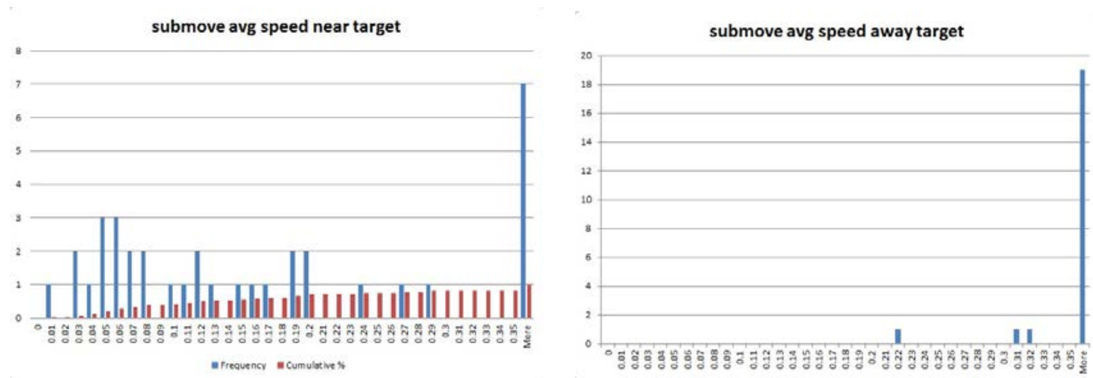


Figure A.14: (Participant 12) Sub-movement average speed distribution near target (left) and away from target (right).

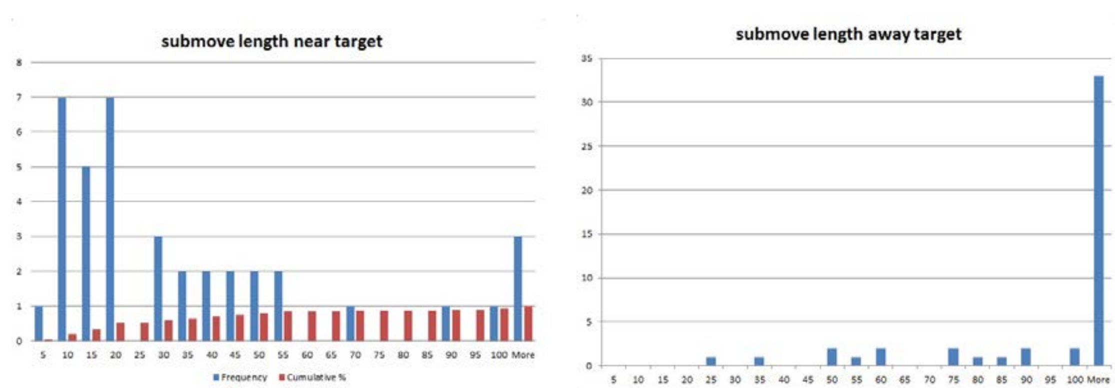


Figure A.15: (Participant 13) Sub-movement length distribution near target (left) and away from target (right).

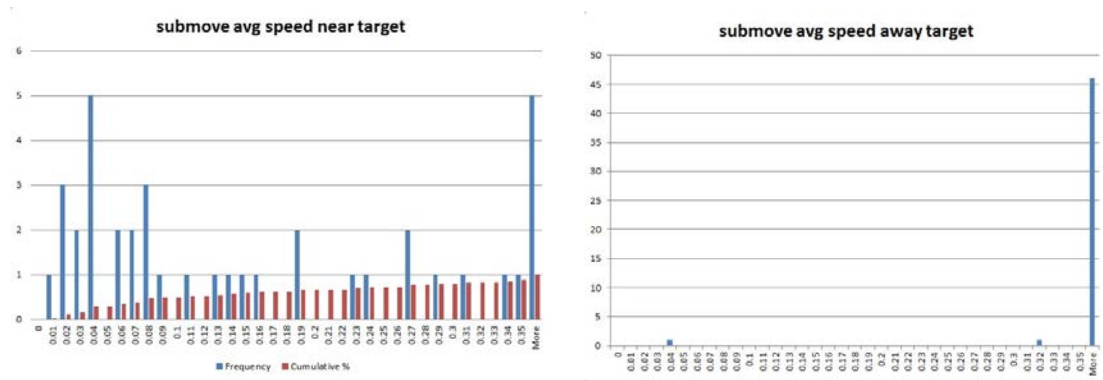


Figure A.16: (Participant 13) Sub-movement average speed distribution near target (left) and away from target (right).

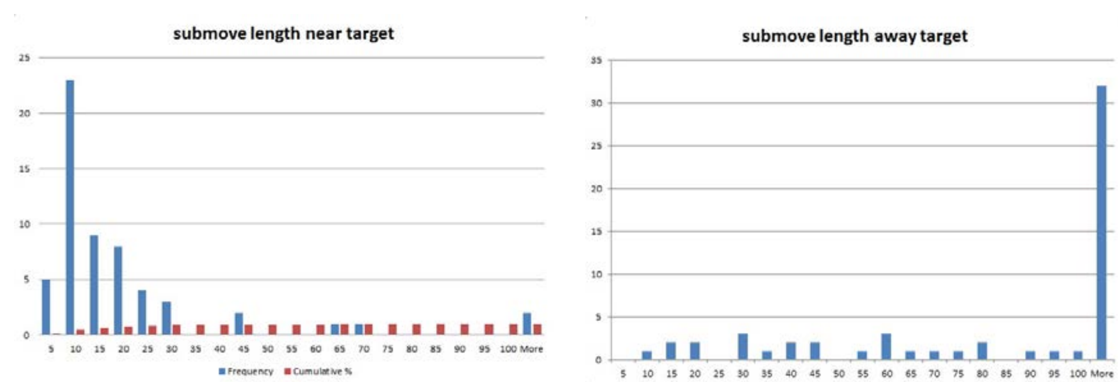


Figure A.17: (Participant 14) Sub-movement length distribution near target (left) and away from target (right).

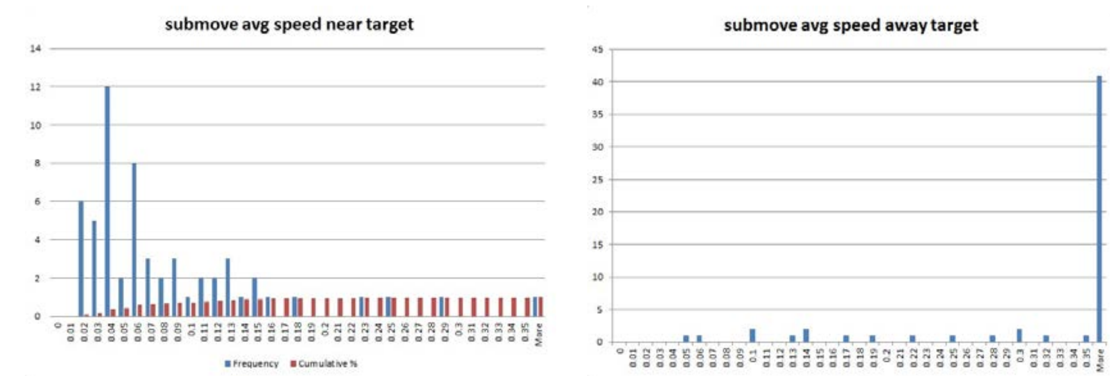


Figure A.18: (Participant 14) Sub-movement average speed distribution near target (left) and away from target (right).

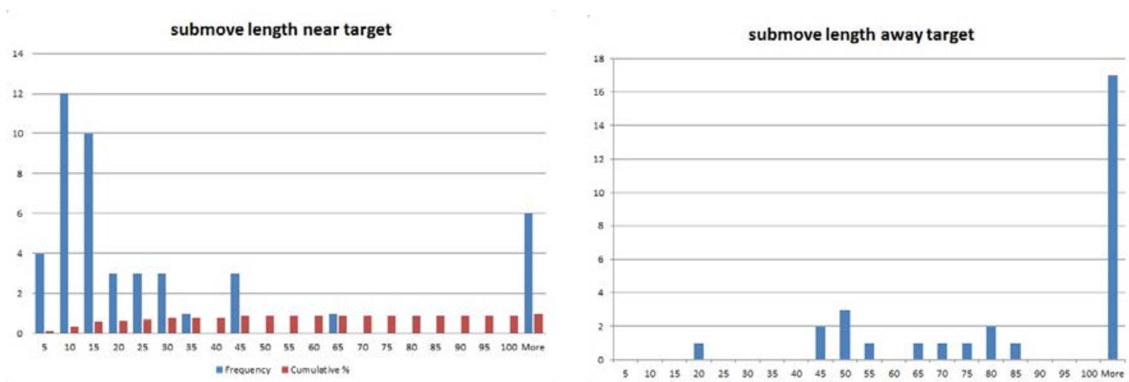


Figure A.19: (Participant 16) Sub-movement length distribution near target (left) and away from target (right).

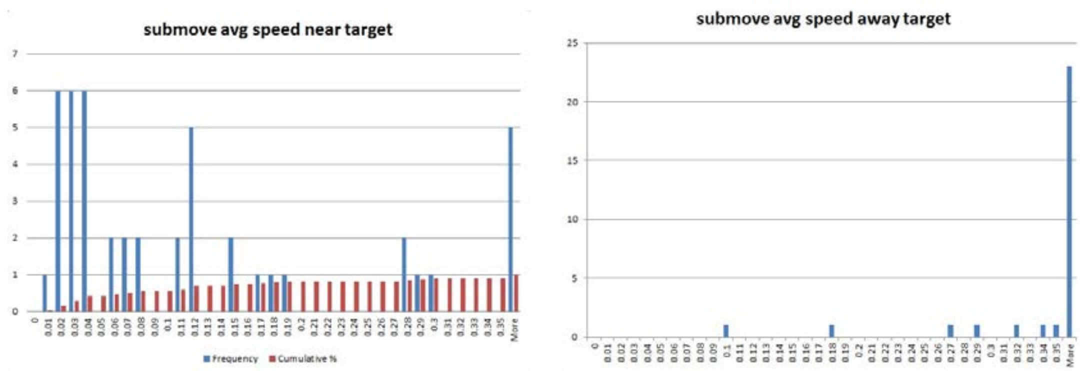


Figure A.20: (Participant 16) Sub-movement average speed distribution near target (left) and away from target (right).

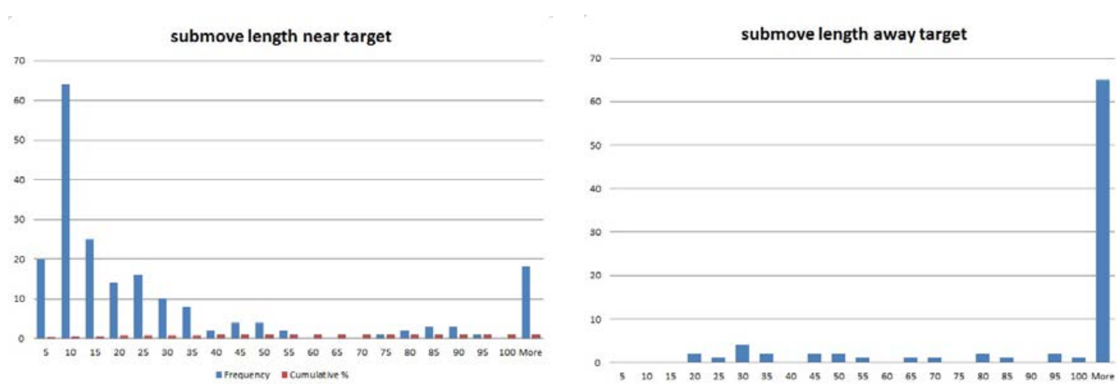


Figure A.21: (Participant 18) Sub-movement length distribution near target (left) and away from target (right).

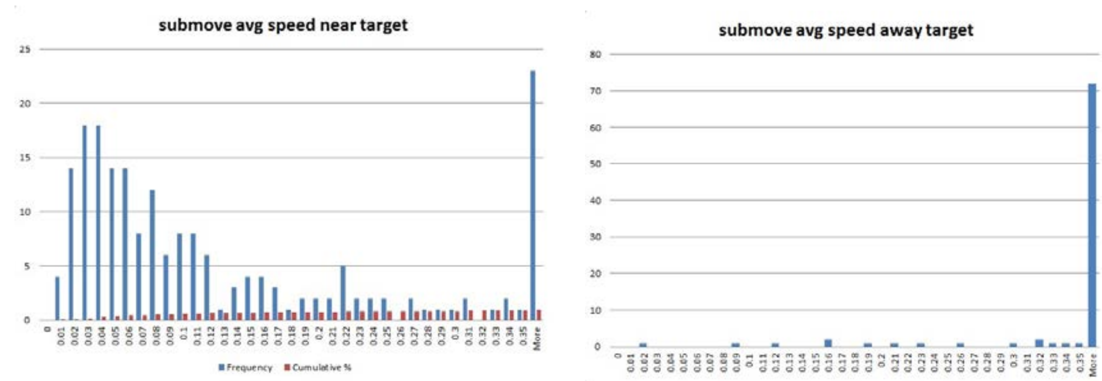


Figure A.22: (Participant 18) Sub-movement average speed distribution near target (left) and away from target (right).

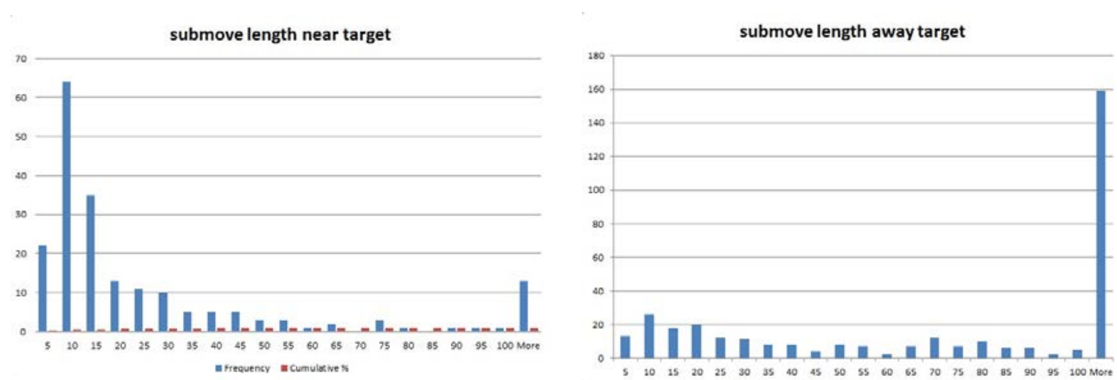


Figure A.23: (Participant 19) Sub-movement length distribution near target (left) and away from target (right).

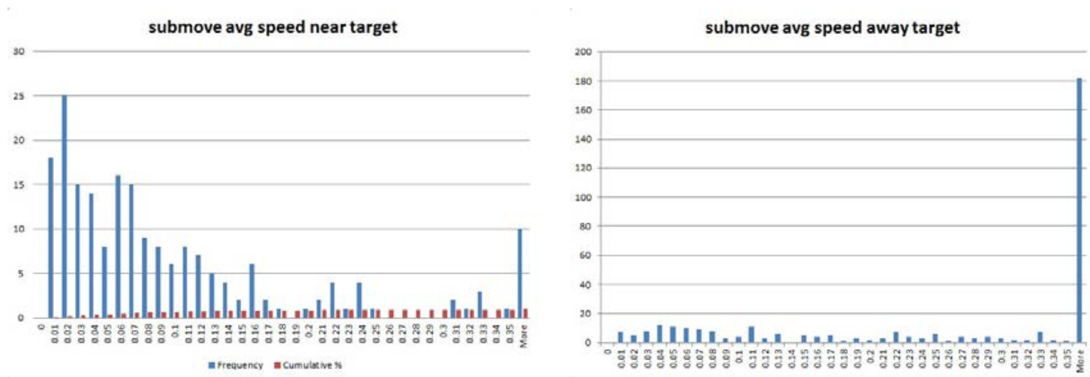


Figure A.24: (Participant 19) Sub-movement average speed distribution near target (left) and away from target (right).

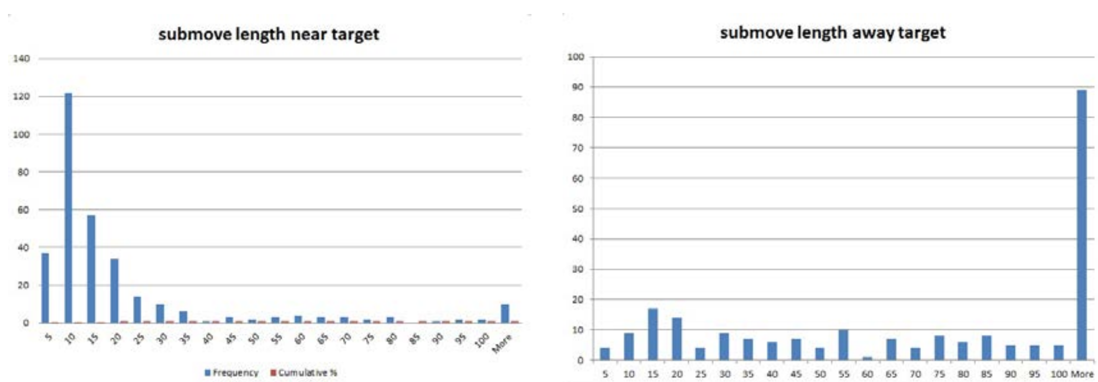


Figure A.25: (Participant 20) Sub-movement length distribution near target (left) and away from target (right).

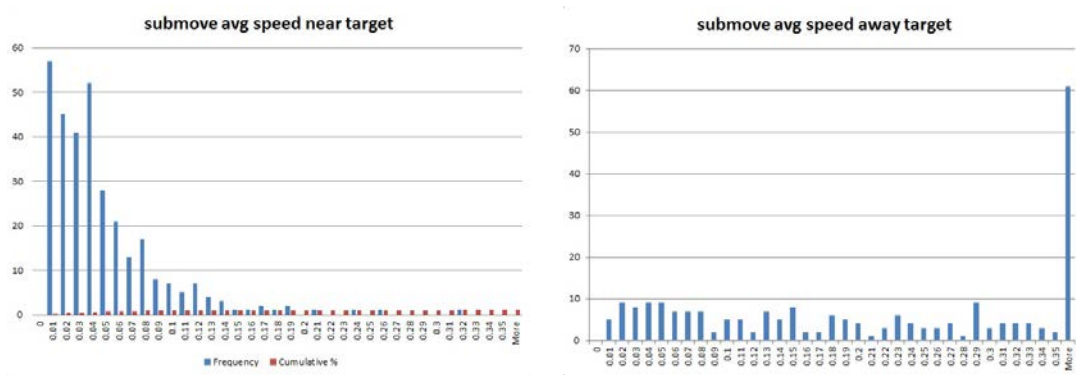


Figure A.26: (Participant 20) Sub-movement average speed distribution near target (left) and away from target (right).

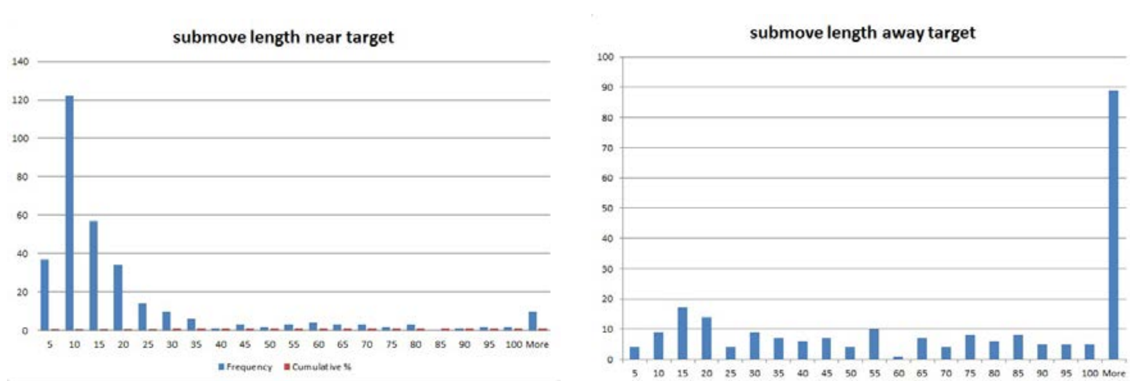


Figure A.27: (Participant 22) Sub-movement length distribution near target (left) and away from target (right).

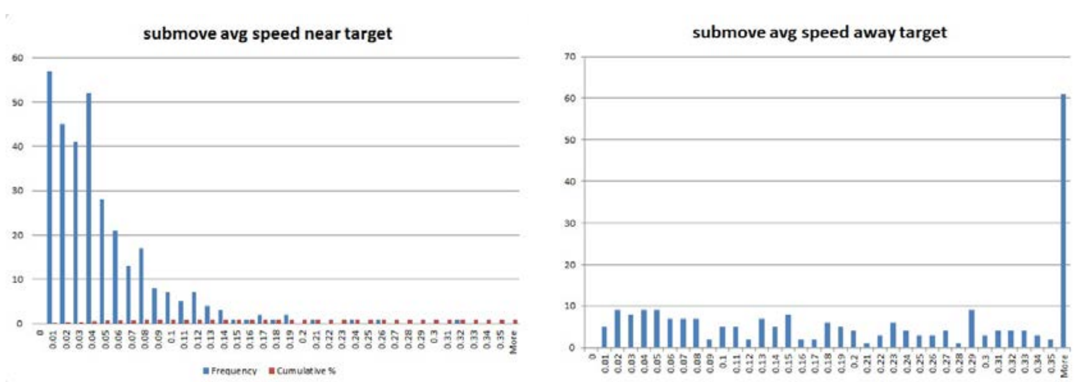


Figure A.28: (Participant 22) Sub-movement average speed distribution near target (left) and away from target (right).

APPENDIX B
PERFORMANCE IMAGES FOR ALL PARTICIPANTS: LONGITUDINAL STUDY

Pairs of images correspond to a day of testing with two blocks each day during a total 10 days over the course of 5 weeks. The images are organize in order of occurrence and labeled accordingly with a numeral from 1-20.

B.1 Alice

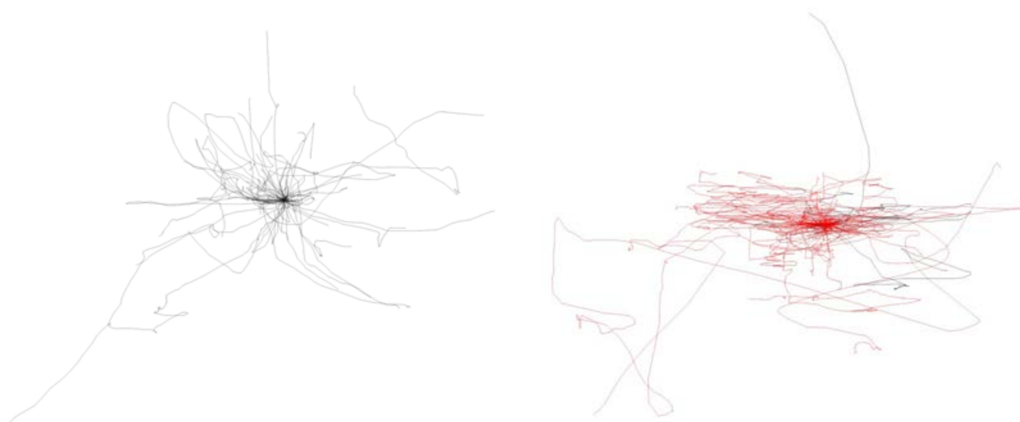


Figure B.1: Normalized sample blocks A1 (left) and B2 (right). Paths in red represent instances where precision-mode activated.

Figure B.2: Normalized sample blocks A3 (left) and B4 (right). Paths in red represent instances where precision-mode activated.

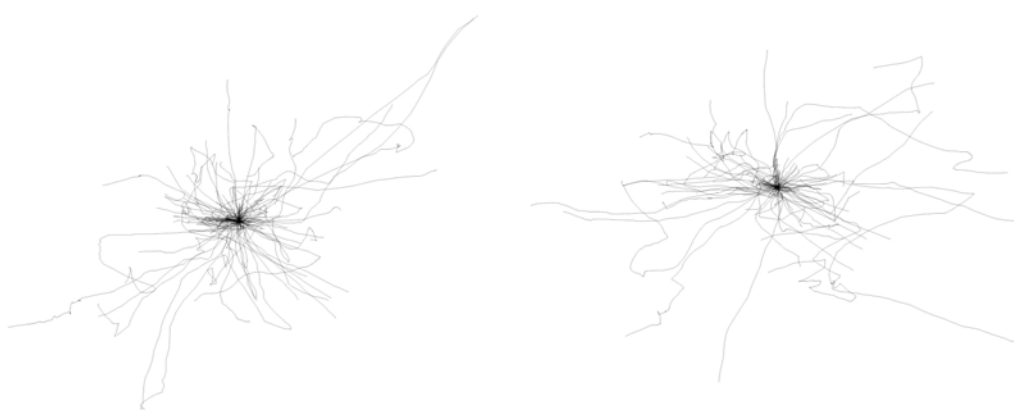


Figure B.3: Normalized sample blocks A5 (left) and A6 (right). Paths in red represent instances where precision-mode activated.

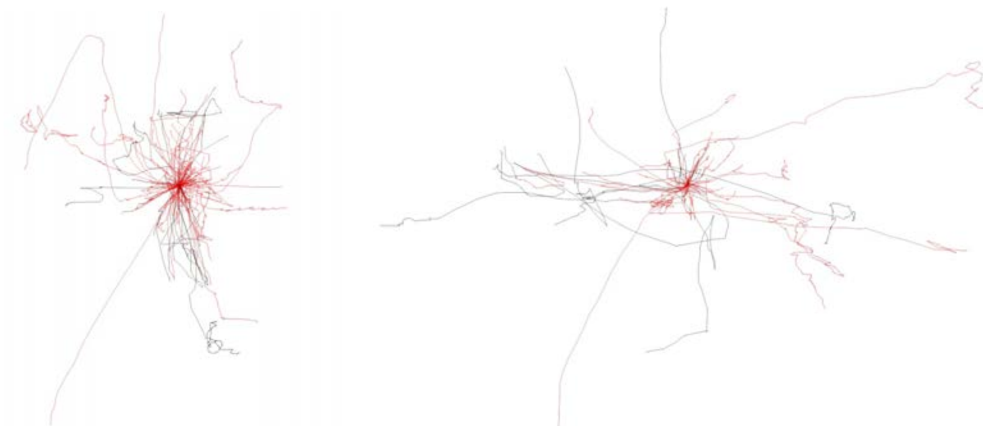


Figure B.4: Normalized sample blocks B7 (left) and B8 (right). Paths in red represent instances where precision-mode activated.

Figure B.5: Normalized sample blocks B9 (left) and B10 (right). Paths in red represent instances where precision-mode activated.

Figure B.6: Normalized sample blocks B11 (left) and A12 (right). Paths in red represent instances where precision-mode activated.

Figure B.7: Normalized sample blocks A13 (left) and A14 (right). Paths in red represent instances where precision-mode activated.

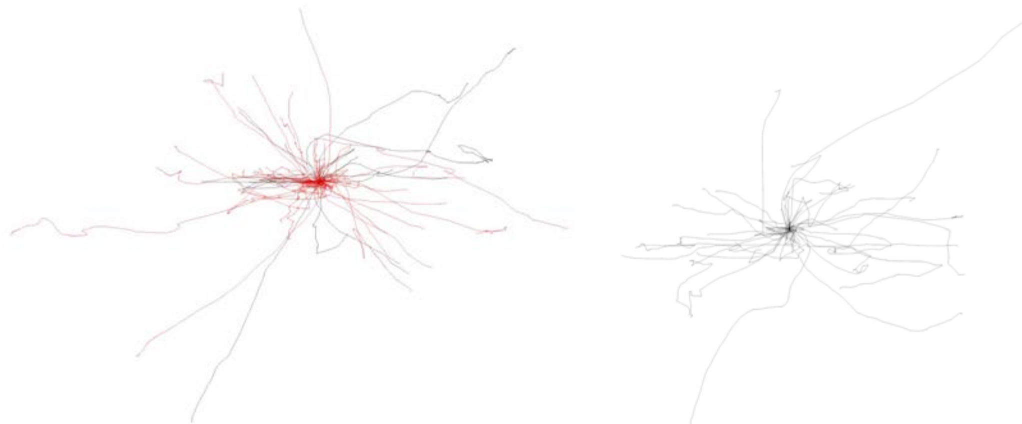


Figure B.8: Normalized sample blocks B15 (left) and A16 (right). Paths in red represent instances where precision-mode activated.

Figure B.9: Normalized sample blocks A17 (left) and A18 (right). Paths in red represent instances where precision-mode activated.

Figure B.10: Normalized sample blocks B19 (left) and B20 (right). Paths in red represent instances where precision-mode activated.

B.2 Karl

Figure B.11: Normalized sample blocks B1 (left) and A2 (right). Paths in red represent instances where precision-mode activated.

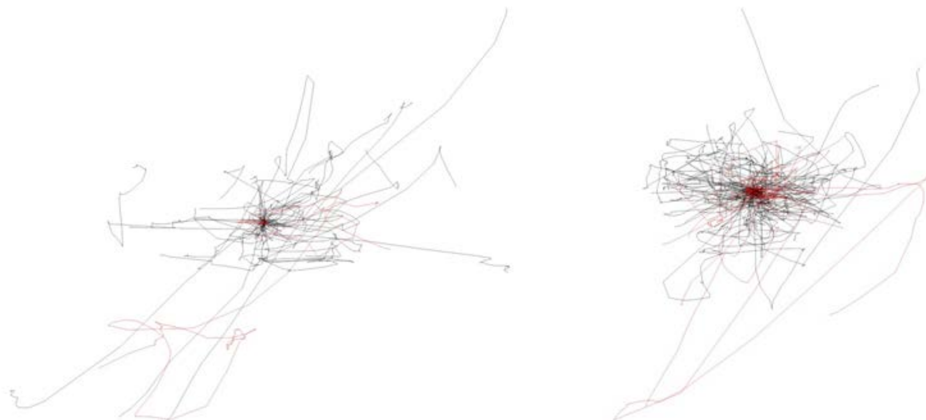


Figure B.12: Normalized sample blocks B3 (left) and B4 (right). Paths in red represent instances where precision-mode activated.

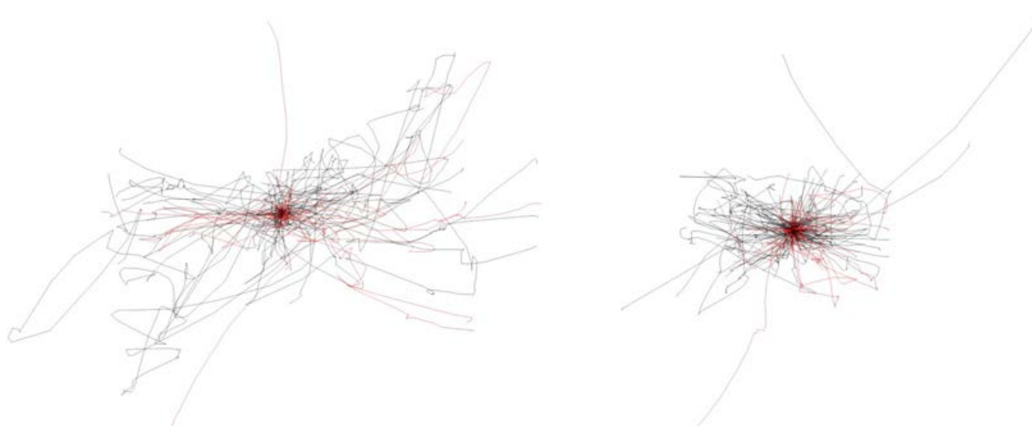


Figure B.13: Normalized sample blocks B5 (left) and B6 (right). Paths in red represent instances where precision-mode activated.

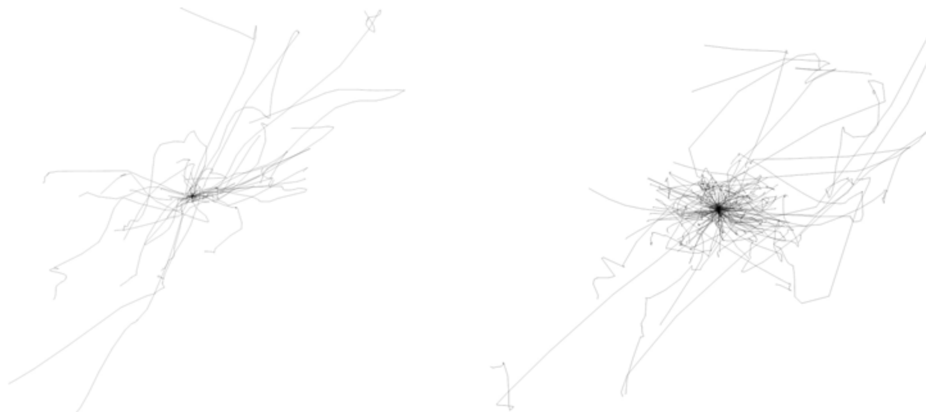


Figure B.14: Normalized sample blocks A7 (left) and A8 (right). Paths in red represent instances where precision-mode activated.

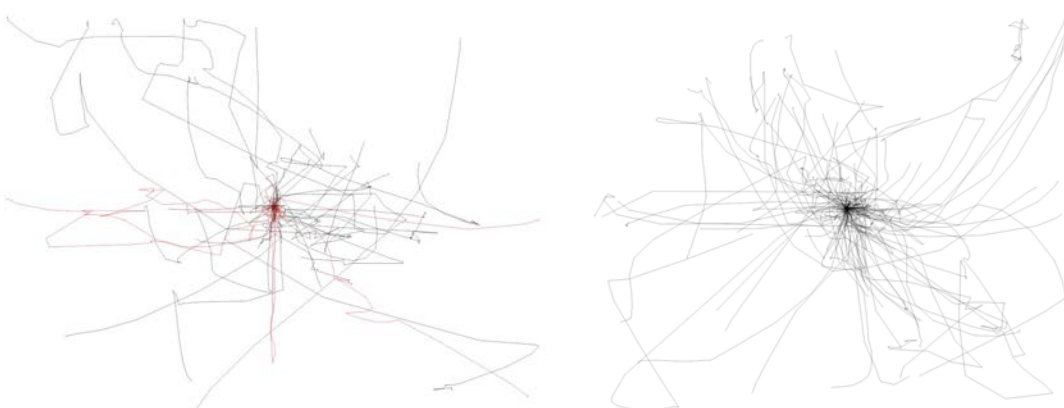


Figure B.15: Normalized sample blocks B9 (left) and A10 (right). Paths in red represent instances where precision-mode activated.

Figure B.16: Normalized sample blocks A11 (left) and B12 (right). Paths in red represent instances where precision-mode activated.

Figure B.17: Normalized sample blocks A13 (left) and A14 (right). Paths in red represent instances where precision-mode activated.

Figure B.18: Normalized sample blocks A15 (left) and B16 (right). Paths in red represent instances where precision-mode activated.

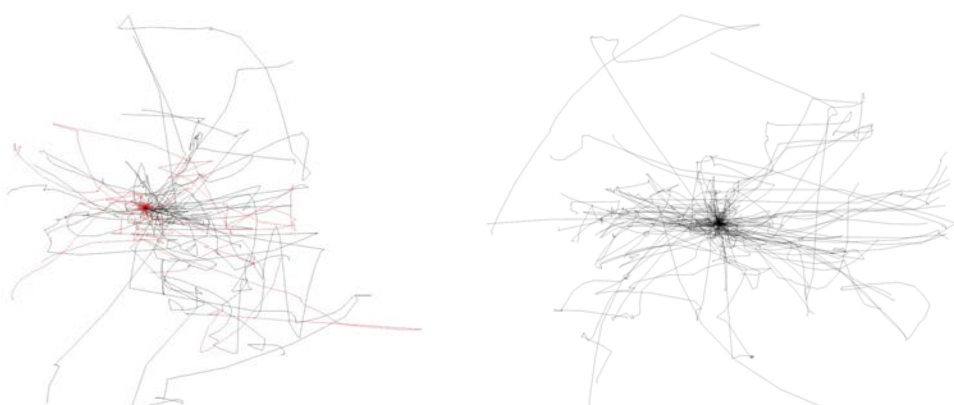


Figure B.19: Normalized sample blocks B17 (left) and A18 (right). Paths in red represent instances where precision-mode activated.

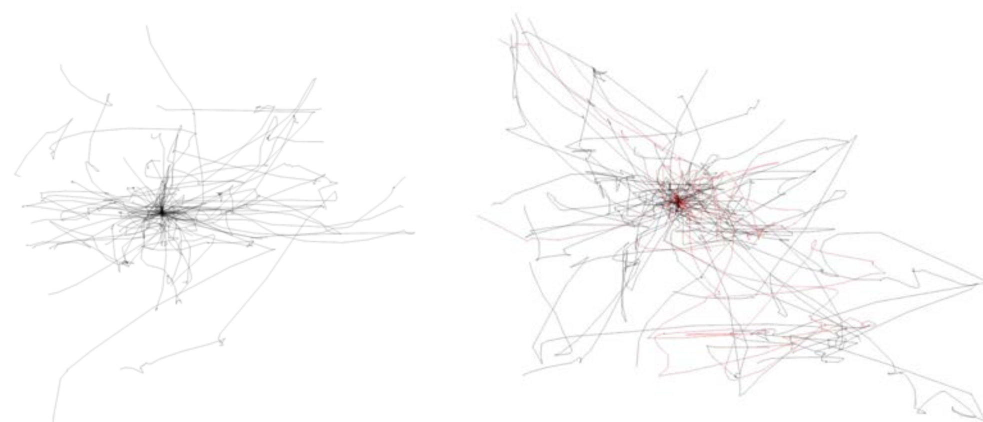


Figure B.20: Normalized sample blocks A19 (left) and B20 (right). Paths in red represent instances where precision-mode activated.

B.3 Joe

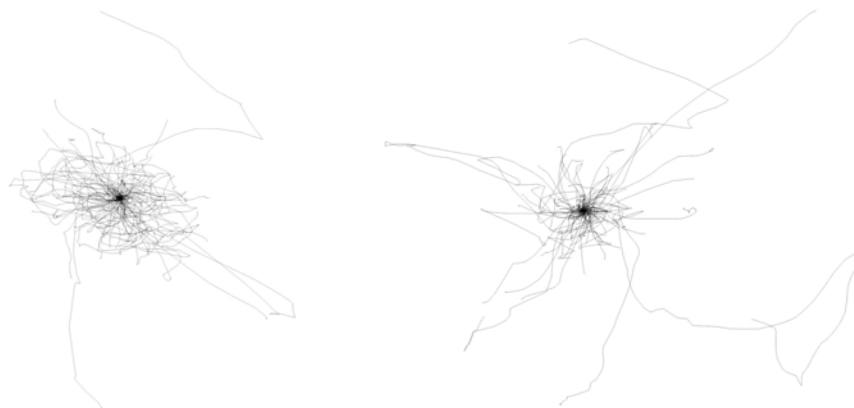


Figure B.21: Normalized sample blocks A1 (left) and A2 (right). Paths in red represent instances where precision-mode activated.

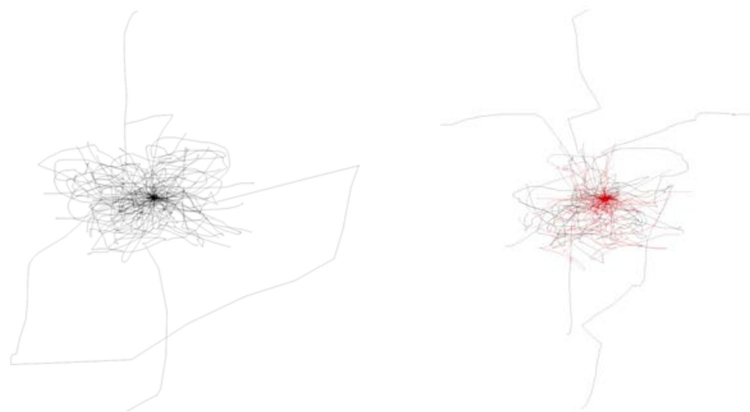


Figure B.22: Normalized sample blocks A3 (left) and B4 (right). Paths in red represent instances where precision-mode activated.

Figure B.23: Normalized sample blocks A5 (left) and B6 (right). Paths in red represent instances where precision-mode activated.

Figure B.24: Normalized sample blocks B7 (left) and A8 (right). Paths in red represent instances where precision-mode activated.

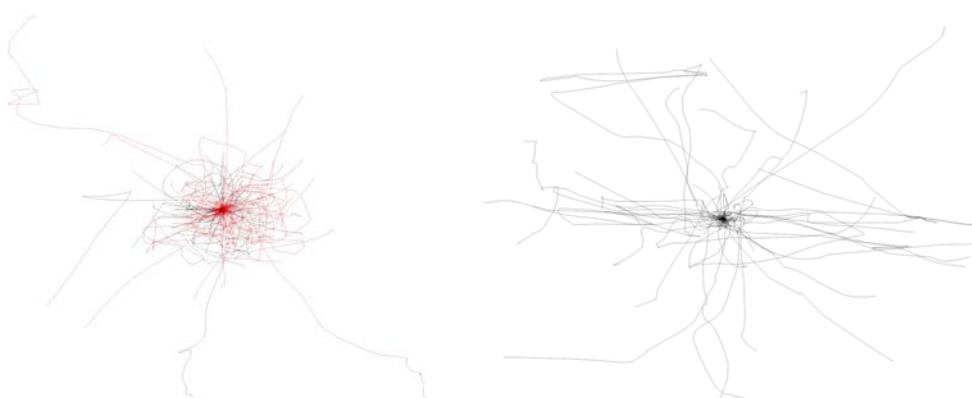


Figure B.25: Normalized sample blocks B9 (left) and A10 (right). Paths in red represent instances where precision-mode activated.

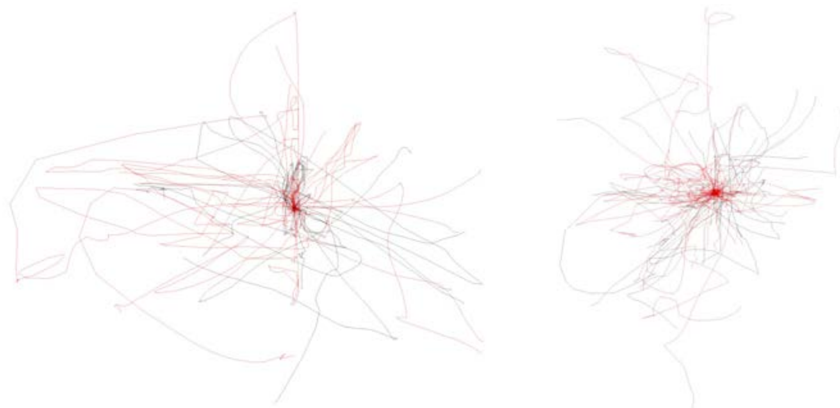


Figure B.26: Normalized sample blocks B11 (left) and B12 (right). Paths in red represent instances where precision-mode activated.

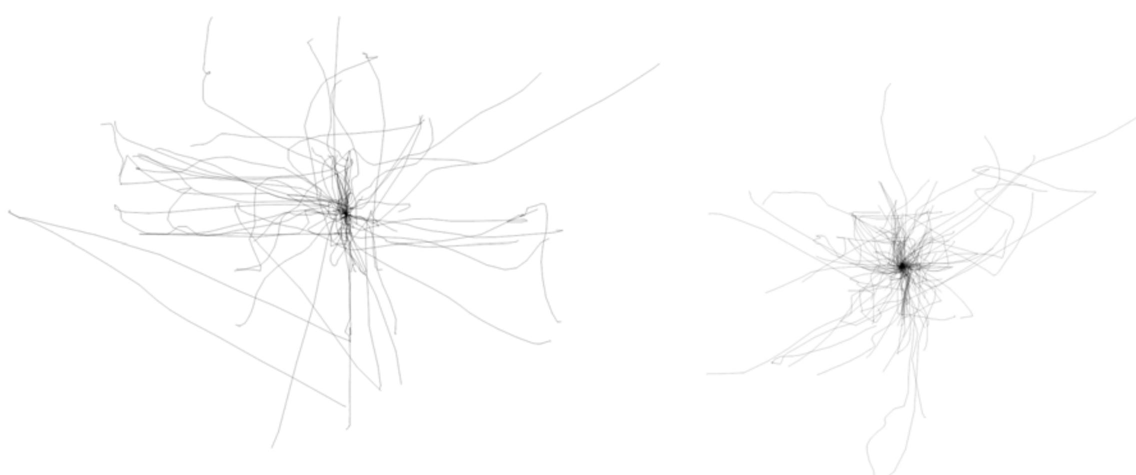


Figure B.27: Normalized sample blocks A13 (left) and A14 (right). Paths in red represent instances where precision-mode activated.

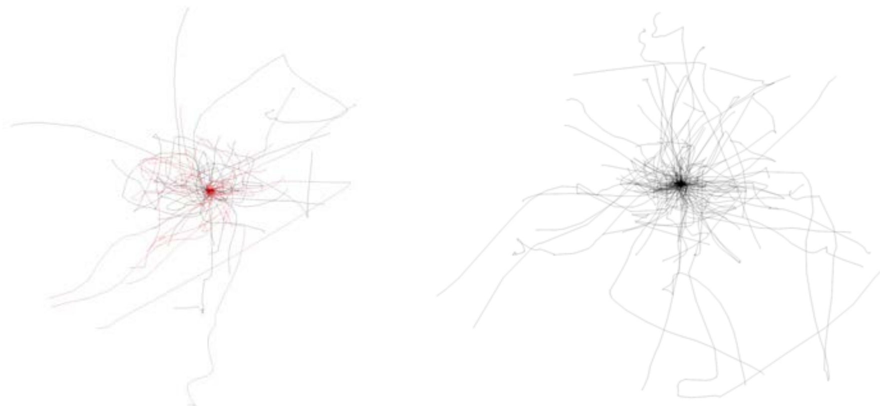


Figure B.28: Normalized sample blocks B15 (left) and A16 (right). Paths in red represent instances where precision-mode activated.

Figure B.29: Normalized sample blocks B17 (left) and B18 (right). Paths in red represent instances where precision-mode activated.

Figure B.30: Normalized sample blocks A19 (left) and B20 (right). Paths in red represent instances where precision-mode activated.

APPENDIX C
LONGITUDINAL STUDY: OTHER RESULTS

C.1 Sub-movement characteristics

Karl's sub-movement characteristics saw an effect in the length and average speeds. For the length we found $t(17) = -2.35$, $p = .03$ with $(M = 10.9, SD = .54)$ for no assistance and $(M = 11.4, SD = .44)$ for assistance (see figure C.1).

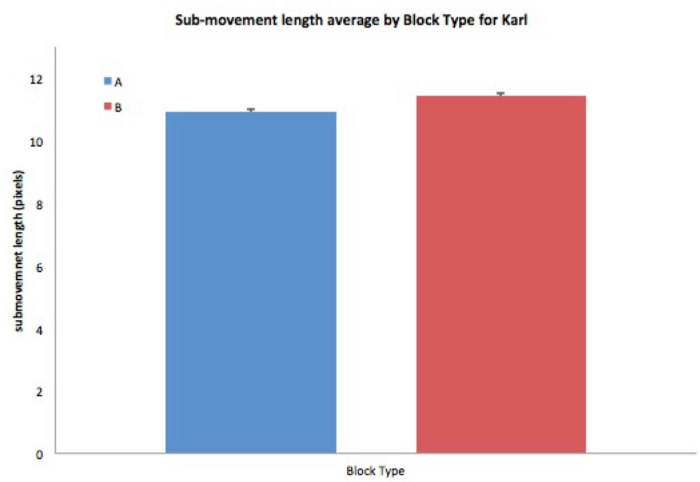


Figure C.1: Average sub-movement length (mean ± SEM) for tasks performed by Karl. Blocks of type A mean no assistance was provided (left blue bar) and blocks of type B mean assistance was provided (right red bar).

For the average speed we found $t(18) = -3.18$, $p = 0.005$ with $(M = .043, SD = .0033)$ for no assistance and $(M = .048, SD = .0038)$ for assistance (see figure C.2). Kolmogorv-Smirnov tests of normality were not significant with $p = .2$ for both sub-movement length and average speed indicating that the data is normally distributed.

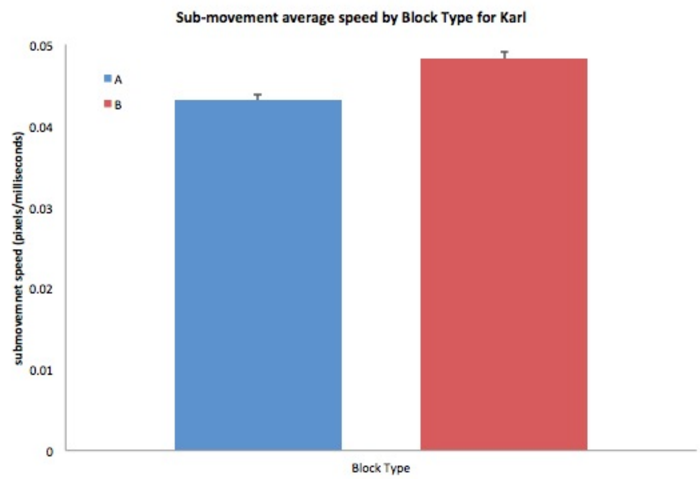


Figure C.2: Sub-movement average speed (mean ± SEM) for tasks performed by Karl. Blocks of type A mean no assistance was provided (left blue bar) and blocks of type B mean assistance was provided (right red bar).

C.2 Sub-movement counts

We looked at the number of sub-movements for different task lengths and found statistically significant results for the number of sub-movements for all participants in different instances (see table C.1). Alice showed a statistically significant effect in the average number of sub-movements away from a click (more than 64 pixels away from a click) for tasks of length between 384 and 512 pixels with $U = 21, Z = -1.96, p = .053$ and $r = .45$, where the medians of blocks type A and blocks type B were 10.7 and

Table C.1: Sub-movement count medians for blocks $A \mid B$, away and near a click for each participant. Sub-movements near are less than 64 pixels from a click. Sub-movements away are more than 64 pixels from a click.

		median		Mann-Whitney U-Test		p	
Participant	length x of tasks in pixels	away	near	away	near	away	near
Alice	$x \leq 128$	2.3|1.7	3.7|4.1	-1.4	-.38	.17	.74
	$128 < x \leq 256$	8.3|8.6	4.3|4.5	-.08	-.15	.97	.91
	$256 < x \leq 384$	10.3|9.1	4.7|3.6	-1.6	-.9	.13	.40
	$384 < x \leq 512$	10.7|14.8	4.4|3.6	-2.0	-.25	.053	.84
	$x > 512$	12.5|18.1	3.8|3.9	-2.2	-.38	.029	.74
Karl	$x \leq 128$	2.4|2.2	1.9|2.8	-.76	-3.2	.48	.001
	$128 < x \leq 256$	3.5|5.1	2.1|2.5	-1.2	-1.6	.25	.12
	$256 < x \leq 384$	3.8|5.5	2.5|2.7	-1.4	-1.1	.19	.28
	$384 < x \leq 512$	6.9|9.3	2.8|2.9	-1.2	-.23	.25	.85
	$x > 512$	6.8|8.9	3.0|3.1	-1.1	-.91	.28	.39
Joe	$x \leq 128$	.69|1.7	2.7|2.8	-1.7	.00	.089	1.0
	$128 < x \leq 256$	2.7|4.3	3.5|3.5	-1.2	-.76	.25	.48
	$256 < x \leq 384$	5.2|5.3	3.3|3.2	-.61	-.53	.58	.63
	$384 < x \leq 512$	4.0|5.6	3.2|3.5	-1.2	-.91	.25	.39
	$x > 512$	6.5|7.9	3.6|3.9	-1.1	-.42	.29	.68

14.83 respectively (see figure C.3). We left the the discussion of the results for Karl and Joe in the appendix C being that we did not find significant results.

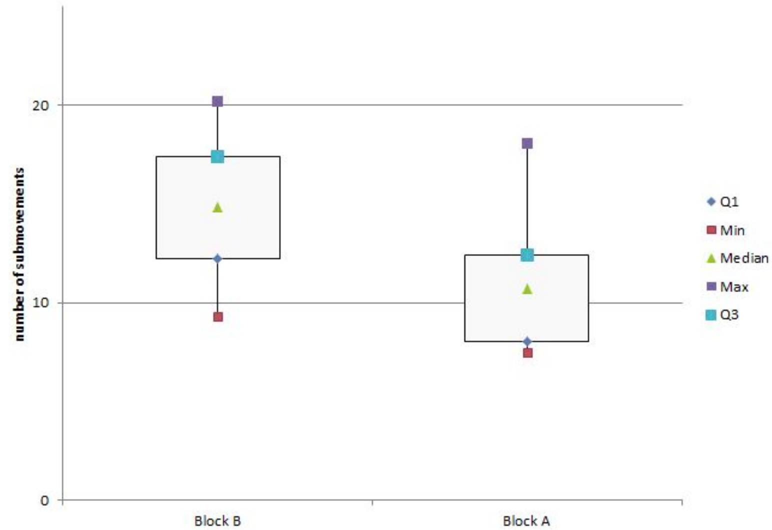


Figure C.3: Distribution of number of sub-movements more than 64 pixels from a click for tasks performed by Alice of lengths from 384 pixels to 512 pixels. Blocks of type A mean no assistance was provided and blocks of type B mean assistance was provided.

Similarly we found statistical significance in the average number of sub-movements away from a click for tasks longer than 512 pixels with $U = 21, Z = -2.19, p = .028$ and $r = .49$, where the medians of blocks type A and blocks type B were 12.49 and 18.12 respectively (see figure C.4).

For Karl we found statistical significance in the average number of sub-movements near a click for tasks shorter than 128 pixels with $U = 8, Z = -3.18, p = .001$ and

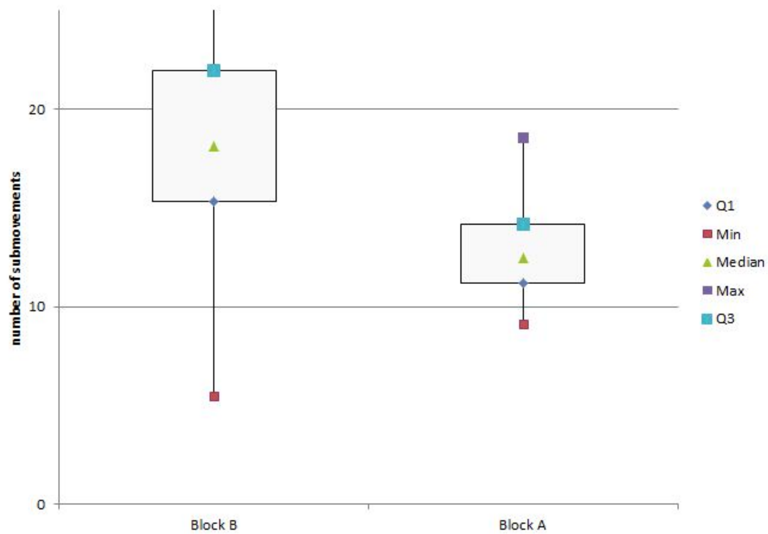


Figure C.4: Distribution of number of sub-movements more than 64 pixels from a click for tasks performed by Alice of lengths longer 512 pixels. Blocks of type A mean no assistance was provided and blocks of type B mean assistance was provided.

$r = .71$, where the medians of blocks type A and blocks type B were 1.85 and 2.82 respectively (see figure C.5).

Joe's results regarding the number of sub-movements were inconclusive. We did however find assistance to have a marginally significant effect in the number of sub-movements away from a click for tasks shorter than 128 pixels with $U = 27, Z = -1.74, p = .082$ and $r = .39$, where the medians of blocks type A and blocks type B were 1.74 and .70 respectively (see figure C.6).

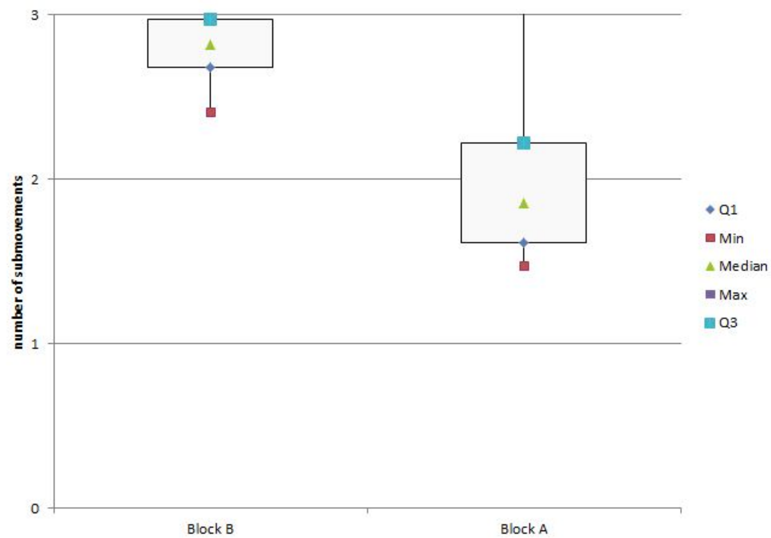


Figure C.5: Distribution of number of sub-movements less than 64 pixels from a click for tasks performed by Karl of lengths shorter than 128 pixels. Blocks of type A mean no assistance was provided and blocks of type B mean assistance was provided.

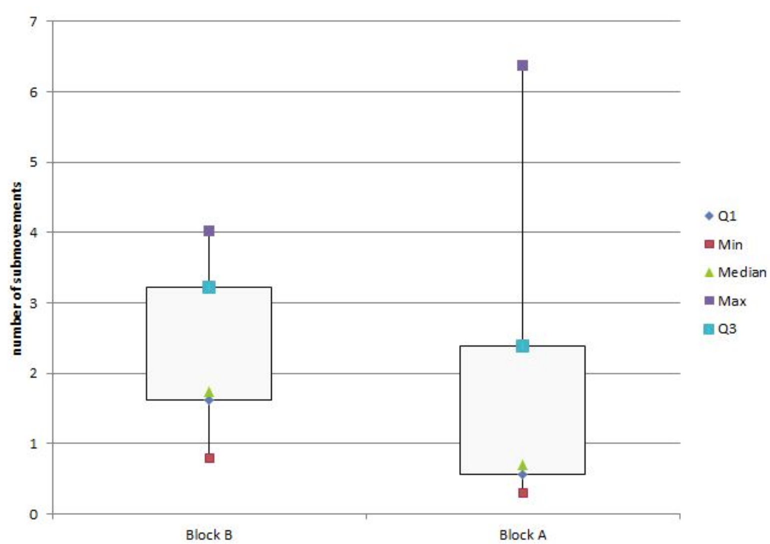


Figure C.6: Distribution of number of sub-movements more than 64 pixels from a click for tasks performed by Joe of lengths shorter than 128 pixels. Blocks of type A mean no assistance was provided and blocks of type B mean assistance was provided.

C.3 Slip rates, average distance from press to release and close clicks ratio

Table C.2: Means and standard deviations of slip rates, average distance from press to release and close clicks ratio for each participant

	Alice			Karl			Joe		
Variable	mean	sd	t stat	mean	sd	t stat	mean	sd	t stat
slips rate A	.038	.05	-.9	.011	.016	.41	.033	.04	-.55
slips rate B	.067	.09		.008	.015		.063	.17	
press to release A	26.9	30.9	-.15	10.0	12.1	-.53	7.8	4.9	-.69
press to release B	28.8	28.3		15.8	32.7		14.9	32.5	
close clicks A	.07	.06	-1.9	.08	.06	-.95	.23	.12	.08
close clicks B	.17	.02		.13	.16		.22	.17	

A summary of the results can be found in table C.2. Slip-rate and average distance from press to release variables are somewhat related to one another since they refer to comparisons between mouse press and mouse release instances. This also serves to suggest that these variables are good indicators of accuracy. Interestingly enough we see that the trends are very similar and we can verify this by looking at figures C.7, C.8 and C.9).

In the case of Alice she began performing very poorly in the first B block, gradually improving with time and then we see a peak at the 8th B block in both trends. For Karl we see the two peaks at the second and fourth A blocks as well as the peak at the eighth B block and a very similar final block pattern for A and B.

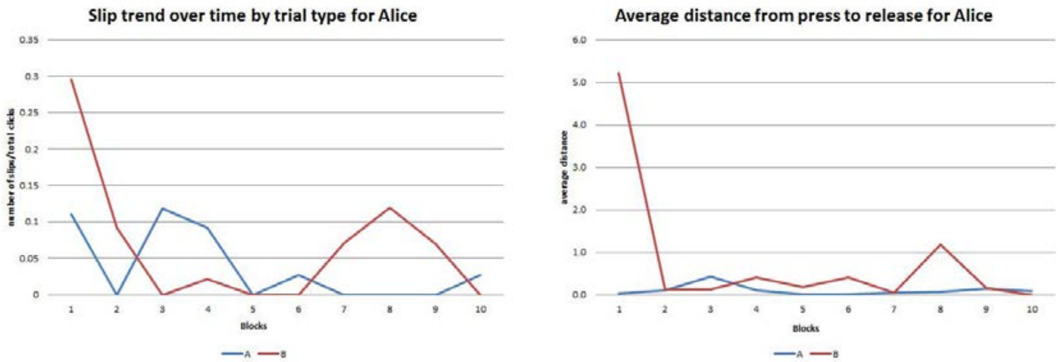


Figure C.7: Slip-rate (left) and distance from press to release trend (right) over time for Alice.

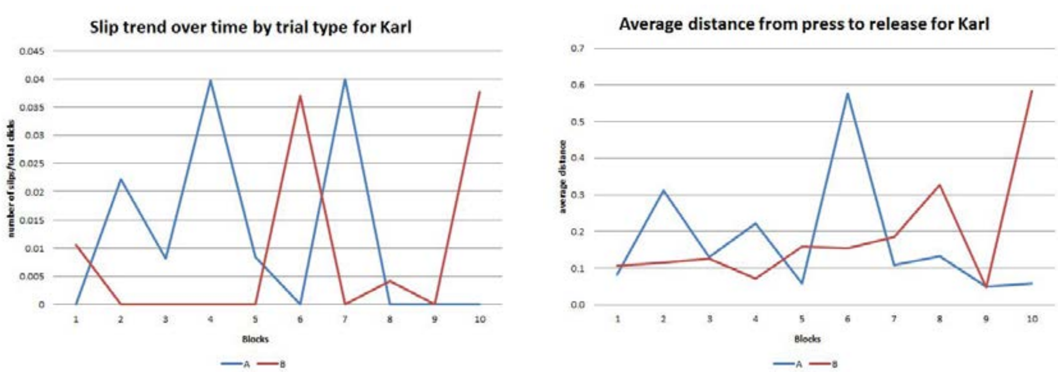


Figure C.8: Slip-rate (left) and distance from press to release trend (right) over time for Karl.

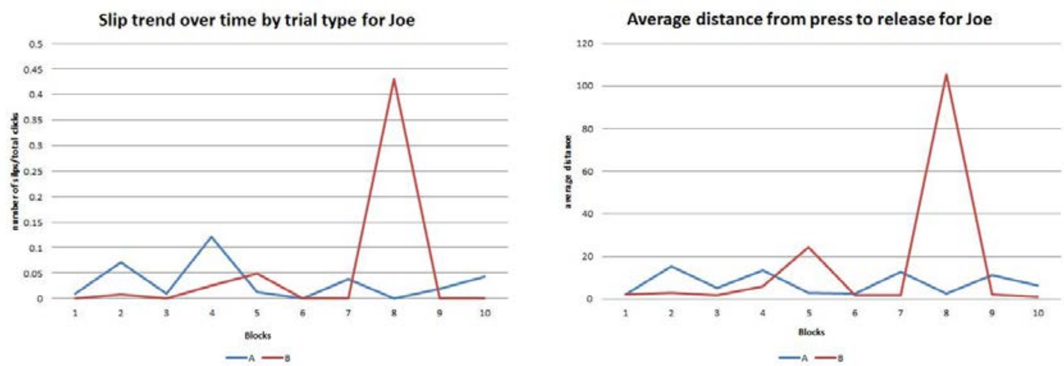


Figure C.9: Slip-rate (left) and distance from press to release trend (right) over time for Joe.

The B block curve seems to be below the A block curve in more instances though not consistently. Most impressive is the comparison of these trends in Joe's case. We dare say the trend is almost identical with the same peaks for all A and B blocks and the same pattern over time. We did not find statistically significant results for the slip-rate nor the average distance from press to release.

We looked at the ratio of total number of clicks sufficiently close to each other over the total number of clicks per block (see figure C.10 and table C.2 for summarized results). Karl did not seem to have any difference over time with assistance. Joe's trend on the other hand suggests that overall he did worse as time progressed. The curve for B almost always dominates the curve for A suggesting that he did better with assistance than without assistance.

We found that Alice showed marginally significant results of $t(12) = -1.89$, $p < 0.083$ with $(M = 0.07, SD = .06)$ for no assistance and $(M = 0.17, SD = .15)$

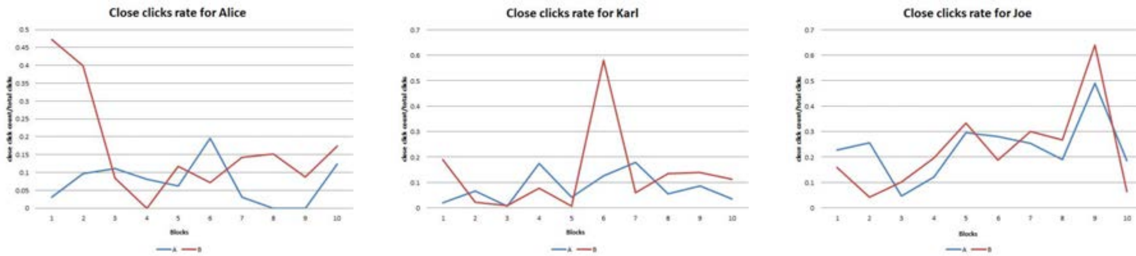


Figure C.10: Time trend by block of close clicks over total number of clicks for all participants.

for assistance. The trend in figure C.10 suggest that the first two B blocks maybe outliers and that in time she did better with assistance than without assistance.

REFERENCES

[1] S R Adapathya, F D Champion, J A Happ, M B Lawrence, and L K Schultz. Device driver syste for minimizing adverse tremor effects during use of pointing devices. United States Patent 6561993, 2001.

[2] D. Ahlström, M. Hitz, and G. Leitner. An evaluation of sticky and force enhanced targets in multi target situations. In *NordiCHI '06: Proceedings of the 4th Nordic conference on Human-computer interaction*, pages 58–67, New York, NY, USA, 2006. ACM.

[3] R. Balakrishnan. Beating fitts' law: virtual enhancements for pointing facilitation. *International Journal of Human-Computer Studies*, 61(6):857–874, December 2004.

[4] R. Benecke, J. Rothwell, J. Dick, and B. Day. Simple and complex movements off and on treatment in patients with parkinson's disease. *British Medical Journal*, 50(3):296–303, Jan 1987.

[5] R. Blanch, Y. Guiard, and M. Beaudouin-Lafon. Semantic pointing: improving target acquisition with control-display ratio adaptation. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 519–526, New York, NY, USA, Jan 2004. ACM.

[6] F. Boller, D. Passafiume, N. Keefe, and K. Rogers. Visuospatial impairment in parkinson's disease: role of perceptual and motor factors. *Archives of Neurology*, 41(5):485–490, Jan 1984.

[7] R. Brown and C. Marsden. Visuospatial function in parkinson's disease. *Brain*, 109(5):987–1002, Jan 1986.

[8] D. Calne. Treatment of parkinson's disease. *New England Journal of Medicine*, 330(9):643–644, March 1994.

[9] A. Cockburn and P. Brock. Human on-line response to visual and motor target expansion. In *GI '06: Proceedings of Graphics Interface 2006*, pages 81–87, Toronto, Ont., Canada, Canada, Jan 2006. Canadian Information Processing Society.

[10] J. Contreras-Vidal and E. Buch. Effects of parkinson's disease on visuomotor adaptation. *Experimental Brain Research*, 150(1):25–32, May 2003.

[11] C. Crook. Young children's skill in using a mouse to control a graphical user interface. *Computers and Education*, 19(3):199–207, Jan 1992.

[12] L. Cunningham, C. Nugent, and D. Finlay. A review of assistive technologies for people with parkinson's disease. *Technology and Health Care*, 17(3):269–279, Jan 2009.

[13] E. Edgington. Statistical inference from n=1 experiments. *Journal of Psychology*, 65(2):195–199, 1967.

[14] E. Edgington. Randomized single-subject experimental designs. *Behaviour Research and Therapy*, 34(7):567–574, 1996.

[15] A Fisk, W Rogers, N Charness, and S Czaja. *Designing for older adults: Principles and creative human factors approaches*. CRC, 2009.

[16] K. Gajos, A. Hurst, and L. Findlater. Personalized dynamic accessibility. *Interactions*, 19(2):69–73, 2012.

[17] K. Gajos, J. Wobbrock, and D. Weld. Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1257–1266, New York, NY, USA, 2008. ACM.

[18] L. Gitlin and R. Schemm. Maximizing assistive device use among older users. *TeamRehab*, 25:25–26, 28, Apr 1996.

[19] T. Grossman and R. Balakrishnan. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–290, New York, NY, USA, Jan 2005. ACM.

[20] Y. Guiard, R. Blanch, and M. Beaudouin-Lafon. Object pointing: a complement to bitmap pointing in guis. In *GI '04: Proceedings of Graphics Interface 2004*, pages 9–16, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.

[21] S. Heim. *The Resonant Interface: HCI foundations for interaction design*. Addison Wesley, 1 edition, March 2007.

[22] J.P. Hourcade. Learning from preschool children's pointing sub-movements. In *IDC '06: Proceedings of the 2006 conference on Interaction design and children*, pages 65–72, Jan 2006.

[23] J.P. Hourcade. Pointassist for older adults: Analyzing sub-movement characteristics to aid in pointing tasks. In *CHI '09: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, 2009. ACM Press.

[24] J.P. Hourcade, B. Bederson, A. Druin, and F. Guimbretiere. Differences in pointing task performance between preschool children and adults using mice. *ACM Trans. Comput.-Hum. Interact.*, 11(4):357–386, Jan 2004.

[25] J.P. Hourcade, K. Perry, and A. Sharma. Pointassist: helping four year olds point with ease. In *IDC '08: Proceedings of the 7th international conference on Interaction design and children*, pages 202–209, New York, NY, USA, Jan 2008. ACM.

[26] A. Hurst, S. Hudson, and J. Mankoff. Dynamic detection of novice vs. skilled use without a task model. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 271–280, Jan 2007.

[27] A. Hurst, S. Hudson, J. Mankoff, and S. Trewin. Automatically detecting pointing performance. In *IUI '08: Proceedings of the 13th international conference on Intelligent user interfaces*, pages 11–19, Jan 2008.

[28] A. Hurst, J. Mankoff, and S. Hudson. Understanding pointing problems in real world computing environments. In *Assets '08: Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, Jan 2008.

[29] F. Hwang, H. Batson, and N. Williams. Bringing the target to the cursor: proxy targets for older adults. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 2775–2780, New York, NY, USA, 2008. ACM.

[30] F. Hwang, S. Keates, P. Langdon, and J. Clarkson. Mouse movements of motion-impaired users: a submovement analysis. *SIGACCESS Access. Comput.*, (77-78):102–109, 2004.

[31] I. Scott I. MacKenzie, T. Kauppinen, and M. Silfverberg. Accuracy measures for evaluating computer pointing devices. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 9–16, New York, NY, USA, Jan 2001. ACM Press.

[32] N. Jordan, H. Sagar, and J. Cooper. Cognitive components of reaction time in parkinson's disease. *Journal of Neurology, Neurosurgery and Psychiatry with Practical Neurology*, 55(8):658–664, Aug 1992.

[33] P. Kabbash and W. Buxton. The "prince" technique: Fitts' law and selection

using area cursors. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 273–279, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[34] S. Keates and S. Trewin. Effect of age and parkinson's disease on cursor positioning using a mouse. In *Assets '05: Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*, pages 68–75, New York, NY, USA, Jan 2005. ACM Press.

[35] W. König, J. Gerken, S. Dierdorf, and H. Reiterer. Adaptive pointing: implicit gain adaptation for absolute pointing devices. In *CHI EA '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pages 4171–4176, New York, NY, USA, Jan 2009. ACM Press.

[36] B. Lowe. Effect of carpal tunnel syndrome on grip force coordination on hand tools. *Ergonomics*, 42(4):550–564, Jan 1999.

[37] I. MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7(1):91–139, Jan 1992.

[38] M. McGuffin and R. Balakrishnan. Acquisition of expanding targets. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 57–64, New York, NY, USA, Jan 2002. ACM Press.

[39] D.E. Meyer, R.A. Abrams, S. Kornblum, C.E. Wright, and J.E. Smith. Optimality in human motor performance: Ideal control of rapid aimed movements. *Psychological Review*, 95(3):340–370, Jan 1988.

[40] Microsoft. Creating windows xp icons, July 2001.

[41] Microsoft. Pointer ballistics for windows xp, October 2002.

[42] J. Muras, E. Stokes, and V. Cahill. Assistive technology in everyday living–a user survey of people with parkinson's disease. *Technology and Disability*, 20(4):271–282, Jan 2008.

[43] NINDS. Tremor fact sheet, June 2006.

[44] P. Onghena. Randomization tests for extensions and variations of abab single-case experimental designs: A rejoinder. *Behavioral Assessment*, 14(2):153–171, 1992.

[45] W. Poewe. The natural history of parkinson's disease. *Neurology*, 47:146S–152S, Jan 1996.

[46] S. Rao, L. Hofmann, and A. Shakil. Parkinson's disease: diagnosis and treatment. *American Family Physician*, 74(12):2046–2054, Jan 2006.

[47] L. Samuelson and L. Smith. Grounding development in cognitive processes. *Child Development*, 71(1):98–106, Jan 2000.

[48] C. Shih, M. Chang, and C. Shih. Assisting people with multiple disabilities and minimal motor behavior to improve computer pointing efficiency through a mouse wheel. *Research in Developmental Disabilities*, 30(6):1378–1387, Jan 2009.

[49] C. Shih, N. Hsu, and C. Shih. Assisting people with developmental disabilities to improve pointing efficiency with an automatic pointing assistive program. *Research in Developmental Disabilities*, 30(6):1212 – 1220, 2009.

[50] H.L. Teulings, J.L. Contreras-Vidal, G.E. Stelmach, and C.H. Adler. Adaptation of handwriting size under distorted visual feedback in patients with parkinson's disease and elderly and young controls. *Journal of Neurology, Neurosurgery and Psychiatry with Practical Neurology*, 72(3):315–324, 2002.

[51] E. Tolosa, G. Wenning, and W. Poewe. The diagnosis of parkinson's disease. *Lancet Neurology*, 5(1):75–86, Jan 2006.

[52] S. Trewin. *Physical Impairment*. Springer London, Jan 2008.

[53] S. Trewin, S. Keates, and K. Moffatt. Developing steady clicks:: a method of cursor assistance for people with motor impairments. In *Assets '06: Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*, pages 26–33, New York, NY, USA, Jan 2006. ACM Press.

[54] S. Trewin, S. Keates, and K. Moffatt. Individual responses to a method of cursor assistance. *Disability and Rehabilitation: Assistive Technology*, 3(1):2–21, Jan 2008.

[55] K. van Mensvoort. Powercursor, 2009.

[56] J. Wobbrock, J. Fogarty, S. Liu, S. Kimuro, and S. Harada. The angle mouse: target-agnostic dynamic gain adjustment based on angular deviation. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 1401–1410, New York, NY, USA, Jan 2009. ACM Press.

[57] A. Worden, N. Walker, K. Bharat, and S. Hudson. Making computers easier for older adults to use: area cursors and sticky icons. In *CHI '97: Proceedings of*

the SIGCHI conference on Human factors in computing systems, pages 266–271, New York, NY, USA, Jan 1997. ACM Press.