Multi-Scale Cursor: Optimizing Mouse Interaction for Large Personal Workspaces

Mehmet Celal Dasiyici

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Dr. Christopher L. North, Chair
Dr. Doug A. Bowman
Dr. Francis Quek

May 8, 2008
Blacksburg, Virginia

Keywords: Pointing, large displays, Fitts' law, interaction techniques, mouse cursor, interaction styles, multiple monitors, large displays, high-resolution, large tiled display, user interface design, user study, multi scale cursor.

Multi-Scale Cursor: Optimizing Mouse Interaction for Large Personal Workspaces

Mehmet Celal Dasiyici

Abstract

As increasingly large displays are integrated into personal workspaces, mouse-based interaction becomes more problematic. Users must repeatedly "clutch" the mouse for long distance movements [61]. The visibility of the cursor is also problematic in large screens, since the percentage of the screen space that the cursor takes from the whole display gets smaller. We test multi-scale approaches to mouse interaction that utilize dynamic speed and size techniques to grow the cursor larger and faster for long movements. Using Fitts' Law methods, we experimentally compare different implementations to optimize the mouse design for large displays and to test how they scale to large displays. We also compare them to techniques that integrate absolute pointing with head tracking. Results indicate that with some implementation level modifications the mouse device can scale well up to even a 100 megapixel display with lower mean movement times as compared to integrating absolute pointing techniques to mouse input while maintaining fast performance of the typical mouse configuration on small screens for short distance movements. Designs that have multiple acceleration levels and 4x maximum acceleration reduced average number of clutching to less than one per task in a 100 megapixel display. Dynamic size cursors statistically improve pointing performance. Results also indicated that dynamic speed transitions should be as smooth as possible without steps of more than 2x increase in speed.

Acknowledgments

I would like thank my advisor Dr. Chris North for the motivation, time and guidance he has given me throughout my thesis. I would also like to thank Dr. Doug Bowman and Dr. Francis Quek for serving on my committee and helping me whenever I needed.

I am very thankful to my lab-mates Christopher Andrews, Robert Ball, Sarah Peck, Beth Yost and Michael DellaNoce for whose help were invaluable. I would also want to thank Firat Alpergin, Regis Kooper, Francisco Oliveira, Filip Blagojevic, Mithilesh Kumar, Jon Howarth, Pardha Pyla, everyone else in the Center for Human Computer Interaction and all of the participants of these studies. Very special thanks to Yonca Haciahmetoglu both as a friend and as a lab-mate for her precious help and infinite patience over my never ending questions.

I have been fortunate enough to be surrounded by great friends through out this process. I want to thank all members of the Turkish Student Association, the "Brazilians", Cranwell International Center stuff and all my friends for making Blacksburg an enjoyable town and for supporting me during my difficult days. You were the main reason why I loved to live in this town and it was a pleasure to meet each one of you.

I would like to thank my family, to whom I owe the greatest debt. I would like to thank my immediate and extended family in Turkey for their priceless love and support. I thank my brother-in-laws Ertan Ozekin and Oner Dikensoy for always treating me as a brother, supporting me and my decisions no matter what. My gratitude goes to my sisters Arzu Ozekin and Ebru Dikensoy for always believing in me, being constantly supportive and being there all the time whenever I needed them. I am proud to be "your brother" and I am lucky to have both of you in my life. Above all, I thank my mother Meral Dasiyici and my father Ibrahim Halil Dasiyici. Without them I would be nothing, without their sacrifices, support and encouragement I would not be in the position that I reside today. I can not thank my parents enough for all these and for their unconditional love, but it is my sincerest hope that I would repay them at least by deserving to be "their son".

Dedicated to the memory of my father, Ibrahim Halil Dasiyici

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1 Motivation

1.1.1. The Scenario

As display technologies continue to improve, future personal workspaces can include large, high-resolution displays containing very large numbers of pixels (Figure 1.1) to facilitate several tasks. Some high-end office workers already have workspaces containing over 30 megapixels while using common operating systems and applications. This upward trend in display sizes is likely to continue, since display technology cost is decreasing [30] and recent evidence indicates significant advantages of larger displays for personal workspaces (e.g. [4, 22, 64]).

Figure 1.1: Large personal workspaces of various sizes. (a) 13 megapixel cubicle. (b) 32 megapixel office. (c) 100 megapixel workspace used in the experiments to ensure scalability.

Large, high resolution displays are particularly beneficial for people who are dealing with massive data sets that involve spatial and/or multidimensional data and thus making heavy use of information visualizations (e.g. [1, 65, 77]). As an illustration, multidimensional complex data is central to the studies of geospatial intelligence analysts [3], epidemiologists [45], and biologists [66]. Thus, they need to make heavy use of information visualizations. Information visualizations are mostly limited by the number of pixels [23], thus the scalability of information visualizations can be improved by making use of large, high resolution displays. In theory, a dataset in any size can be visualized, if the number of pixels to be used in the visualization is unlimited. Hence, the usage of large, high resolution* displays in single user workspaces offer advantages in increasing the ability to gain insight from a dataset. Indeed, studies indicate that large, high resolution displays help to significantly improve human performance on several tasks such as in basic perception and navigation [5] and in multi tasking (e.g. [4, 22]).

Our main focus is on a scenario in which the workspace is intended for an individual user in a home or office for productivity tasks, gaming, digital media management or data analytics tasks where heavy usage of visualizations is necessary. The workspace we consider, consists of a large high-resolution display screen (currently tiled LCD monitors). Ideally, the screen is curved around the user so that most of the display space can be easily accessed with minimal movement by a seated user, perhaps by rotating the chair and the keyboard tray.

1.1.2. The Problem

Despite advantages of large, high resolution displays on human performance in several tasks, a problem with such large workspaces is the interaction technique for pointing. The current implementation of the mouse interaction technique is problematic with large displays [56]. On the other hand, other techniques, such as usage of joystick, text keys, usage of head tracking or eye tracking input frequently have difficulty surpassing the mouse in terms of both accuracy and performance as can be seen in the results of [3, 44, 81, 35, 17]. Moreover, mouse has other advantages over alternative interaction techniques such as providing a lazy usage behavior by requiring little overhead to acquire

* The word “resolution” generally refers to the density of pixels on the screen [2S], but we use “resolution” to refer to the number of pixels on a display.

the device and to move the device.

The primary problem with using the mouse device in large, high resolution displays involves trying to move the cursor long distances. When the cursor is moved faster to facilitate long distance movements, target acquisition becomes a problem since it gets harder to stop on top of a target. While maximizing target acquisition, as the cursor is moved slower, users are forced to repeatedly “clutch” the mouse by lifting it off the surface. This problem, commonly referred to as clutching, is a tedious operation and might cause worse performance (e.g. [63]).

A related problem is the lack of visibility of the mouse cursor during such movements as the ratio of the screen space that the cursor takes from the whole display gets smaller in a large, high resolution display. Losing the cursor while moving it or while trying to find the cursor before starting to move it are general problems related to cursor visibility in large, high resolution displays. The cursor visibility problem is another reason for the poor pointing performance of the mouse device in large displays.

A straight forward solution to the clutching problem in large displays[†] is speeding up the cursor. Speeding up the cursor makes it harder to acquire small targets. Also, when the cursor goes faster than normal, it gets harder to detect and follow the cursor in a large, high resolution display where the cursor already takes less percentage of space from the whole display. Using a bigger cursor in size is a straight forward solution for the cursor visibility problem of the mouse device, but that would cause target occlusion problems.

These problems decrease the usability of the mouse device in large displays and therefore limit the scalability of the mouse device to larger displays. We investigate these problems through out the thesis to question whether we can scale up the mouse to large, high resolution displays.

Furthermore, other studies of large display interaction techniques typically compare new interaction designs with the current mouse implementation (e.g. [3, 8, 81]). This is problematic, since current mouse implementation has usability problems in large, high

[†] Large displays may refer to large, low-resolution displays like projector based displays or large, high-resolution displays like multi-monitor displays. We use the large displays term to refer to the large, high-resolution displays through out the thesis.

resolution displays. Thus, we would like to provide a good design to be used in such comparison studies to be fair to the usage of the mouse device for benchmarking purposes.

1.2 Potential Solution: Multi-Scale Cursor (MSC)

By dynamically adapting the control-display (C-D) ratio according to the device movement speed (e.g. [73]), Multi-Scale Cursor (MSC) concept basically grows or shrinks the cursor size proportional to the cursor speed. We tested the dynamic C-D ratio adaptation, dynamically resizing the cursor ideas to scale up the mouse device to large, high resolution displays. These ideas are used to decrease the pointing amplitude to facilitate faster pointing. We also tested usage of head tracking input integrated with the mouse input to facilitate pointing and compared this idea with the MSC concept.

Figure 1.2: The Multi-Scale Cursor (MSC) cursor moves faster and grows bigger for long distance movement, then slows down and gets smaller for fine detailed control.

Fitts' Law indicates that it is faster to acquire closer targets as compared to distant targets and it is faster to acquire bigger targets as compared to smaller ones. Thus, decreasing the pointing amplitude and increasing the target width might be possible alternatives for improving the mouse device's performance on large, high resolution screens.

The C-D ratio adaptation and the usage of head tracking input decreases the pointing amplitude in the motor and visual space. C-D ratio is the ratio between the amount of movement in a device in motor space to the amount of movement in the effected displayed object in visual space. Adapting the C-D ratio to facilitate pointing is also being used in modern operating systems such as Windows XP [52]. We conducted two studies using these modifications to investigate the scalability of the mouse device to large displays. We will investigate the solution space in detail in the chapter 3.

1.3 Research Questions

The problem of lacking adequate input techniques for large, high resolution displays leads to several questions that we will investigate through out the thesis. Different devices like joystick, trackball, laser pointer, etc. might be possible alternatives for interacting with large, high resolution displays. However, other devices have difficulty surpassing the mouse device in terms of both accuracy and performance. (e.g. [17, 26]). Also, since the mouse is often used as a baseline for comparison against other innovative large-display interaction techniques, it is important to provide a good mouse design for comparison. Therefore, in our studies we questioned how we can improve the scalability of the mouse device to large, high resolution displays.

The scalability of a mouse design is the effect of the screen size on pointing performance. Pointing time, number of mouse clutches and number of error clicks are all used to determine the pointing performance of a mouse design. As implied by Fitts' Law [47] the pointing performance of a mouse design changes according to the target distance. As the target distance gets longer the pointing performance of each mouse design deteriorates. One of the main factors that affect scalability of a mouse design is the deterioration rate in pointing performance as the target distance gets longer. In a given screen size there might be long target distances as well as short and medium ones. Therefore, if a mouse design can perform well on short distances and does not deteriorate much as the target distance gets longer, then that mouse design is said to scale well to that given screen size.

We investigated the scalability of the mouse device to large, high resolution displays by answering the following questions throughout the thesis:

How can we improve the scalability of mouse interaction to large displays? Can we redesign or further optimize the mouse interaction to be more scalable to large displays? Can we reduce or eliminate clutching and cursor visibility problems? Can it perform well at accurately pointing to small targets while also facilitating large cursor movements from one side of the screen to the other without any additional clutching action?

The specific sub-questions related to the above research questions are:

1. Can the dynamic size behavior of the cursor lead to better scalability of the mouse device to large displays?
2. Can the dynamic speed, (i.e. dynamic control-display (C-D) ratio adaptation) help to improve the scalability of the mouse device to large displays? What are the best design choices (i.e. number of thresholds, speed values, how to switch speeds) for adapting C-D ratio?
3. How can we further scale the mouse device beyond the performance of the MSC concept? How would integrating absolute pointing techniques and the MSC concept affect the scalability of the mouse device in large screens?

The corresponding hypotheses are:

1. Dynamic size behavior of the MSC helps to improve user performance in detecting and following the cursor and in acquiring targets by providing better visual feedback about the cursor without causing any additional problems like the target occlusion problem. Thus dynamically resizing the cursor according to the cursor speed helps to improve the scalability of the mouse device to large displays.
2. As the dynamic C-D ratio adaptation can ease pointing to distant targets, it can statistically improve user performance and thus improve the scalability of the mouse device to large displays when the threshold and ratio values are selected correctly. Having more threshold levels would result in smaller mean movement times and less number of mouse clutches. Because, decreasing the ratio differences between C-D ratio levels, while maintaining the same top and bottom ratio levels would provide better fine control and ease the ability to control the cursor movement on the screen. The gap between each C-D ratio level should not be large.
3. Integrating absolute pointing techniques and the MSC concept would result in smaller mean movement times and less number of clutches than using only the MSC concept.

1.4 Significance

This study helps interaction designers to better understand the scalability limits of the mouse device into large, high resolution displays. The study also helps interaction designers to better understand different design choices, while trying to improve the mouse implementation as we move from the small desktop screen to large, high resolution displays.

The study also helps users to interact with large displays more effectively. The MSC concept decreased the mean movement time around 1 second as compared to the control cursor which can be seen in sections 4.4 and 5.4. While using a large screen, if mean movement time decreases 1 second for each target acquisition the user will gain a lot of precious time in one day and will have a chance to be more productive.

In an industrial point of view, the techniques purposed in this study (MSC) do not require any new equipment. It also does not require any additional learning effort from users. Therefore the techniques purposed in this study are low-cost. The MSC concept is also easy to implement and can be patched to the current operating systems without much effort.

1.5 Overview of Thesis

This thesis is organized as follows: Chapter 2 introduces rich literature about interaction techniques and specifically pointing facilitation techniques to improve the performance of mouse interaction. Chapter 3 investigates the solution space, explains the scalability framework used in the thesis and also explains the design details of the MSC concept. Chapter 4 explains details of the first experiment conducted and its results. Chapter 5 explains details of the second experiment conducted and its results. Finally, Chapter 6 discusses the results of the experiments, explains concluding remarks, describes future work and points out specific contributions.

Chapter 2. Literature Review

2.1 Evaluations of Large Displays

Before describing novel interaction techniques, we need a better understanding of the rising importance of large, high-resolution displays. A significant amount of research has been conducted on evaluation of large, high-resolution displays for multiple users (e.g. [25, 41, 70]). Multiple users benefit from large displays as they provide more space for physical navigation, such as body and head movements, which allow users to utilize their motor skills and peripheral visions. However, in the light of the scenario under investigation, improving large display interaction techniques for a single user is our main goal. Before discussing the ways of improving single user large display interaction techniques, it is necessary to understand the benefits brought by larger displays and an increased number of pixels.

Large, high-resolution displays allow users to utilize their motor skills and peripheral vision, as they are enabled being able to physically navigate, to form a cognitive map [6]. In a previous study, where subjects were asked to route trace in a given map, their performances were compared in three different display size settings. The study showed an improvement in route tracing performance, a decrease in the number of mouse clicks and a decrease in the amount of window management as the screen size gets larger [7]. Large, high-resolution displays also resulted in better performance in tracing routes in a given map than panning and zooming in smaller displays [5, 7]. Sabri, et al. [64] compared scores, number of wins and losses and the time spent panning the map in a game in three different display sizes (one screen, three screens and nine screens). The study showed that screen size has a statistically significant effect on number of wins, game scores and time spent for navigation. As the size of the screen increases, participants tended to win more, gained higher game scores and spent less time on navigation.

Research in this area has also shown that large displays improve human performance on several tasks such as better user performance in 3D virtual navigation [74], multi tasking [4, 22, 69], basic navigation [5], spatial performance even though the visual angle is maintained [75], and an increase in memory [76]. Some of these benefits are caused by increased field of view and some of them caused by increased dots per inch (DPI). Effects of the display configuration on user performance have also been studied by Shupp, et al. [67]. The study showed that curved configuration of the large display improves human performance as compared to the flat version since it decreases the physical navigation required by bringing the distant targets closer to the user.

Large, high resolution displays have experimentally proven significant benefits for humans in several tasks as explained above. Therefore, using large screens in an office setting like in our scenario would benefit users. Thus, the question arises of finding the most efficient way of interacting with those large high-resolution screens.

2.2 Alternatives for Interacting with Large Displays

In order to fully take advantage of several cognitive and performance benefits for single users brought by large, high resolution displays (e.g. [4, 21, 22]), it is necessary to solve large screen interaction problems. The mouse, currently the most common interaction technique, has usability problems like cursor visibility and clutching in large high-resolution screens [56]. Thus, several innovative interaction techniques for large displays have been developed as alternatives to the current mouse implementation. These alternatives include using head-tracking input (e.g. [3, 81]), eye tracking input (e.g. [44, 68]), infrared laser tracking input [18], pen based interaction (e.g. [37, 62]), hybrid pointing [28] and laser pointers (e.g. [54, 60]).

In the scenario under investigation, users do not need much mobility and generally require a keyboard in an office setting. As the mouse device possesses high level of accuracy in pointing at any pixel on the screen, it has a relative advantage over pointing with a finger, wand, laser pointer or touching the screen. The mouse device also causes less fatigue than directly pointing to a pixel or touching it. Consequently, the mouse is a reasonable alternative that must be investigated further. In fact, in this scenario, other techniques frequently have difficulty surpassing the mouse in terms of both accuracy and performance as can be seen in the results of [3, 35, 44, 81]. On the other hand, the standard implementation of the mouse is problematic with large, high resolution displays due to the difficulty of reaching distant targets, which causes higher frequency of clutching, and cursor visibility problems [63]. We investigate those usability problems of the current mouse implementation to test its scalability to large high-resolution displays.

2.3 Mouse Techniques

In large high-resolution displays, the current implementation of the mouse device is not as effectively usable as it is in smaller screens [56]. Reaching distant targets, tracking/detecting the cursor, and crossing bezels are known problems for the mouse usage in large screens. As the mouse is the main interaction technique for a single user in an office setting with both large displays and single displays, several innovative techniques have been developed to improve its usability.

In the studies, where several innovative techniques are developed to improve the mouse usability (e.g. [51, 13, 32]), Fitts' Law is largely used as a tool to compare different designs/techniques in the Human-Computer Interaction (HCI) field [47]. Fitts' Law is an important law in the HCI field, since it is being used as a tool to compare performances of different designs and to predict performances of each interaction technique. To compare different interaction techniques in our studies we also used Fitts' Law pointing tasks, as it is commonly used as a research tool in HCI studies [47].

More importantly, Fitts' Law indicates different ways in which pointing can be facilitated. In the following sections, we will discuss Fitts' Law and how the law and its implications have been used in facilitating pointing. We will specifically concentrate on how the Fitts' Law and its implications have been used in increasing the performance and usability of the mouse device.

2.3.1 Fitts' Law and Its Applications in HCI

Fitts' Law states that it is easier to point to larger, closer targets than pointing to smaller, distant targets. It explains the relationship between movement time (MT) necessary to point to a target and the index of difficulty (ID) of that task. MT is a linear function of ID, where ID is the logarithm of the ratio of target distance (A) to target width (W).

$$MT = a + b \times ID \text{ [47]} \quad (1)$$

$$ID = \log_2(A/W + C) \text{ [47]} \quad (2)$$

In the MT equation (Equation 1) a and b are empirical constants. ID is in bits and C is a constant which takes the values 0, 1/2 or 1. C = 1 is used in the Shannon formulation of Fitts' Law and it keeps ID always positive and fits measured data better [46, 47]. That is the reason for using C = 1 in our analysis.

The constants a and b in the MT equation (Equation 1) depend on the factors such as the pointing device (e.g. mouse, trackball, joystick), the user age rank, and the control display (C-D) ratio, which is the ratio of the distance that the device moved to the distance that the pointer moved on the screen. Several studies have been conducted to facilitate pointing by adapting C-D ratio (e.g. [16, 29, 48]) and by seeking better input techniques (e.g. [17]).

The constants a and b are defined as the slope and intercept of the line that fits to the measured MT versus given ID values. Normally, those measured MT values are gathered from user performances for many selections where A and W are varied. After determining a and b values, Fitts' Law enables the prediction of future pointing performances when the factors which affect a and b values stay the same. Moreover Fitts' Law also enables the characterization of input devices by determining their Index of Performance (IP). IP is the number of bits the device can transmit in a second, regardless of the particular target size or the target distance. IP is calculated as IP = 1/b [79] and also called "throughput" in the literature.

As can be seen from Equations (1), (2), pointing performance can be improved by decreasing pointing amplitude, increasing target size, and adapting C-D ratio, which would decrease movement time (MT) in Equation (1). As the pointing amplitude gets larger, it gets harder to acquire a target, since mean movement time (MT in Equation (1)) is directly proportional to the pointing amplitude. Also, pointing performance is inversely proportional to the target width, as it is easier to acquire bigger targets than smaller ones. Moreover, the constants in Equation (1) can be affected by C-D ratio. Thus, according to Fitts' Law facilitating pointing can be accomplished in several alternative ways, mainly by:

- increasing target width,
- decreasing pointing amplitude,
- adapting C-D ratio/gain.

We will investigate these three categories in the following subsections.

2.3.1.1 Facilitating Pointing Through Increasing the Target Width

Fitts' Law indicates that as target size increases, it becomes easier to point to that target. Therefore, changing the target size to facilitate pointing is largely investigated in the literature. Studies about changing target size to facilitate pointing can be classified into two groups, enlarging the cursor size and enlarging target size.

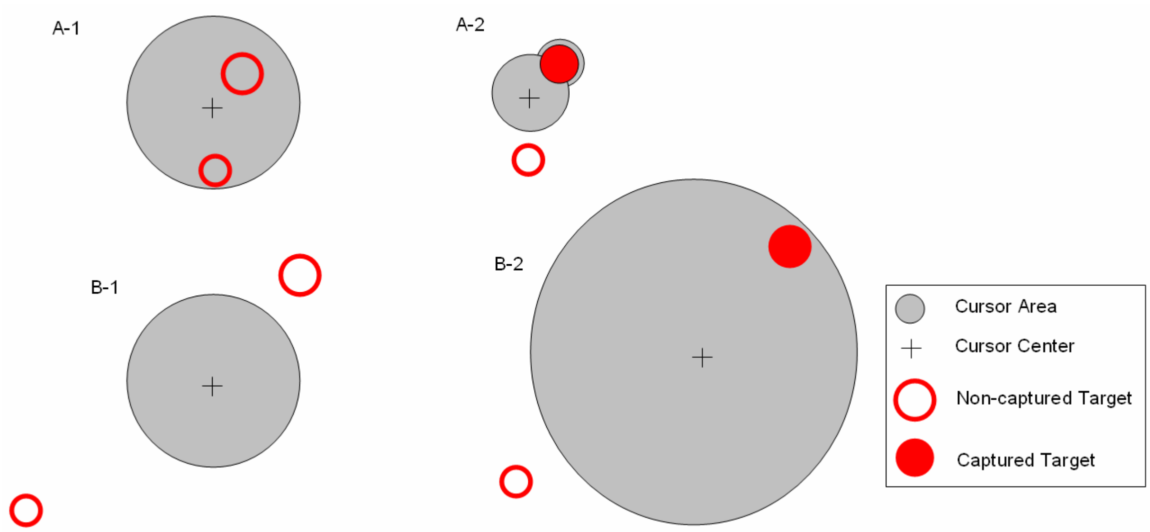


Figure 2.1: Figure depicts the bubble cursor behavior on two different usage scenarios (A and B). Bubble Cursor [32] skips empty space by pointing to the closest target all the time.

Enlarging the cursor size to be able to facilitate pointing has been investigated in the same sense as enlarging target size, but the difference is in visual feedback. Designs that enlarge the cursor size to facilitate pointing put more stress on cursor size than the target size. Area cursors (e.g. [42, 78]) and the Bubble Cursor [32] enlarge the cursor size and skips empty spaces to facilitate pointing. The Bubble Cursor [32] dynamically changes cursor size and shape to be able to jump between targets to skip empty space so as to be able to point to the nearest target all the time.

The second part of facilitating pointing via changing target width is expanding the target size. By using the information gathered from the cursor movement trajectory, targets can be dynamically expanded to facilitate pointing. Effects of different types of target expansions on pointing performance [50, 51] and human response to target expansion [80] have also been investigated in the literature.

Target and cursor size expansion require both the knowledge of target positions and cursor movement trajectory. However, to support the scalability of the number of potential targets without any information about the target places on the screen or more complex spatial tasks such as image editing, we focus on the general case of pointing to any pixel on the display. Also, our scenario under investigation requires a general solution that not only supports desktop level pointing but also supports application level pointing. In that case, knowing the position of each target and processing those positions each time the user moves the cursor would be extremely expensive and can not be a viable option.

2.3.1.2 Facilitating Pointing Through Decreasing the Pointing Amplitude

Another straightforward implication of Fitts' Law is that as the distance to a target (pointing amplitude) increases, it becomes harder to acquire that target. Thus, several techniques have been developed to decrease the distance traveled to acquire targets. Studies addressing pointing amplitude to ease pointing can be investigated in threefold.

First, pointing amplitude can be decreased by moving the cursor closer to a target or easing movement towards a target. Object Pointing [34] is an example of techniques that decrease pointing amplitude to ease reaching targets. This approach allows the cursor to always point to a target by guessing cursor movement direction and skipping empty spaces. To address the same problem, there are several other techniques which try to facilitate pointing by changing pointing amplitude like displacing the cursor onto the closest target without truly warping the cursor (e.g. [12, 71]) or using the help of the haptic feedback (e.g. [53, 58, 59]).

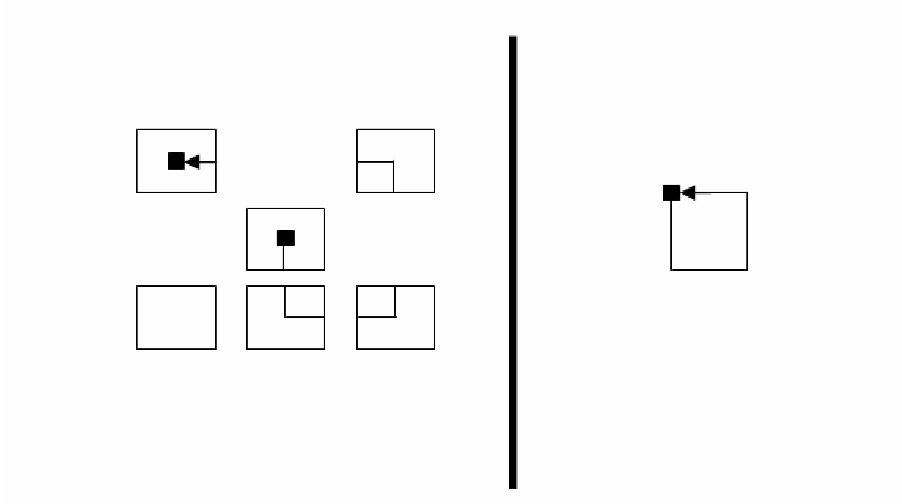


Figure 2.2: Object Pointing [34] decreases pointing amplitude by skipping empty spaces. Left of the diagram shows movement in the motor space, while right of the diagram shows cursor movement in the visual space.

Second, pointing amplitude can be decreased by moving targets closer to the cursor. The most relevant example for this category is Baudisch, et al.'s Drag-and-Pick [8] technique. Drag-and-Pick temporarily brings potential targets closer to the cursor for quick selection to facilitate faster pointing.

Decreasing pointing amplitude to facilitate pointing involves moving either the target closer to the cursor or the cursor to a target, thus they require the knowledge of target positions and processing those positions in each mouse move event. Our scenario under investigation requires a general solution to support scalability of the number of potential targets on the display for both desktop and application level pointing. Therefore, knowing the position of each target and processing those positions each time the user moves the cursor would not be a reasonable solution. Thus, we prefer to address adapting the control display (C-D) ratio, which in a sense also changes the pointing amplitude to facilitate pointing. It can be used to decrease the distance that the device must be moved to acquire a distant target.

The third category of techniques is the usage of absolute pointing to facilitate pointing. It can be done by using head tracking, eye tracking, hand, wand, gesture input as well as by combining various absolute pointing techniques with mouse usage. Absolute pointing techniques (e.g. [44]) and their combination with mouse usage (e.g. [2, 81]) enables users to instantly warp the cursor to a distant location in order to decrease the pointing amplitude. Warping the cursor to a distant location to facilitate pointing has been examined on a single monitor either manually [24] or automatically by combining gaze and mouse input [81], and on multiple monitors by combining head tracking and mouse input [3, 11]. In our second experiment, we integrate some of these strategies with the multi-scale cursor (MSC) approach and compare them with the MSC-only approach.

2.3.1.3 Facilitating Pointing Through Adapting the Control Display (C-D) Ratio

Control display (C-D) ratio affects movement time by affecting a (intercept) and b (slope) values in the Fitts' Law curve formed by the Fitts' Law Equation (1) [51]. Effective adaptation of C-D ratio has been studied to facilitate pointing. Control display (C-D) ratio is the ratio between the amount of movement in a device (e.g. mouse) in motor space to the amount of motion in the affected displayed object (e.g. cursor) in visual space. If C-D ratio is constant then there is a linear mapping from motor space to the visual space. Mapping between motor space and visual space is defined as "C-D gain" [29] or "C-D ratio" [49] in the literature. We will use the terminology C-D ratio throughout the paper.

As a and b in Equation (1) are affected by the C-D ratio, there have been several studies conducted to investigate its effects. In a linear C-D mapping, Gibbs [29] investigated effects of C-D ratio on pointing performance and found that as the ratio increases, movement time also monotonically increases. This means that a higher C-D ratio results in worse performance in linear C-D mapping. In contrast, Jenkins and Connor [39] found a U-shaped performance-gain function in a linear C-D mapping. In a U-shaped performance-gain function, when the gain (C-D ratio) is medium, the performance peaks, and when the gain is close to the minimum or the maximum, the performance decreases and gets closer to the minimum. In terms of the C-D ratio adaptation effect on user performance, many studies showed an effect of the C-D ratio adaptation on user performance (e.g. [48]), multiple studies found a U-shaped gain function (e.g. [2, 36]), several studies found straight linear gain function (e.g. [29]), multiple researchers did not believe the performance gain from the C-D ratio adaptation (e.g. [14, 38]) and some studies showed a performance improvement by the C-D ratio adaptation (e.g. [20, 43, 78]).

Even though there is no consensus on whether the C-D ratio adaptation provides benefits or not, several techniques have been developed using C-D ratio adaptation to facilitate pointing. Semantic Pointing [13] changes C-D ratio according to the distance between the target and the cursor to ease pointing, which results in slowing down the cursor speed when it is closer to a target and increasing its speed when there is no target around. Area cursors (e.g. [42, 55]) change C-D ratio according to the cursor position to facilitate pointing in the same way. However, these techniques again require the knowledge of positions of targets beforehand and to process them each time the user moves the controller device.

Techniques using the C-D ratio adaptation without requiring any knowledge of target position have also been developed. Trankle used a dynamic C-D ratio adaptation related to the speed of mouse movement [73]. The main idea behind this technique, so called "acceleration", is currently being used in modern operating systems such as in Windows XP [52].

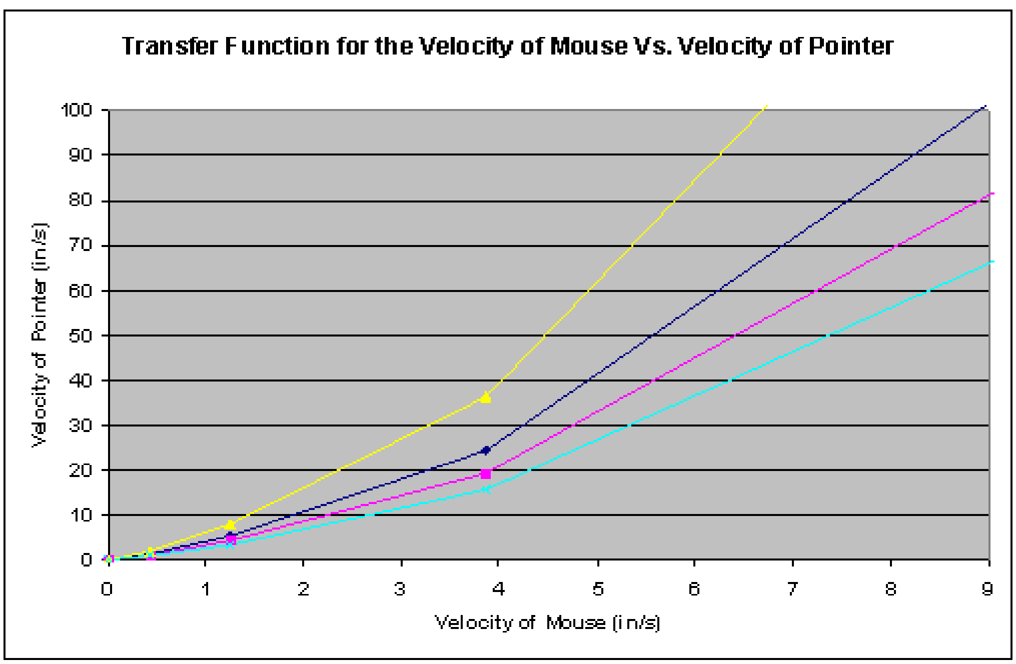


Figure 2.3: Windows XP facilitates faster pointing by using dynamic C-D ratio adaptation. Graph explains Windows XP's C-D transfer function.

In Trankle's technique, when users move the controller device faster, the cursor moves more than it should. C-D ratio dynamically increases or decreases according to the controller device movement speed. In the case of large displays, non-linear C-D mappings is said to be too counterintuitive to be a general solution for pointer movement [72]. Contrarily, non-linear C-D mappings were successfully applied and studies showed performance improvement with C-D ratio adaptation (e.g. [20, 43, 78]).

We used the idea of dynamically adapting C-D ratio according to the device movement speed [73], to test the scalability of the mouse device to large displays.

2.3.2 Mouse Techniques for Large, High-Resolution Displays

After reviewing general pointing facilitation techniques for the mouse device, it is necessary to investigate the literature about mouse interaction's usability problems in large displays and different mouse techniques developed for large, high-resolution displays in the light of our scenario under investigation.

In large, high-resolution displays the mouse has two main problems. As the screen size gets larger, reaching distant targets causes the problem referred to as the clutching problem, which is the action of moving the mouse device off the surface several times in order to move the cursor to distant targets in large displays. Clutching is the most obvious usability problem of the mouse device in large, high resolution displays. Different techniques investigated in the section 2.3.1.2, regarding facilitating pointing through decreasing the pointing amplitude, would be possible solutions for the clutching problem. But most of those techniques like Drag-and-Pick [8] and Object Pointing [34] require the knowledge of target positions and processing those positions in each mouse move. Therefore this would not be a plausible approach in the scenario under investigation, since it does not support the scalability of the number of potential targets in both application and desktop levels.

There are other techniques which try to facilitate reaching distant targets in large screens without any knowledge about target positions. The Missile Mouse [63] shoots the cursor across the screen when triggered by a small physical movement. Because indirect or rate-based controls are often problematic, we focus on direct control techniques. There are also absolute pointing strategies on multiple monitors by combining head tracking and mouse input [3, 11] to facilitate pointing. In our second experiment, we integrated some of these strategies to the MSC approach by modifying the head-tracking techniques to eliminate the dependence on the presence of monitor bezels to broaden its application. We also compared these head tracking and mouse input integrated techniques with the MSC-only approach.

Cursor visibility is another problem of using the mouse device in large, high-resolution displays. As the percent of the space that the cursor takes on the screen gets smaller, it becomes difficult to visually detect and follow the cursor on display. To address the problem of cursor visibility, the High-Density Cursor uses super-sampled cursor trails [10]. This technique is essentially orthogonal to the techniques we studied, and could be integrated for further improvement. Baudisch, et al. also introduces different alternatives, namely changing the cursor size according to cursor speed and using different levels of frequencies of cursor trails [10]. We decided to use the dynamic cursor size paradigm to test the scalability of the mouse device to a large, high resolution display.

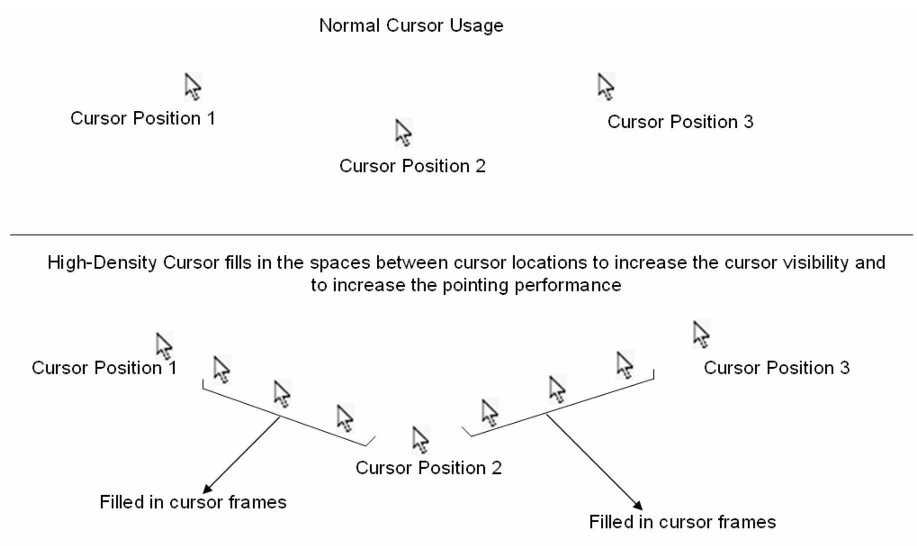


Figure 2.4: High-Density Cursor [10] uses super-sampled cursor trails to improve pointing performance.

In multiple monitors, bezels also irritate users and pose a usability problem for the mouse interaction, due to their distortive side effects such as resolution mismatch, while moving the cursor from one monitor to the other one. Baudisch, et al. investigated different aspects of dealing with bezels in multiple large monitor screens with Mouse Ether [9]. Mouse Ether tried to reduce the distortive effects of monitor, resolution or bezel mismatch in multiple monitors by keeping the cursor speed same in all of the screens. Mackinlay, et al. tried to solve the monitor mismatch in multiple monitor displays in order to facilitate viewing text and images just like in a single monitor in wideband displays [40]. Moreover, Nacente, et al. tried to solve the same problem by using user's head position and displaying the cursor according to the user's perspective [55].

2.4 Summary

In summary, this work builds on and extends previous research by considering the mouse device as the main interaction technique and testing how it scales to large, high resolution displays. After reviewing the literature on both large, high resolution display interaction techniques and mouse techniques; we decided to test the scalability of the mouse device to large displays. We believe that with some modifications on the current mouse implementation, it can still be a viable way of interacting with large, high resolution displays, especially in our scenario of large personal workspaces.

Chapter 3. The Solution Space

Different devices such as the joystick, the trackball, and the laser pointer might be possible alternatives for interacting with large, high resolution displays. Since other devices have difficulty surpassing the mouse device in terms of both accuracy and performance. (e.g. [17, 26]), we preferred to improve the mouse interaction design in our studies. We also tested the scalability of the mouse device to large, high resolution displays. However, the mouse has usability problems in large, high resolution displays. The main problem behind the low performance of mouse interaction on large displays is the difficulty of target acquisition. The most prominent problems decreasing the target acquisition performance of the mouse interaction on large displays are the clutching and the cursor visibility problems. Therefore, we will investigate solution spaces for the clutching and the cursor visibility problems in the following subsections.

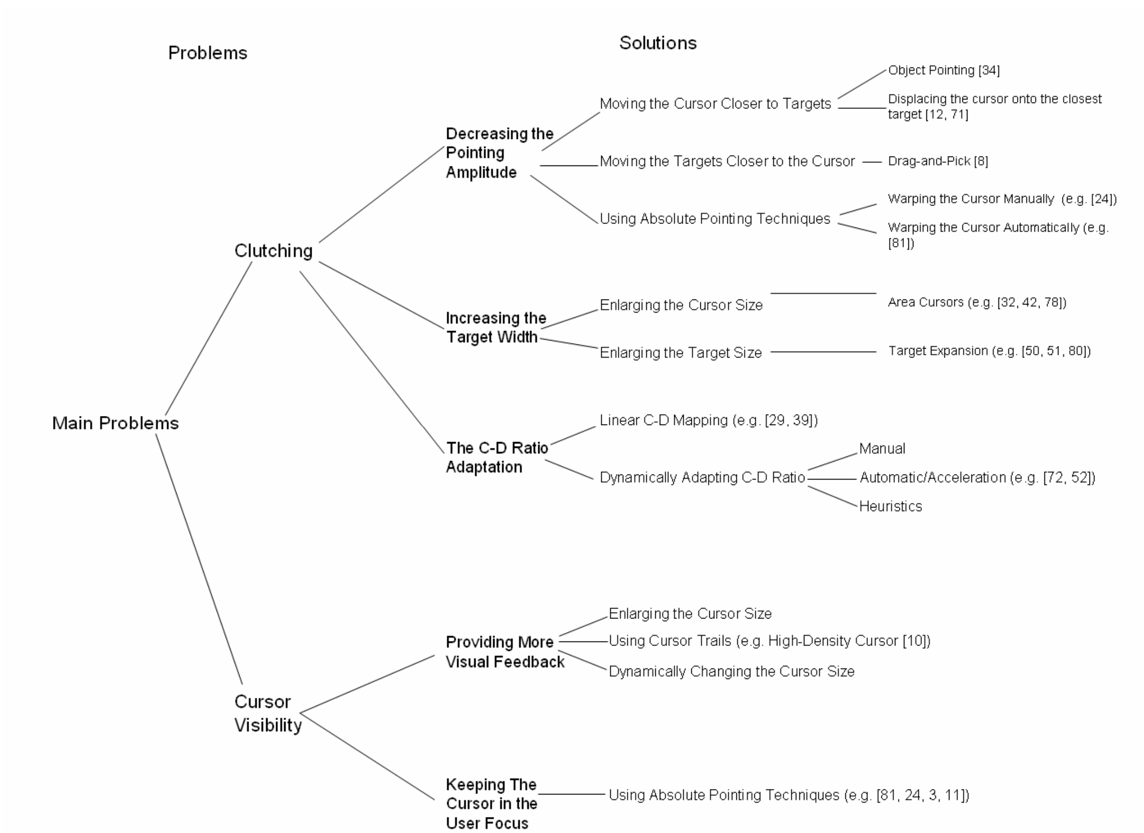


Figure 3.1: The solution space for the usability problems of mouse device on large screens.

3.1 The Solution Space for the Clutching Problem

Alternatives for solving the clutching problem, which are indicated by Fitts' Law, include decreasing the pointing amplitude, increasing the target width, and adapting the C-D ratio. The pros and cons of these alternatives are explained in Chapter 2. The scenario under investigation requires supporting the scalability of the number of potential targets for both application and desktop level pointing. Hence, we focus on the general case of pointing to any pixel on the display without requiring any information about target positions. In order to support the scalability of the number of potential targets we prefer to investigate the C-D adaptation to solve the mouse clutching problem. Because it provides a way of facilitating pointing without requiring any information about the target positions and it offers a general solution for both application and desktop level pointing. It also does not make use of any indirect controls unlike the Missile Mouse technique [63].

The C-D ratio is the ratio between the distance that the device moves and the distance that the pointer moves on the screen. Changing the static C-D ratio of the device (e.g. [48]), changing the C-D ratio dynamically according to device movement speed (e.g. [73]), or changing the C-D ratio dynamically according to the distance to targets (e.g. [13]) are possible ways to facilitate pointing by adapting C-D ratio.

We investigate dynamically changing C-D ratio according to the device movement speed (e.g. [73]) in order to support the scalability of the number of potential targets, since it does not require any knowledge about target positions. This technique facilitates fine control at the same time it facilitates long movements, since it has many different C-D ratio levels.

Dynamically adapting C-D ratio actually changes cursor speed dynamically with respect to the device movement speed. Hence, we call this technique *dynamic speed*. It is also referred as "acceleration". Typical operating system defaults for a mouse device use this technique by providing different acceleration levels which are switched based on physical mouse speed (e.g. Windows XP [52]). Dynamic Speed allows the mouse to move the cursor at different speeds, by changing C-D ratio according to device movement speed, depending on the needed scale of operation. Design options for solving the clutching problem are investigated in the subsections below.

3.1.1 Dynamic Speed Cursor

Dynamic speed cursor aims to facilitate pointing by performing well at pointing to small targets at the same time facilitating long distance movements from one side of the screen to the other one without any additional clutching action. There are three issues related to the usage of the dynamic speed technique. First, how does the user switch cursor speeds? Options include manual control by a button, automatic control by acceleration, or by intelligent analysis of task or screen contents, or statistical analysis of previous operations. Heuristics such as moving out of a window or across a screen bezel might switch to a faster mode. As indicated in [33], users typically leave items in similar places. As an illustration, an instant messaging window typically stays in the same position on a large display. Making use of muscle memory, cursor position history might be used as an alternative heuristic.

Second, how many different cursor speeds can be selected? At least two are required to accomplish the stated goals, but more levels might be helpful for a smoother transition, and a continuous spectrum is possible.

Third, how fast should each cursor speed be? To accomplish the goals, the slowest speed probably should remain the same as current system defaults. But the fastest speed must enable the cursor to cross the screen without any need to clutch the mouse device. A range of speeds between them is also possible.

Following the literature on C-D ratio/gain, we decided to take the advantage of the dynamic C-D gain and experiment with it in order to see its effects on number of mouse clutches and user performance. We used the method of dynamically changing the C-D ratio (i.e. dynamic speed), which possesses different levels of C-D ratio for different speed levels of the mouse device movement. As there is no consensus on performance gain from C-D ratio adaptation, we experimented with different numbers of C-D ratio levels, in between the same top and bottom levels. We compared their performances in order to both establish guidelines about how to design better mouse implementations by using the C-D gain and to test the scalability of the mouse device to larger displays.

While dynamically changing the C-D ratio according to the device movement speed, the C-D ratio for each level and the number of thresholds that separate those ratio levels involve infinitely many selections. To compute the necessary top speed (i.e. value of the top C-D ratio level) of the cursor, we must define “one mouse movement” – that is, the distance and the time of people’s typical maximum mouse movement without clutching. We conducted formative studies and found an average length of approximately 4,000 pixels in an average of 0.8 seconds and approximately 5 inches of mouse movement (using default base cursor speed and no acceleration, with a Logitech G7 mouse), although it varies a lot. Thus, to move the cursor across a large 16,000 pixel wide screen in 0.8 seconds, while the user only moves the actual cursor 4,000 pixels, would require at least 4x acceleration.

We investigate the effects of the number of different C-D ratio levels on user performance by conducting controlled experiments. Each C-D ratio level is divided by a threshold speed value. When the user moves the device faster than the threshold speed value the cursor is moved according to the next level’s C-D ratio. While doing that, we keep the C-D ratio for the upper and lower level the same for each cursor type design and investigate the effects of the number of thresholds on user performance.

3.1.2 Integrating Absolute Pointing Techniques with the Mouse Input

Using absolute pointing techniques such as using head tracking (e.g. [3]) and eye tracking (e.g. [22, 81]) inputs also facilitates pointing through decreasing pointing amplitude. Head tracking input, like eye gaze input, indicates our area of focus. In fact, head tracking input can be used to determine eye gaze direction with 87 to 89% of accuracy [57]. Thus, using head tracking will enable users to have the cursor in their area of focus all the time. Moreover, other absolute pointing techniques such as hand-based techniques or laser pointing either require more muscle power or have the disadvantage of preventing lazy usage. Users should raise their hand and lower it down any time they want to point to a particular position on the display. This causes higher stress on users’ hands as compared to the lazy usage of the mouse device by leaning hands on a desk or a keyboard tray while moving the device.

Therefore, we made use of head tracking inputs to facilitate pointing by integrating them with the multi-scale cursor concept. There are two issues involved with the usage of head tracking input in pointing tasks. First, how does the user move the cursor and click to targets when head tracking input is being used? Options include using the mouse device for fine control of movement and for clicking or using only head tracking input and clicking with a button. Second, how should the large cursor movements be controlled? Placing the cursor wherever the user's head is pointing can be done manually when the user presses a button or automatically when the user changes her/his area of focus.

3.2 The Solution Space for the Cursor Visibility Problem

In order to solve the cursor visibility problem of the mouse device,

- more visual feedback may be provided or
- absolute pointing techniques may be used.

In order to provide more visual feedback about the cursor to solve the cursor visibility problem, there are three design options. The cursor size may be enlarged, cursor trails may be used [10], or cursor size might be changed dynamically according to the cursor speed.

Usage of absolute pointing techniques also can partially solve the cursor visibility problem by warping the cursor wherever the user is looking and hence keeping the cursor always in the field of focus of the user.

3.2.1 Providing More Visual Feedback

To address the cursor visibility problem, following the High Density Cursor [10] idea, which uses more visual feedback to improve user performance, we decided to test changing cursor size dynamically (i.e. dynamic size). As the cursor goes faster, it increases its C-D ratio and it gets harder to follow the cursor on a display. Thus, we made the cursor bigger each time it increases its C-D ratio level. In this way, the cursor gets smaller when the user is moving the mouse device for fine control. This solves the cursor visibility problem without causing data obscuring problems.

Dynamic Size tightly couples the displayed size of the cursor with its dynamic speed. Thus, when the dynamic cursor switches to a faster speed, it also grows larger in size. One possibility is a directly proportional relationship, such that a cursor moving at 4x speed would be 4x in size (width). The purpose of dynamic size is twofold. First, it provides the user with direct visual feedback about the scale of operation of the mouse so as to avoid confusion and overshooting targets. Second, it solves the problem of visibility in accomplishing the original goals. That is, when moving slowly for fine control, the standard small cursor size avoids obscuring data on screen. When moving fast, the larger cursor size helps users visually track the cursor moving rapidly across the screen, and obscuring detailed data that is being passed over is irrelevant during a large scale operation. Without a dynamic size cursor, the percent of the screen space occupied by the cursor gets smaller as screen resolution gets higher, making it difficult to visually detect the cursor.

3.2.2 Usage of Absolute Pointing Techniques

Usage of absolute pointing techniques which warps the cursor wherever the user's field of focus on the screen is partially solves the cursor visibility problem. If the cursor always stays inside the user's field of focus, it will not be hard to detect and follow the cursor. But this can cause additional problems, such as irritation caused by unintentional cursor movements.

We also used cursor trails coupled with head tracking input usage. While warping the cursor, when users change their head position, we used cursor trails as an animation to increase the awareness about where the cursor is and where it is going.

3.3 General Solution: Multi-Scale Cursor

We propose a multi-scale framework for mouse interaction, in which the mouse operates at multiple levels of scale. The goal is to test the mouse device's scalability by enabling rapid large-scale cursor movement while maintaining performance of small fine-grained control. First, following the dynamic speed cursor [73] and improving user performance via using more visual feedback [10] ideas, we introduce and experiment on a dynamic speed & size (DSS) cursor that can grow and shrink between large and small scales of operation. The standard small slow cursor becomes big and fast when larger movements are needed. The MSC concept basically involves usage of dynamic C-D ratio adaptation according to the device movement speed and grows or shrinks the cursor size proportional to the cursor speed. Second, we experiment on integrating quick coarse absolute pointing techniques, such as head-tracking, as a potential alternative to support large-scale movement and to compare those techniques to the MSC concept.

A multi-scale mouse technique can operate the cursor at multiple levels of scale, ranging from fine detailed pixel-level control to rapid movements across a large screen. A critical goal is to reduce or eliminate clutching. Users should be able to move the cursor from one side of a large workspace to the other with one swift mouse movement, without the need to repeatedly lift and move the mouse again. Moreover, we must also preserve the ability to carefully manipulate the cursor to point to small targets or any individual pixel on the screen. Thus, multi-scale cursor affects both speed and size of the cursor (Figure 3.1). We used Fitts' Law [27] as a framework for comparing designs. None of the previous experiments have examined mouse input at the level of the scale discussed here (100 megapixels).

Figure 3.2: The Dynamic Speed & Size Cursor (a subset of MSC) moves faster and grows bigger for long distance movement, then slows down and gets smaller for fine detailed control.

We conducted two experiments with the aim of seeking and evaluating better mouse cursor designs for large displays and to test the mouse device's scalability. In the first experiment we aimed to find the best MSC design, by testing several design alternatives for scalability. In the second experiment, we aimed to test the best MSC design from experiment 1 against other successful approaches for fast coarse pointing, such as head-tracking interaction techniques.

The ultimate goal is to design a scalable interaction technique and to determine whether the mouse device can still be an effective interaction technique in large displays that works well for reaching targets at all distances, from near to very far. Thus, the goal of these studies is to explore the design space, to evaluate the alternatives in terms of movement time, number of mouse clutches, and error rates in typical Fitts' Law pointing tasks and to optimize the design in order to effectively test the scalability of the mouse device to large, high resolution displays. Since the scalability is the major concern, we tested the techniques on a very large, curved 100 megapixel display to ensure that the longest cursor movement distances were on the extreme high end.

3.4 Scalability

The scalability of a cursor design (mouse implementation) is the effect of the screen size on the pointing performance. As indicated by Fitts' Law [47], the pointing performance of a mouse design changes according to the target distance. As the target distance gets longer, the pointing performance of each mouse design deteriorates. The scalability of a mouse design is determined according to this deterioration rate. If a cursor design can perform well on short distances and does not deteriorate much as the target distance gets longer, then that mouse design is said to be scaling well to that given screen size. The framework that we used for testing how a cursor design scales to a given screen size is explained below.

In order to determine whether or not a cursor design performs well on short distances, we compared the short distance pointing performance of our designs and the default Linux cursor (control cursor), which is commonly used on one-monitor personal computers. In order to facilitate pointing to any pixel, a cursor design can not have a significantly lower performance on short distances as compared to the Linux cursor.

We investigated pointing performance of each cursor design to test how they scale to a given screen size. Number of mouse clutches, pointing time, and number of error clicks are all used to determine the pointing performance of a cursor design. We used pointing time also to determine how much the performance of a cursor design deteriorates as the target distance gets longer.

We also fit our data to Fitts' Law, since pointing time is investigated in Fitts' Law [79] and used to distinguish the performances of each design by the definition of index of performance. Index of performance [79] is bits per second that a cursor design can transmit. A higher index of performance indicates a better pointing performance and thus better scalability.

In our experiments we kept the screen size constant. We tested each mouse design in six different target distances. In this case, the mouse design, which performs as well as the control cursor on short distances and does not deteriorate much as the target distance gets longer, can be said to be scaling better to even a 100 megapixel display.

We experimented on the solution space by a number of different designs with the scalability framework explained above. We tested three different C-D ratio adaptation methods to search for the best design choices in terms of the number of thresholds, number of C-D ratio levels and speed values for each level. We varied the number of C-D ratio levels in three different designs. The first design had only two C-D ratio levels, the second design had three C-D ratio levels and the third design had four C-D ratio levels. We also tested the effect of using more visual feedback on the scalability of the mouse device to large, high resolution displays. Moreover, we also investigated the best way to switch cursor speeds and compared changing the cursor speed manually or automatically.

We also investigated further scaling up the mouse device to large displays beyond the performance of the MSC concept. We integrated the MSC concept with the absolute pointing techniques and compared their performances with the MSC-only concept and with the control cursor.

Chapter 4. Experiment 1: Designing the MSC

We first investigate the dynamic speed and size technique for large high-resolution workspaces. The goal of the first experiment is to determine effective settings for the number of thresholds, the configuration of cursor speeds and the method of changing the cursor speed (manual or automatic). We also want to make steps toward optimizing the multi-scale cursor design by comparing several alternative implementations. To accomplish this goal, we investigate how the dynamic speed and dynamic size behaviors of the cursor will each individually affect user performance in terms of mean movement time, number of mouse clutches, and error rates in Fitts' Law pointing tasks.

4.1 Cursor Types

We tested eight different cursor settings (4 different speed behaviors x 2 different size behaviors) in the experiment-1 (Table 4.1). We, generally, had three categories of cursors; dynamic speed cursor (with two different versions DC1T and DC2T), manual cursor, which allows users to change the acceleration of the cursor manually, and a control cursor. Also, each of those four different cursor types (control cursor, DC1T, DC2T, manual cursor) has sub-versions according to their size behavior which might be dynamic or static size.

To investigate the *dynamic speed*, which consists of dynamically changing C-D ratio according to device movement speed, variable, we combined its various sub-factors (method of switching speeds, number of speeds available, and speed values) into a single variable for simplicity and because there are interdependencies. Note that the base speed, 1x, is identical for all the cursor conditions. These designs resulted from formative testing to pick out the most promising designs. The four selected dynamic speed cursors were as follows:

Control Cursor (1x 2x): The standard Linux operating system cursor type offers 2 cursor speeds, a base speed and an accelerated speed that is automatically selected when the mouse is physically moved faster than a predefined threshold. The accelerated cursor speed is set to the default value of 2x of the base speed.

DC1T (1x 4x): Like the control cursor, this is a dual-speed (1 threshold) cursor that could be either fast or slow depending on how fast the user moves the mouse, but with a faster acceleration. If the user moves the mouse slower than threshold speed, then the cursor moves with the speed 1x. If the user moves the mouse faster than the threshold speed then the cursor moves with the speed 4x. The threshold is 100 base speed pixels in 0.2 seconds. This design attempts to achieve greater scalability, but potentially at the cost of more speed discontinuity.

DC2T (1x 2x 4x): A triple-speed cursor that has two different threshold levels and three different speeds as 1x, 2x and 4x. The thresholds are 100 and 450 base speed pixels in 0.2 seconds, respectively. This design attempts to establish a smoother multi-scale speed up process.

Manual Cursor (1x 6x): A manually controlled dual-speed cursor that can be either fast or slow depending on a user mode selection. By pressing the right mouse button or the space bar, users can toggle between high-speed cursor mode and the normal speed mode. The base speed in manual cursor is set to 1x and the high speed is 6x. Manual control gives users more explicit control of the speed, eliminating discontinuity, and therefore offers the potential for usable increased top speed of 6x.

As an illustration on the usage of *dynamic speed*, if the user was using DC1, and was detected as physically moving the mouse slowly across the desk, for m pixels, below the given threshold of 100 pixels in 0.2 seconds, then the operating system would move the cursor m pixels on the screen accordingly. Then, suppose the user made a faster physical movement, for n pixels, of the mouse on the desk, with a speed above the threshold of 100 pixels in 0.2 seconds, then the operating system would move the cursor 4*n pixels on the screen. In the manual cursor, users can change the speed of the cursor whenever they want by pressing the space bar.

To investigate the *dynamic size* variable, we compare the proportional dynamic size technique against static size:

Static Size: The arrow cursor always stays the same size as the default system cursor size, which is approximately 20x20 pixels.

Dynamic Size: The arrow cursor dynamically grows and shrinks in visible size proportional to the amount of speed up. For example, when the mouse is in 2x speed mode, then it will be 2x in size (length and width), 40x40 pixels.

Thus, a total of eight different cursor types are compared in the experiment. Each of the four different dynamic speed cursor configurations has two sub-types as static and dynamic size.

4.2 Hypothesis

Our hypothesis was that dynamic size behavior of the mouse and automatically changing cursor speeds will help users finish the tasks in less mean movement time as compared to other alternatives. Our second hypothesis was that a greater number of thresholds in a cursor design will result in both less mean movement time and less number of mouse clutches, and thus will result in better performance as compared to fewer thresholds.

4.3 Hardware and Software Specifications

The experimental hardware consisted of fifty 20-inch LCD monitors (1600x1200 resolution each) and 25 computers running GNU/Linux to create a large, high-resolution display. Each computer powered two monitors. We set the screens on stands in a 10x5 matrix. To set the distance between the user and each column of the large screen to the same value, we curved the screen 120 degrees in total. This allows users to see a small target anywhere on the screen, while seated, with minimal body movement to look from side to side.

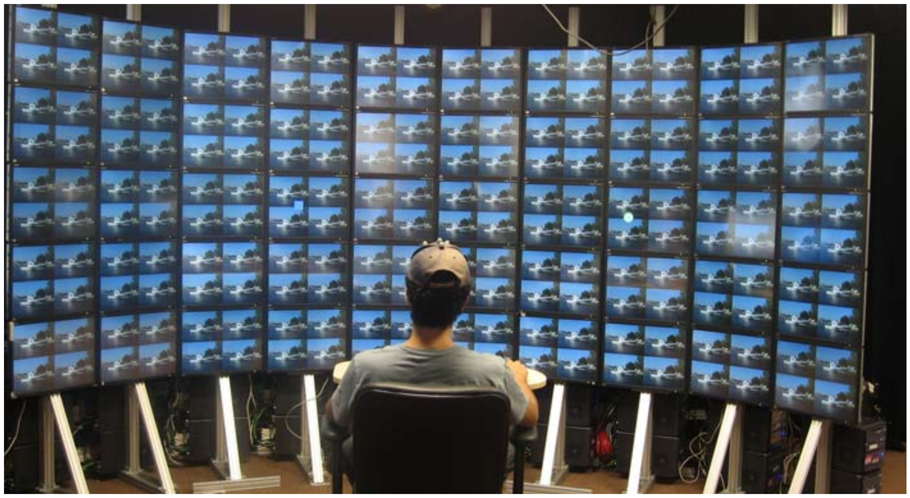


Figure 4.1. Experimental floor plan. (a) A user interacting with the display. (b) Floor plan of the display configuration with respect to the room.

Subjects sat at a distance of approximately 86 inches from each column of the screen (Figure 4.1). Participants used a wireless keyboard and wireless optical two-button mouse. The reason for using the wireless devices was due to the distance of the participants from the computing cluster, which was behind the display.

We implemented a working prototype of the cursor techniques in an OpenGL [61] application using Glut [31] and Chromium [19], an open source library for parallel rendering of OpenGL.

4.4 Design and Procedure

The independent variables used in the first experiment are:

- Cursor speeds: Control cursor, DC1T, DC2T, Manual cursor
- Cursor size: Static Size – SS, Dynamic Size – DS
- Target width: 20, 60, 180 pixels
- Target distance: 200, 3200, 6200, 9200, 12200, 15200 pixels
- Angles: 0, 180 degrees.

In some cases we discuss cursor type as a combined variable of cursor speed and cursor size. The dependent variables are task compilation time, number of mouse clutches, percentage of error clicks, and subjective satisfaction. The tasks used in this experiment are typical Fitts' Law pointing tasks.

A repeated Latin Square design was used for counterbalancing of the cursor types. A within-subject design was used to reduce variability between participants. Each participant performed the same tasks in random order for each of the eight different cursor types.

For each cursor type, subjects used the cursor to perform 6 trial practice tasks. They then performed a block of trials for that cursor type. Each block consisted of a single trial for each of the 18 distance-width combinations at 2 different angles for a total of 36 trials per block. Each of starting and ending targets of those 36 trials were predefined, and the order is randomly shuffled for each block. So, each block actually consisted of the same tasks, but in a different order due to randomization. Each distance-angle pair's starting-ending targets were randomly positioned on the display and we prevented any target to fell on a bezel. That is to say, each distance-angle pair's starting-ending targets were set on the same position, but those positions were randomly selected.

If a participant missed the target and clicked somewhere else, we passed that task and counted that task as an error task. Error tasks were given back to the user at the end of each block in random order. Targets were placed on a background wallpaper to simulate typical visual clutter in a desktop environment. We used circular end targets and square start targets. As soon as a user clicks to a starting target, we warped the cursor to the center of the square shaped starting target to minimize the variability of the target distance that the cursor needs to be moved until reaching the end target. We used green color to indicate which target should be clicked in that round. We used blue color to indicate the passive target, which should not be clicked on that round. Subjects were asked to finish the tasks as quickly and accurately as possible. To motivate subjects, a 30$ cash prize was offered for the subject with the shortest time to finish all the 108 tasks.

The particular settings of our experiments were slightly different from typical Fitts' Law experiments. Although we used angles of 0 and 180 degrees, which includes only one dimensional moves, to simulate more realistic tasks we used distributed in varied positions on the screen in a random order, instead of two vertical bars placed only in the middle horizontal dimension. Moreover, as the screen size gets larger it gets harder to detect an object on the display. In the experiment we aimed to count only the time it takes to move the cursor from the starting target to the ending target, not the visual search for finding the targets on such a large screen. In order to decrease the time that it takes to find the starting and target positions, we turned off the wallpaper until users click the first target on a black screen. When users click the first target, we turned on the wallpaper on the display to simulate a desktop noise (Figure 4.2). To avoid counting the time for subjects to locate the targets, we forced them to click the ending target first, then click the starting target, and then return back to the ending target and click it. That is to say, each of our tasks consisted of three clicks and we only counted the time between the second click and the third click. This forced users to find both targets first before clicking the start target. The task used in the experiment is explained in detail in Figure 4.2.

All tasks were tracked automatically by our software. After each task was completed successfully the software automatically logged and time stamped the performance for each successful task compilation. Mouse clutches were observed visually. Also, participants completed a questionnaire after the experiments about which cursor technique they preferred.

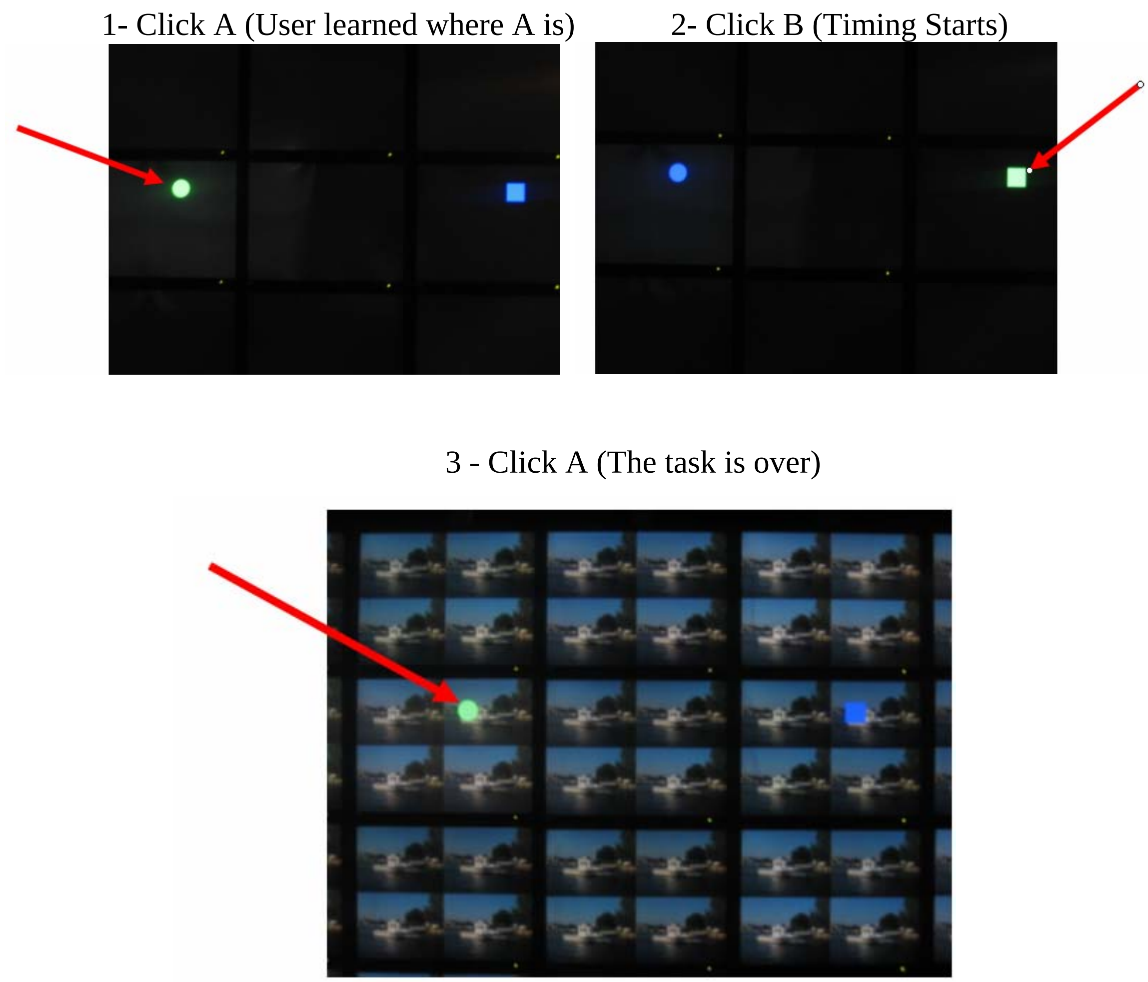


Figure 4.2: Experiment background set up. (a) Black background before the first click. After the first click user learns the location of the target A. (b) Black background after the first click and before the second click. User needs to click the target B. (c) After the second click the timing starts, the cursor is warped to the center of the target B and the background image is turned on to simulate a typical desktop noise. Now, user needs to click the target A as soon as possible.

There were 16 participants (8 male, 8 female), ranging in age from 21 to 40, in this experiment. None of the participants were color blind; all had normal or corrected normal vision. The participants were a mix of graduate students and individuals residing in Blacksburg. They were recruited by solely word of mouth. Only two of the participants used large display systems before to conduct scientific experiments.

4.5 Results

4.5.1 Movement Time

We performed all our statistical analysis in SAS's JMP using standard ANOVA and Chi Squared techniques after removing outliers. We defined outliers by using Grabbs' test. Grabbs' test detected only around 1.3 % of the data as outliers.

4.5.1.1 The Effect of the Cursor Type on Movement Time

To determine an overall winner, we found a statistically significant effect of cursor type on performance time F (3, 3974) = 10.37, $p < 0.0001$. Then we compared each cursor type with the Tukey adjustment and we found that DC2T with dynamic size performed statistically significantly better than all the other cursors in overall time, with the mean time of 2.31 seconds, and the static size manual cursor performed statistically the worst with the mean time of 3.10 seconds (Figure 4.3).

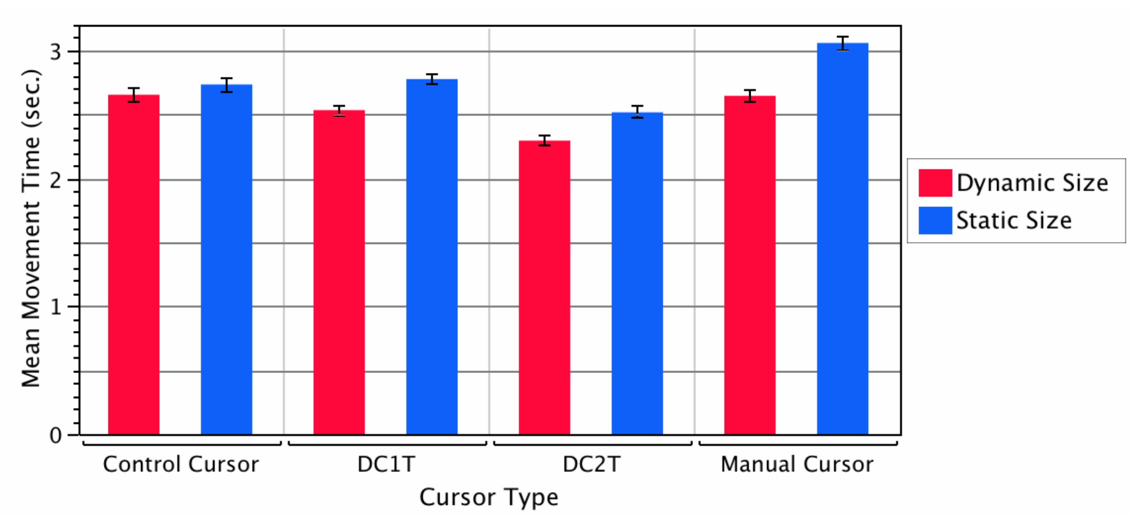


Figure 4.3: Average overall performances of all 8 cursor types in the Experiment 1.

Although DC1T and DC2T had the same base speed and the same top speed, there was a statistically significant difference between their overall performances, due to the additional intermediate speed of DC2T. The faster acceleration speed of DC1T (4x) over the control cursor (2x) did not appear to make much difference. The manual cursor was problematic with static size, because there was no visual feedback about the current speed mode. Adding dynamic size helped, but again the increased acceleration (6x) was not advantageous over the slower control mouse (2x). One possible interpretation is that larger jumps in C-D ratio can be problematic, while more small smoother speed increases are helpful.

4.5.1.2 The Effect of the Cursor Speed and the Cursor Size on Movement Time

Cursor speed configuration has a statistically significant effect on performance F (3, 3974) = 80.58, p < 0.0001. Comparing each cursor speed configuration with the Tukey adjustment, showed that DC2T performed significantly better than the others in overall performance. Control cursor performed significantly worse than the other cursor types. Performances of DC1T and manual cursor speed settings were significantly different.

We found a statistically significant effect of *cursor size* on performance F (1, 3974) = 130.86, p < 0.0001. Although each of the two subtypes of each cursor have identical speed configuration, the size behaviors of each cursor affected the performance. Dynamic size cursors performed better than static size cursors.

4.5.1.3 The Effect of the Target Width on Movement Time

As expected, target size (i.e. target width) significantly effected performance F (2, 3974) = 922.93, p < 0.0001. This is true for all the cursor types (Figure 4.4).The outlier here is dynamic size manual cursor, where the effect of target size is amplified more than other cursors. This might indicate that users chose to stay in that cursor's Big-Fast mode. Moreover, for every cursor, when target width gets larger, mean movement time decreases.

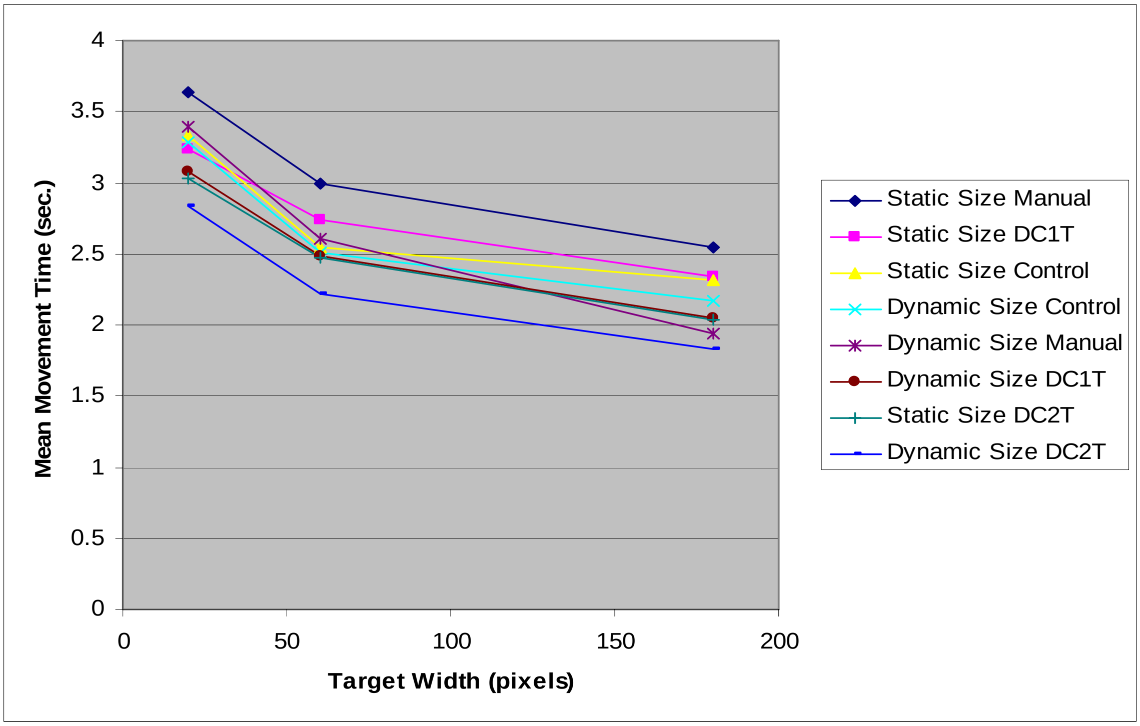


Figure 4.4: Mean movement time (sec) vs. target size (pixels).

4.5.1.4 The Effect of the Target Distance on Movement Time

Distance to targets (pointing amplitude) also significantly affected performance F (5, 3974) = 1143.93, $p < 0.0001$. However, the amount of effect varied for each cursor type used, due to the significant interaction between cursor speed and target distance F (15, 3974) = 9.91, $p < 0.001$ (Figure 4.5).

Figure 4.5 shows an interesting relationship between cursor technique and distance to target which explains the significant effect of cursor type on performance F (15, 3974) = 9.91, $p < 0.001$. Although the best overall performance was found in the dynamic size DC2T, we were not able to statistically differentiate the control cursors from DC2T for the first two short range targets, nor from dynamic size manual cursor for the longest range targets. For the other 3 target distance ranges (6200, 9200, 12200 pixels), DC2T outperformed all the other cursors. This might indicate that the design of DC2T successfully embodies the advantages of both the control and manual cursors (Figure 4.5).

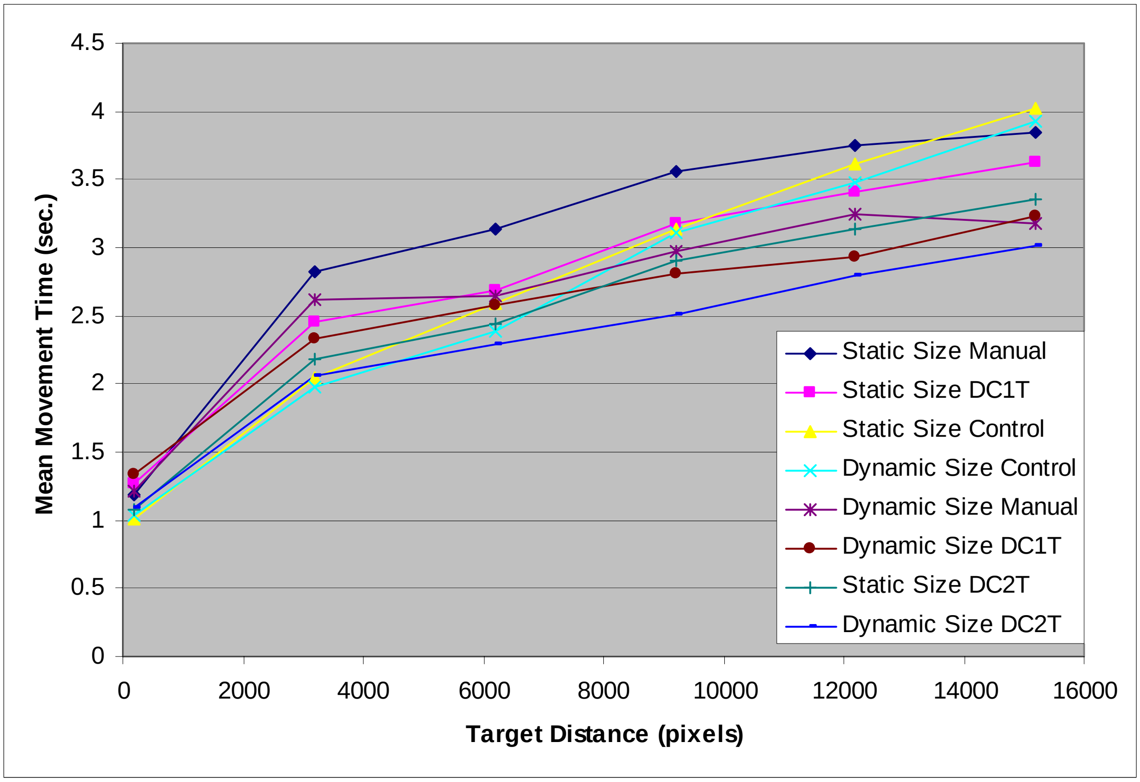


Figure 4.5: A chart showing how each cursor technique varied in mean movement time.

When the distance to target is small, such as 200 pixels, the control cursors' mean movement time is faster than most other cursors. However, as the distance increases, an interaction effect is seen that the control cursor performs increasingly worse and ultimately has the worst performance at long distances. After 3200 distance, the control mouse seems to degrade linearly. This might indicate that the incremental motion of clutching (discussed later) causes a more linear pattern. To illustrate the point, the difference in performance between the short and long distance for the control cursor is 3.10 seconds, whereas the difference between the short and long distance for the dynamic size DC2T was 1.97 seconds.

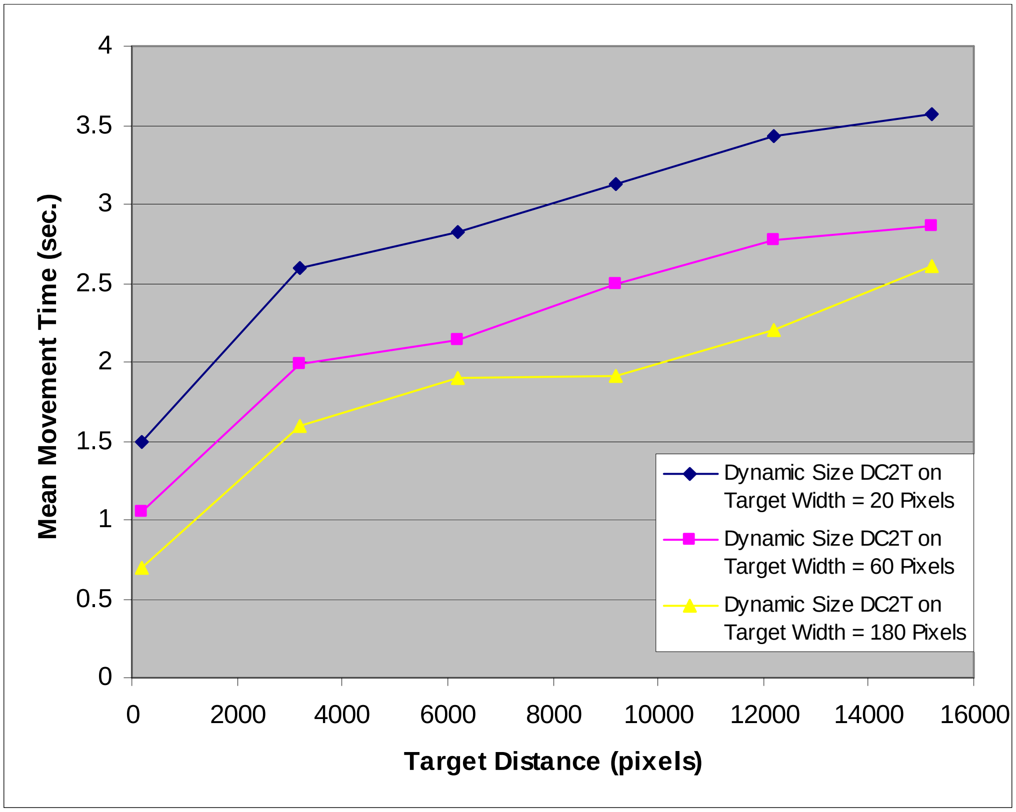


Figure 4.6: Mean Movement time (sec.) versus target distance (pixels) visualized for dynamic size DC2T in three different target sizes.

Figure 4.6 shows how DC2T's mean movement time depends on target size and target distance. The three different colors represent mean movement times of DC2T with three different target sizes. As can be seen in the graph, the target size is a major factor in pointing performance. In each of the distances DC2T's mean movement time decreases when the target size increases. There is not even a single target distance point where DC2T performs better on a smaller size target. The figure also shows the gap between the mean movement times of target distances 200 and 3200 pixels. This gap gets smaller as the target distance increases and DC2T manages to flatten the line between mean movement times between the distances 12200 and 15200 pixels.

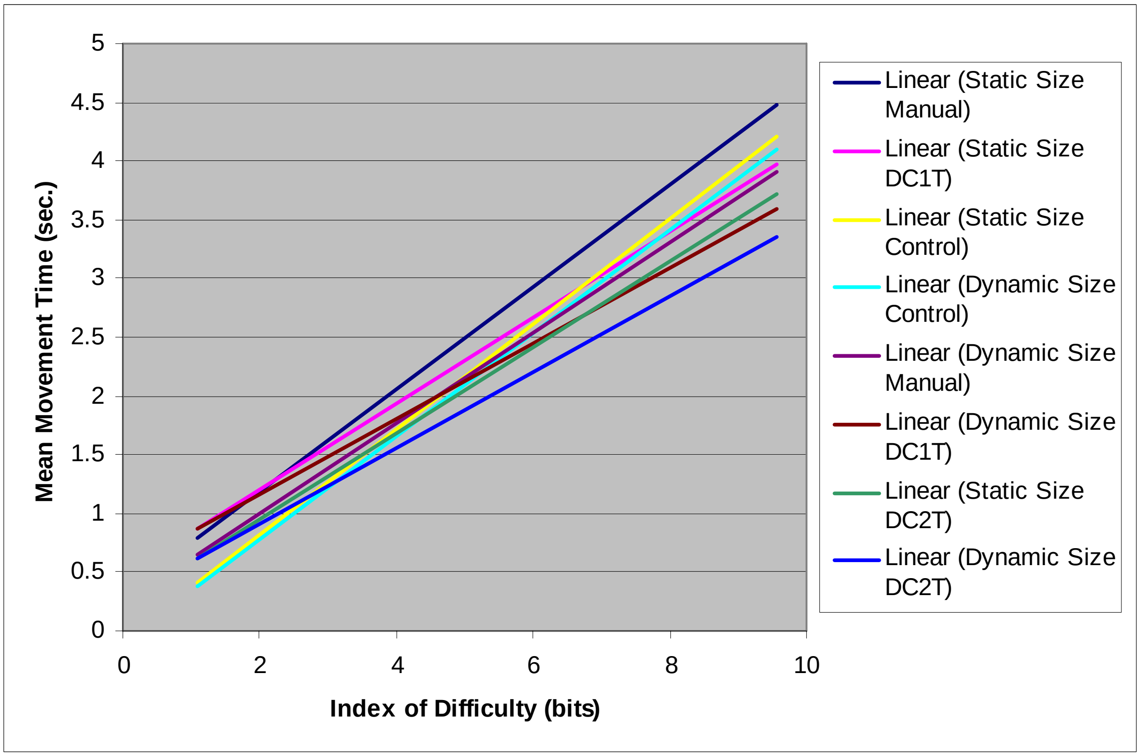


Figure 4.7: Linear Regression results for Mean Movement time (sec) vs. Index of Difficulty (bits).

Index of difficulty (ID) is computed using Fitts' Law Index of Difficulty ID = $\log_2(A/W + 1)$ [44], where A is the amplitude, representing distance to the target, and W is the width of the target. As implied by the equation, ID is directly proportional to the pointing amplitude and inversely proportional to the target width (W). Fitts' Law Movement Time (MT) can be expressed in terms of ID by the equation; MT = a + b x ID [44]. Figure 4.7 shows the relationship between index of difficulty (ID) and the mean movement time (MT). Figure 4.7 illustrates the linear regression results. It can be clearly seen that both static and dynamic size control cursors' mean movement time was close to the dynamic size DC2T's mean movement time in the beginning, but their performance had the most amount of variation as the target distance increases. Both static and dynamic size control cursors ended up having the highest mean movement time after the static size manual cursor for higher levels of index of difficulty. Dynamic size DC2T had less variation in its performance than all the other cursor types.

Linear regression results of fitting our data to the Fitts' Model showed that cursor types in this experiment resulted with r^2 values ranged between 0.872 and 0.976. The indices of performances for each cursor ranged from 2.243 to 3.119 bps (Table 4.2). Static size control cursor had the lowest index of performance (IP). Dynamic size DC2T and DC1T possess the largest IPs which are 3.10 and 3.11.

Cursor Type	**Size Type**	**a**	**b**	**Index of Performance**	R^2
Control Cursor	Static	-0.061202	0.445653	2.243897	0.872184
DC1T	Static	0.4778959	0.3650953	2.739011	0.942993
DC2T	Static	0.219028	0.3659048	2.732951	0.960481
Manual Cursor	Static	0.3296155	0.4340321	2.303977	0.976951
Control Cursor	Dynamic	-0.098561	0.4384147	2.280945	0.871303
DC1T	Dynamic	0.525895	0.3205179	3.119951	0.96592
DC2T	Dynamic	0.2685887	0.3223424	3.102291	0.965757
Manual Cursor	Dynamic	0.2331634	0.384393	2.601504	0.946963

Table 4.1: Linear regression results of fitting the data into Fitts' Model for the Experiment 1.

4.5.2 Number of Mouse Clutches

With the dynamic speed cursor, the number of clutches (when the user lifts the mouse from the surface) decreased dramatically (Figure 4.8). To illustrate the point, the average number of mouse clutches per task for the dynamic size DC2T was 0.23, which is less than one clutching per task, whereas it was 1.26, which is five times more than DC2T's clutches per task, for the static size control cursor.

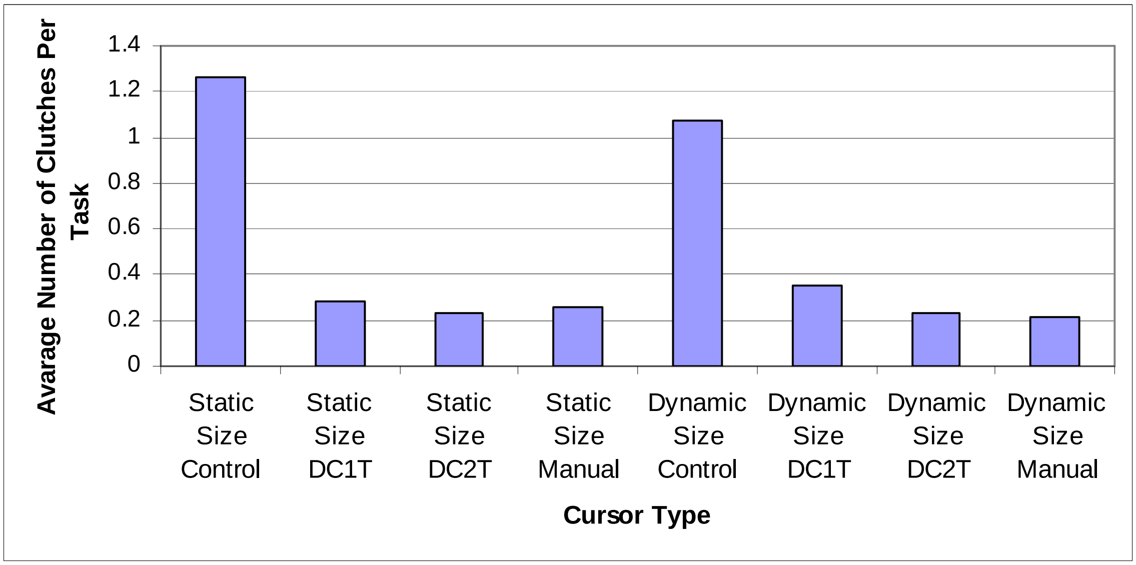


Figure 4.8: Number of "clutches" for each cursor type per task.

4.5.3 Error Rate

The only significant effect on number of errors was target size $F\ (2, 113) = 20.62$, $p < 0.0001$. Participants tend to make more errors on smaller sized targets, as expected. Most error rates were below 1%, which is well below the <4% range in standard Fitts' law experiments (Figure 4.9). This is probably because the larger targets were very easy to click.

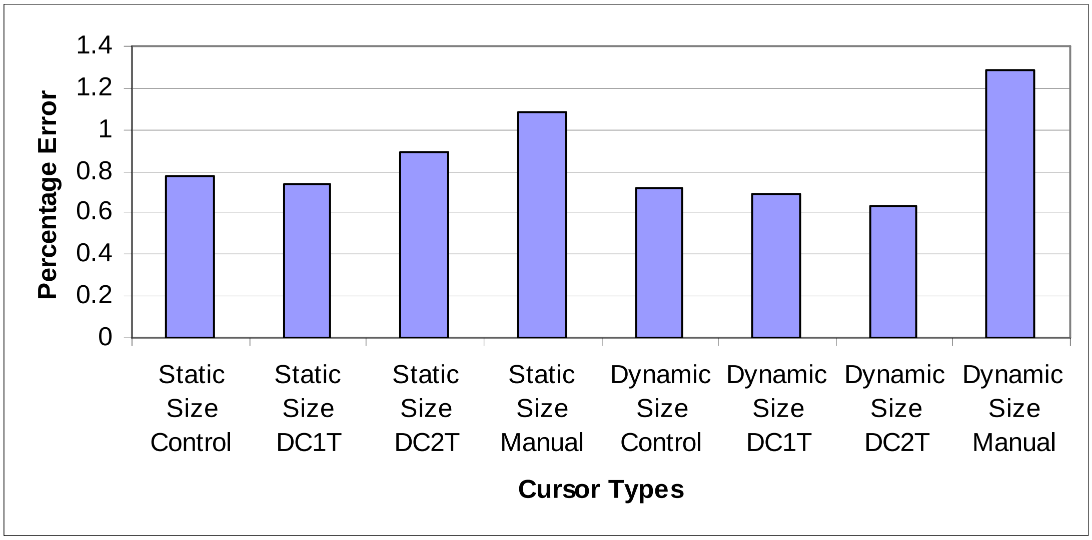


Figure 4.9: Percentage of errors for each cursor type.

4.5.4 Survey Results

The technique most preferred as the easiest to use was dynamic size DC2T. It was chosen as the easiest to use by 11 of 16 participants. The other 5 participants elected dynamic size manual cursor as the easiest to use cursor. Both static size and dynamic size control cursors frustrated subjects due to high frequency of mouse clutches. Thus, nobody preferred them.

All of the participants elected dynamic size cursor types as the easiest to use. They noted that they liked the visual feedback about the speed of the mouse when it changes its size. They thought it was natural to recognize that the larger sized cursor should go faster, which helped them control the cursor properly. Moreover, some of the participants asserted that with the help of the dynamic size cursor they no longer needed to follow the cursor on the screen with their eyes. Instead, they could look straight at the target and follow the cursor with their peripheral vision.

Participants did not like both dynamic and static size DC1T, since participants found them jumpy on short distances. Our belief is that irritation is caused by the fact that DC1T changes its speed from 1x to 4x. On the other hand, participants noted that they did not need to think about how they should sensitively move the mouse device for a long or short distance with the DC2T, as it moves the cursor smoothly and performs well on both short and long distances. One participant explained, "I am not afraid of overshooting targets, since I know that even if I overshoot, it would be easy to come back [with DC2T]". DC2T eliminated the problem of being jumpy by having two thresholds and three different speeds as 1x, 2x, and 4x.

The dynamic size manual cursor was the second most preferred technique. Five of the subjects, who preferred dynamic size manual cursor, noted that this cursor type gives full control of the cursor speed to the user and so they knew the cursor speed even before they began to move the cursor. But two of the subjects who preferred this cursor type noted that, if they were asked to use a mouse like this in their daily usage, it might be quite tiring to change the mouse speed manually all the time.

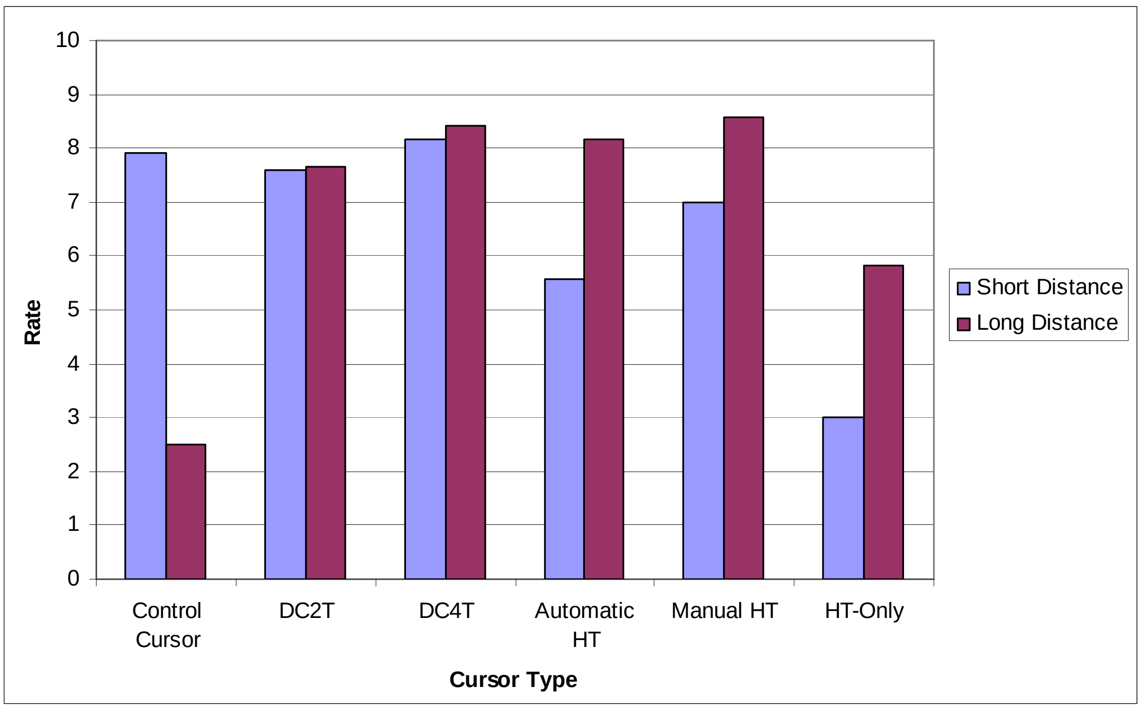


Figure 4.10: User rates of each cursor type on short and long distances.

We also have given questionnaire after each block related to that particular cursor type. We wanted participants to rank each cursor type's ease of use in short distances and long distances in a scale of 10, where 0 is not effective at all, 10 is highly effective. Figure 4.10 shows that participants ranked DC2T as highly effective on both short distances and long distances. Participants rated DC2T as good as control cursor on short distances, which indicates that they did not have to deal with overshooting problems for close targets. It also shows that the transitions between C-D ratio levels were smooth and did not bother users. Moreover, participants ranked DC2T higher than dynamic size manual cursor on long distances, even though dynamic size manual cursor has a higher speed than DC2T which might ease long distance movements. Our belief is that users ranked DC2T higher than dynamic size manual cursor since DC2T requires only usage of one hand while manual cursors requires the usage of both hands and thus increases the stress on the user's motor skills.

It is interesting that participants ranked *dynamic size* version of the same cursor type on the same distance type higher than the same cursor's *static size* version. This indicates that participants felt more comfortable with the *dynamic size* cursor types, thus they ranked them higher than *static size* cursor types.

4.6 Discussion

In the light of our first study, we are able to identify some better mouse cursor designs for large displays. It is interesting that although DC1T and DC2T had the same base and high speeds their performances were statistically different. Moreover, DC1T was described as jumpy; on the other hand DC2T was described as easy to control. Thus, we can conclude that the speed differences between thresholds must not be too large, probably about 2x difference. Decreasing the speed differences between each speed value of a dynamic cursor by adding more thresholds resulted in better user performance.

Dynamic size cursors also performed significantly faster than static size cursors. Although users already knew where the cursor was in the beginning of the task, multi scale cursors improved the pointing performance and provided better user control over the cursor movements by providing more visual feedback about both where the cursor is and about its speed. It was also hard to see small fast moving cursor on a noisy background as stated by some of the participants. With the dynamic size cursors we increase the cursor visibility while avoiding data obscuring problems.

Although the manual cursor had a higher maximum speed than DC2T, it performed statistically worse than DC2T. Also, the manual cursor had some usability problems stated by some of our participants regarding the usage of extra buttons to change the speed of the cursor.

Overall, we recommend focusing on automatic dynamic speed and size cursors for further refinements. Subjects could perform well on both long and short distances, had less mouse rowing, and avoided cursor visibility problems with DC2T. After this experiment, we began to believe that dynamic size and speed cursors like DC2T could well enable mouse interaction for large displays in the future. Next, we continued our studies of the multi-scale mouse concept by merging our dynamic speed and size cursor with absolute pointing techniques in the field such as [3] and [81], to compare more context aware systems to our cursor designs.

Chapter 5. Experiment 2: Integrating Absolute Pointing Techniques with the Mouse Input

The goal of the second study is to determine how improved dynamic speed and size cursors perform as compared to head-tracking interaction techniques for coarse absolute pointing, and whether integrating those techniques will further improve the multi-scale mouse concept. Also, because the multiple thresholds method for dynamic speed performed well in the Experiment 1, we wanted to explore how further increasing the number of thresholds in dynamic cursors will affect user performance.

5.1 Cursor Types

In this experiment we compared six different cursor configurations. We had three categories of cursors: We had two versions of dynamic size and speed cursor; two versions of a combination of head tracking and mouse input; and a default Linux operating system cursor and a head-tracking only cursor as controls for comparison.

DC2T with dynamic size: The best overall performance in the first experiment (1x 2x 4x).

DC4T with dynamic size**:** An expansion of the dynamic cursor which has the same lowest and highest speed as DC2T, but has 4 thresholds, C-D ratio levels and 5 different speed and size values as 1x, 1.75x, 2.5x, 3.25x, and 4x. We included DC4T in our experiment to see if we can further improve DC2T's performance by increasing the number of thresholds, C-D ratio levels.

Secondly, following the ideas purposed in [4], [67] and [40], we wanted to compare best of MSC cursor designs by combining them with head tracking inputs. The motivation is to attempt to improve long distance movements by using absolute pointing techniques to immediately warp the cursor to a desired location. Absolute pointing is typically not very accurate, so coupling it with the MSC may be a good solution. Head-tracking can move the cursor to where the user's head is looking, which is a close approximation to gaze.

We created two versions of head-tracking by merging DC2T implementation and head tracking inputs.

AutoHT: An automatic head-tracking implementation, which has a threshold value of 2000 pixels to move the cursor wherever users head is pointing. If user moves his head for more than 2000 pixels, the cursor is warped to wherever his head is pointing on the screen. If the user's head position stays within the 2000 pixel wide region, then the cursor is not moved automatically. We combined dynamic size DC2T with this so that users can move the cursor for fine control with the mouse.

ManualHT: A manual version of the head-tracking technique attempts to decrease the level of automation and reduce user irritation. With ManualHT, we do not move the cursor automatically, but whenever the user presses the space bar on the keyboard, we warped the cursor to the position that the user's head is pointing. This will avoid unintentional automatic moves.

HT-Only: A follow up to the idea in [40], but we used head tracking inputs instead of eye tracking input to design an alternative technique to compare with the ones in [67] and with best of MSC designs. HT-Only is a mouse-free interaction technique which makes use of only head-tracking inputs to move the cursor and uses keyboard to click targets. While using this technique, when users first click a point we magnify the local area by 4x, then users need to move the cursor with their head on top of the target and press the space button again to be able to select the target. Unlike normal mouse interaction, this technique forces users to press space button twice to select an item, first to magnify the 300x300 pixel^2 area by 4; then to select the target. This will serve as a control for the combined head-tracking techniques.

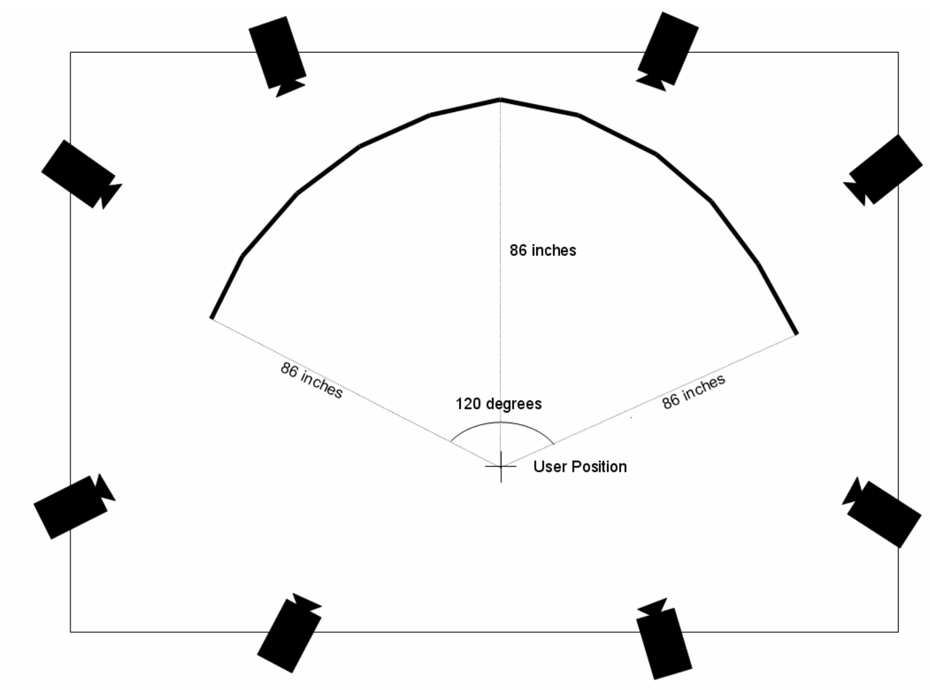
Control Cursor: The standard Linux operating system cursor type, with mouse acceleration set to 2x as in the first experiment.

5.2 Hypothesis

Our hypothesis was that the HT (Head Tracking Input) cursors will help users finish the tasks faster and with less mouse clutches as compared to MSC cursors. Our second hypothesis was that MSC cursors will cause fewer clutches than the control cursor, but more clutches than Auto-Ht and Manual-HT cursors.

5.3 Hardware and Software Specifications

Hardware and software specifications were the same as in the first experiment with only one addition. To track each user's head movements we used the VICON system. Each user wore a hat which has three markers on it. We track the markers and use ray casting to determine where the user's head is pointing on the screen. Subjects again sat at a distance of approximately 86 inches from each column of the screen and used a wireless keyboard and wireless optical two-button mouse (Figure 5.1).



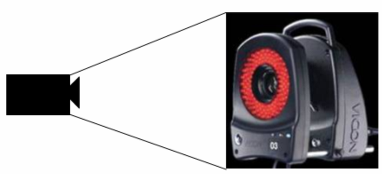


Figure 5.1. Experimental Set up. (a) Experimental floor plan of the display configuration with respect to the room and Vicon motion tracking infrared cameras. (b) Vicon cameras used in the experiments.

5.4 Design and Procedure

The design and procedure was the same as in the first experiment with two minor changes in the design and one minor change in the procedure. In the second experiment we compared six different interaction techniques. We also used the distance traveled by the mouse in pixels as another dependent variable in the second experiment. Moreover, in the first experiment each task was consisted of three clicks and we opened wallpapers on the screen after the first click, but in this experiment we opened images after the second click (Figure 5.2). We changed this because we did not want our users to forget about their target position while searching for their starting position in a noisy environment.

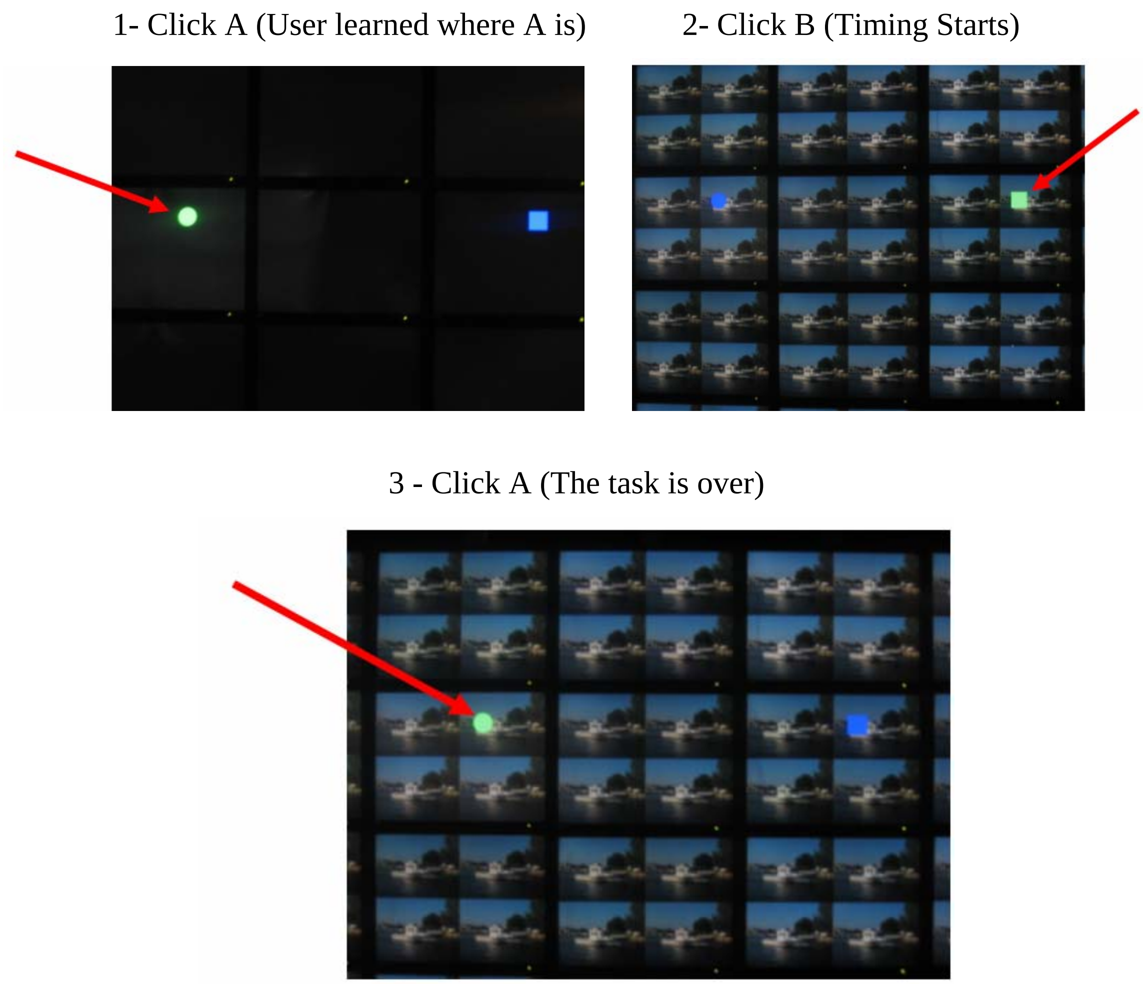


Figure 5.2: Experiment 2 background set up. (a) Black background before the first click. After the first click user learns the location of the target A. (b) The background image is turned on to simulate a typical desktop noise after the first click and before the second click. User needs to click the target B. (c) After the second click the timing starts, the cursor is warped to the center of the target B. Now, user needs to click the target A as soon as possible.

In this experiment, there were 12 participants (8 male, 4 female) ranging in ages from 22 to 40. None of the participants were color blind; all had normal or corrected normal vision. They were recruited by word of mouth. Only one of the participants used large display systems before to conduct scientific experiments.

5.5 Results

5.5.1 Movement Time

We performed all our statistical analysis in SAS's JMP using standard ANOVA and Chi Squared techniques after getting rid of outliers. We defined outliers by using Grabbs' Test. Grabbs' test detected only less than 2 % of the data as outliers.

5.5.1.1 The Effect of the Cursor Type on Movement Time

We again found a statistically significant effect of cursor technique on performance $F(5, 2326) = 748.51$, $p < 0.0001$. The result of the comparison of cursor types with the Tukey adjustment showed that HT-Only performed statistically the worst with the mean of 6.28 seconds. Control cursor was statistically significantly better than HT-Only with the mean of 3.23 seconds. The remaining cursor types performed statistically better than both control cursor and HT-Only cursor. DC2T's mean movement time was significantly less than Auto-HT's mean movement time, but there was no significant difference in movement times of Auto-HT, Manual-HT and DC4T. DC2T's mean movement time was not significantly different than Manual-HT and DC4T. The mean movement times of DC2T, DC4T, Manual-HT and Auto-HT varied between 2.53 seconds to 2.77 seconds (Figure 5.3).

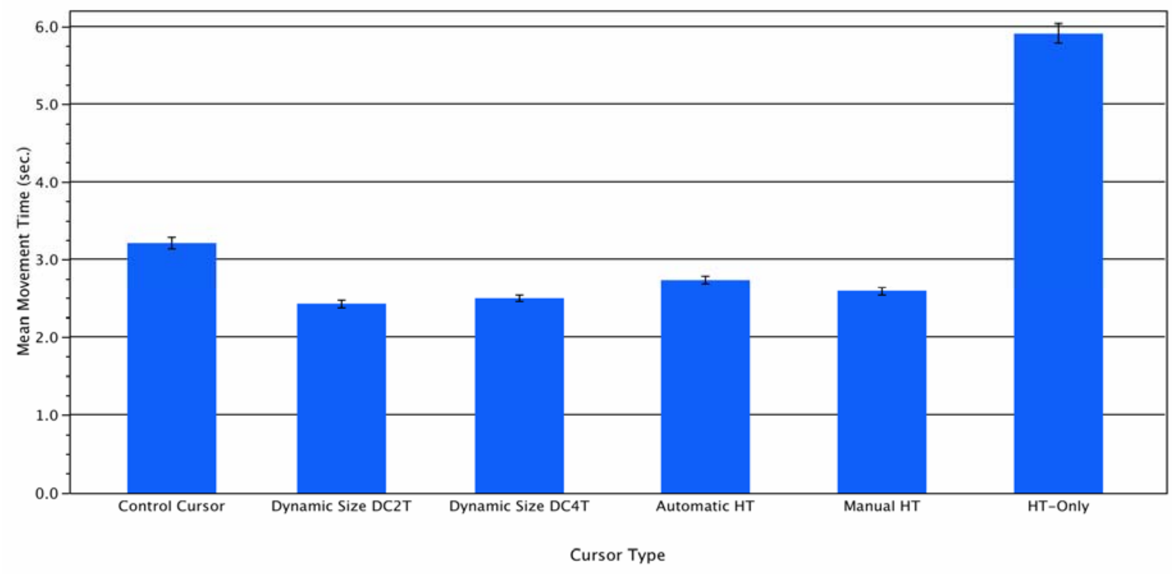


Figure 5.3: The mean movement times of all cursor types.

The mean movement time of the DC4T was 2.54 seconds, and the DC2T was 2.53 seconds. Button controlled ManualHT has the third lowest mean movement time which is 2.70 seconds and it was followed by AutoHT with the mean of 2.77 seconds.

5.5.1.2 The Effect of the Target Width on Movement Time

Target width significantly effected performance F (2, 2326) = 7.29, $p < 0.0001$. This is true for all the cursor types, since it took less time to click bigger targets than smaller ones (Figure 5.4).

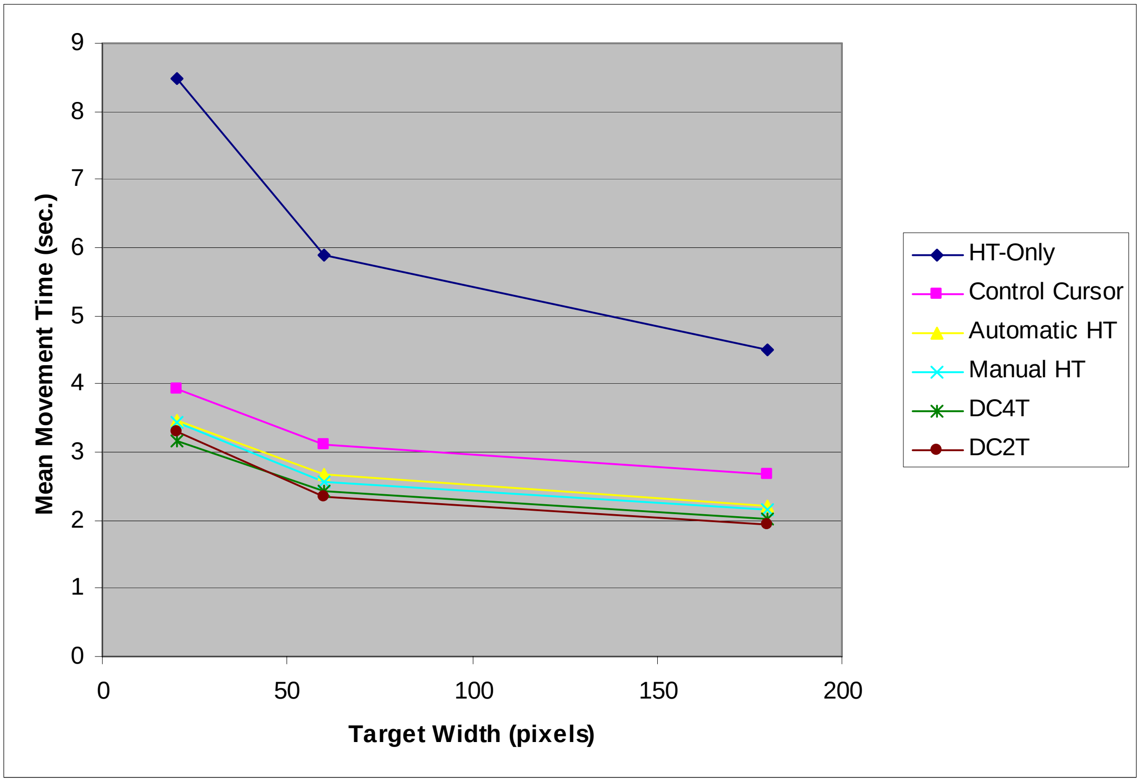


Figure 5.4: Mean performance time (sec) vs. target size (pixels).

Figure 5.4 shows that when target width gets larger, mean movement time decreases. This is true for all the cursor types. HT-Only is an outlier in the Figure 5.4 since its mean movement time is significantly higher than the mean movement times of all the other cursor types in this experiment. It can be seen that control cursor and HT-Only have higher mean movement times in each of the three different target widths. This performance difference between control cursor, HT-Only and all the other cursor types also led to a statistically significant performance difference between them. DC4T, DC2T, Manual HT, Automatic HT's mean movement times were significantly less than control cursor's and HT-Only's mean movement times.

5.5.1.3 The Effect of the Target Distance on Movement Time

Distance to targets (pointing amplitude) also significantly effected performance F (5, 2326) = 264.18, $p < 0.0001$. However, the amount of effect varied for each cursor type used, due to the significant interaction between cursor type and target distance $F(25, 2326) = 7.29$, $p < 0.0001$ (Figure 5.5).

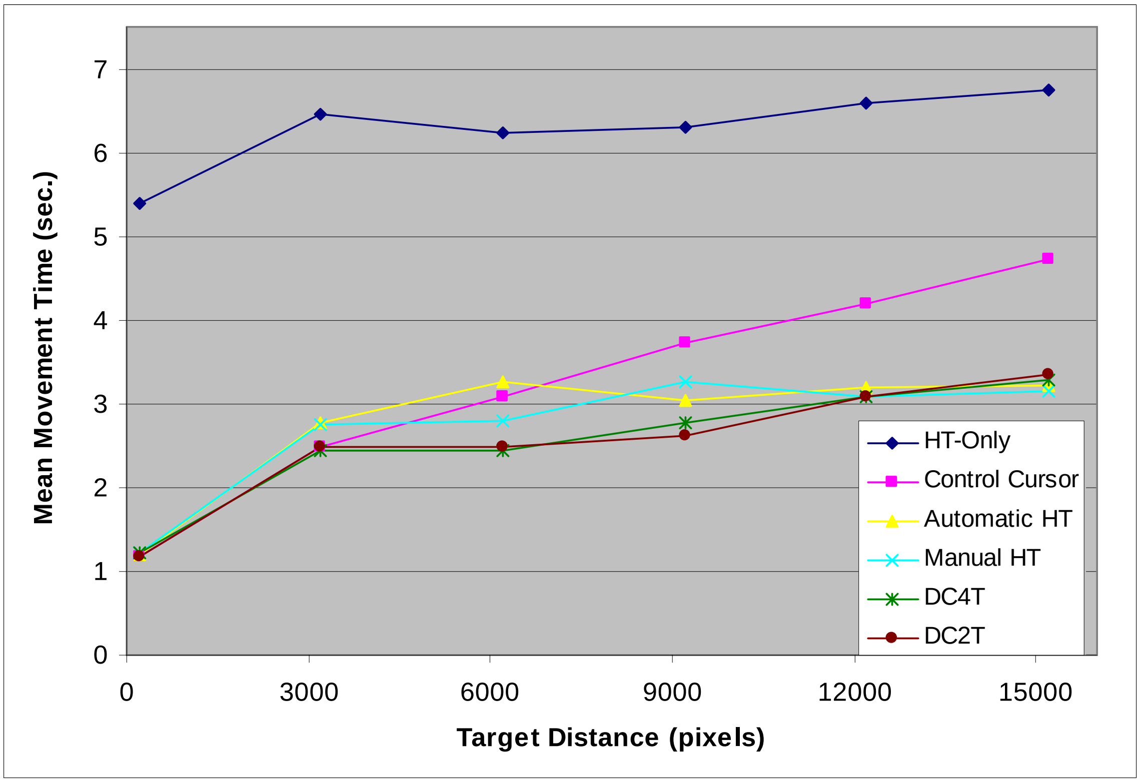


Figure 5.5: A chart showing how each cursor type's mean movement time varied according to distance to target.

Figure 5.5 shows how distance between targets affected performances of each cursor. Even though the overall mean movement times of DC2T, DC4T, AutoHT and ManualHT was significantly lower than the mean movement time of the control cursor. There was no significant difference between control cursor and DC2T, DC4T, AutoHT, ManualHT for the first two short range targets (Figure 5.5). The same distance and performance interaction was seen for control cursor, as in the first experiment, which causes the control cursor to perform increasingly worse and finally the second worst performance. To illustrate the point, the difference in performance between the first to last distance for the control cursor was 3.55 seconds, where the difference between the first and last distance for the dynamic size DC4T was 2.08 seconds.

Figure 5.5 also indicates that if we increase the target distance more than the maximum, the HT cursors will start performing better than the MSC cursors. But HT cursors failed to perform better than MSC cursors even on a 100 megapixel display.

The DC4T possesses more number of thresholds than the DC2T and it possesses the second lowest mean movement time which is very close the lowest movement time possessed by the DC2T. We investigated the performance of DC2T in the Experiment 1 results section. We now investigate how DC4T's mean movement time varies according to target size and target distance (Figure 5.6).

Three different colors in the Figure 5.6 depict the mean movement times of DC4T in three different target sizes. The graph clearly shows the indisputable effect of target size on mean movement time. In each of the distances DC4T's mean movement time decreases when the target size gets bigger. The gap between the mean movement times of target distances 200 and 3200 pixels gets smaller as the target distance increases. In the larger target distances DC4T manages to flatten the line between mean movement times for all target sizes. This flattening effect can easily be seen in the lines connecting mean movement times of the target distances 12200 and 15200 pixels for all target sizes.

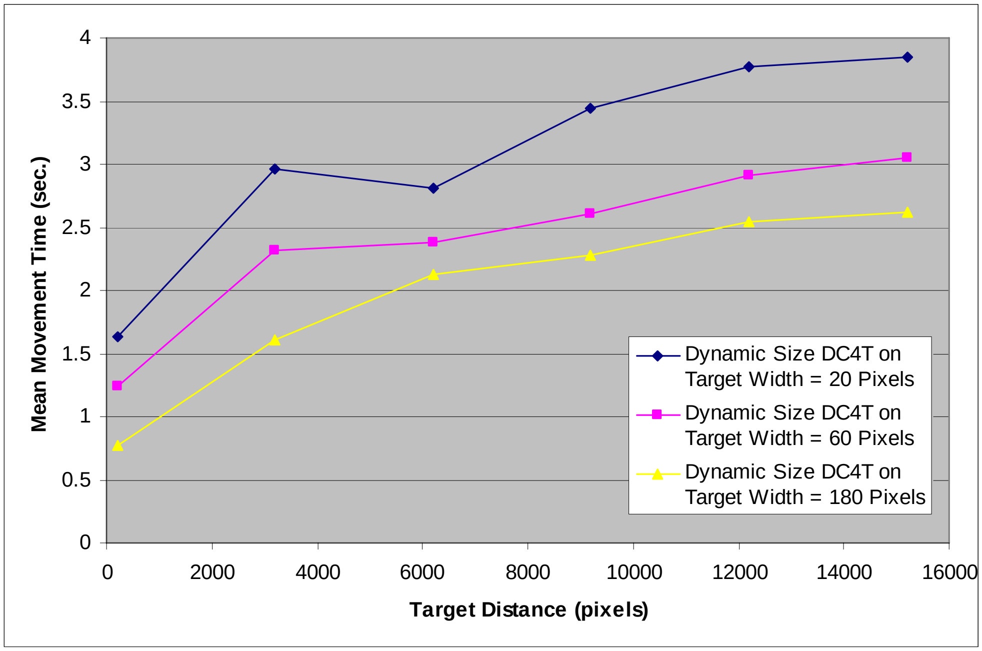


Figure 5.6: Mean Movement time (sec.) versus target distance (pixels) visualized for dynamic size DC4T in three different target sizes.

Index of difficulty (ID) is computed using Fitts' Law Index of Difficulty ID = $\log_2(A/W + 1)$ [44], where A is the amplitude, which is distance to the target, and W is the width of the target. ID is directly proportional with the pointing amplitude and inversely proportional with the target width. Figure 5.7 shows the relationship between index of difficulty (ID = $\log_2(A/W + 1)$ [44]) and the mean movement time (MT = a + b x ID [44]).

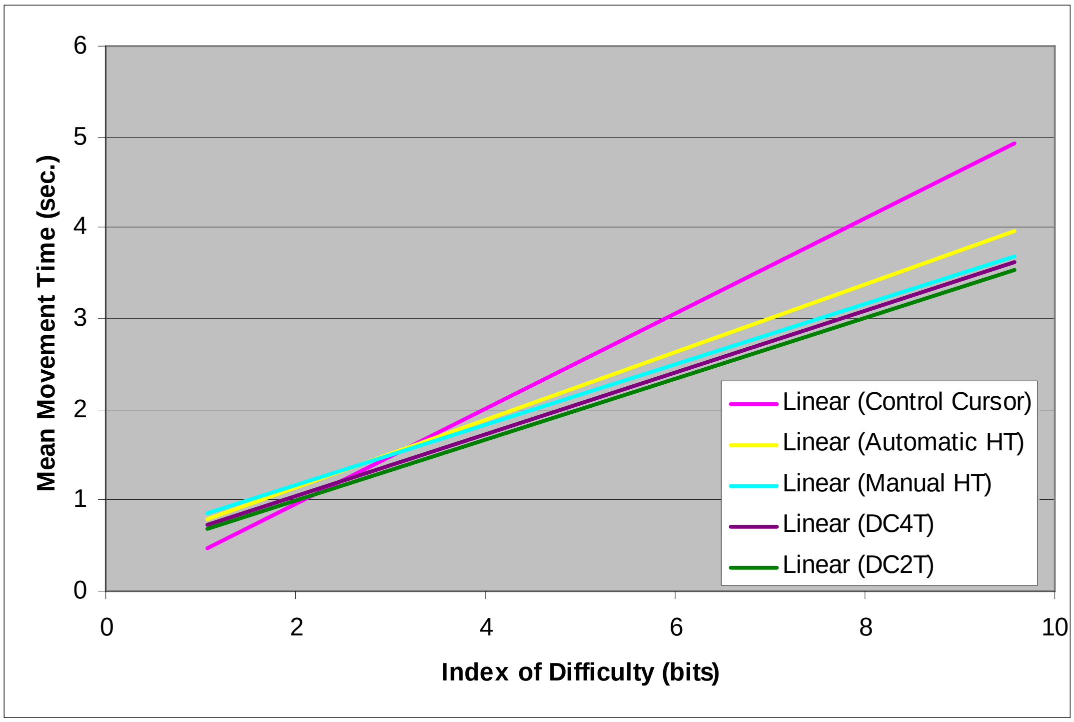


Figure 5.7: Linear Regression results for Mean Movement time (sec) vs. Index of Difficulty (bits).

HT-Only's mean movement time was tremendously higher than other cursor types' mean movement times in every target distances and targets used in the experiment. In order to better visualize the part where other cursor types' movement times are close to each other, we removed HT-Only from the Figure 5.7. It can be seen in the Figure 5.7 that control cursor's mean movement time was close to other cursor types' mean movement times in the beginning, but it had the most amount of variation in its performance and had the highest mean movement time for higher levels of index of difficulty. DC2T and DC4T had less variation in their performances than control cursor and automatic HT. Also, either DC2T or DC4T had closer mean movement times for all of index of difficulty levels. Automatic HT performed closer to DC2T and DC4T for small index of difficulties. Its performance decreased with respect to DC2T and DC4T as the index of difficulty increases. We believe that unintentional cursor movements caused by automatic HT caused this performance decrease.

Cursor Type	**a**	**b**	**Index of Performance**	**R^2**
Control Cursor	-0.082778	0.5238318	1.909009724	0.879702
DC2T	0.3280798	0.3348249	2.986635701	0.957462
DC4T	0.3705719	0.3387066	2.952407777	0.958057
Automatic HT	0.39013	0.3729447	2.681362679	0.927793
Manual HT	0.5085163	0.3325243	3.007299015	0.971903
HT-Only	Unknown	Unknown	Unknown	0.649084

Table 5.1: Linear regression results of fitting the data into Fitts' Model for the Experiment 2.

Linear regression results of fitting our data to the Fitts' model are depicted in the Table 5.1. It resulted in r^2 values more than 0.879, except HT-Only which did not fit to the Fitts' law and its r^2 value was 0.64. The indices of performances for each cursor, except HT-Only, ranged from 1.90 to 3.00 bps. HT-Only did not fit well to Fitts' Law since the "head jitter" problem dominated the performance as compared to time it takes to move the head position. Control cursor had the lowest IP. Dynamic size DC4T, DC2T and ManualHT possesses very close the indices of performances with corresponding IP values 2.95, 2.98, and 3.00.

5.5.2 Error Rate

Cursor type had a statistically significant effect on number of errors (clicking off-targets) F (5, 631) = 37.782, $p < 0.0001$. DC4T caused the lowest error rate with 31 total off-clicks and HT-Only caused the highest error rate with 341 error clicks (Figure 5.8). The second significant effect on number of errors was target size F (2, 634) = 26.591, $p < 0.0001$. Participants tend to make more errors on smaller sized targets as in the first experiment. Only the error rate of HT-Only was well beyond the <4% range in standard Fitts' law experiments. Error rate of other cursor types was in the <4% range.

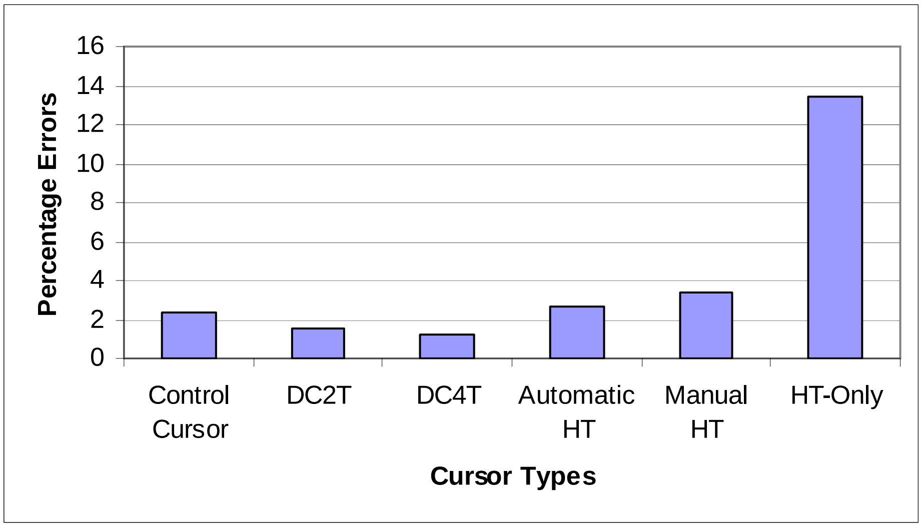


Figure 5.8: Percentage of errors for each cursor type.

5.5.3 Number of Pixels Traveled

We also recorded number of pixels that the mouse travels (in terms of the 1x base speed) during the recorded tasks in order to compare each cursor type in terms of hand fatigue that they might cause. Figure 5.9 shows that the number of pixels traveled by using mouse device in DC2T and DC4T was around seven times more than in AutoHT and ManualHT. This shows that using head tracking input in interaction with large displays could decrease hand fatigue and mouse clutches.

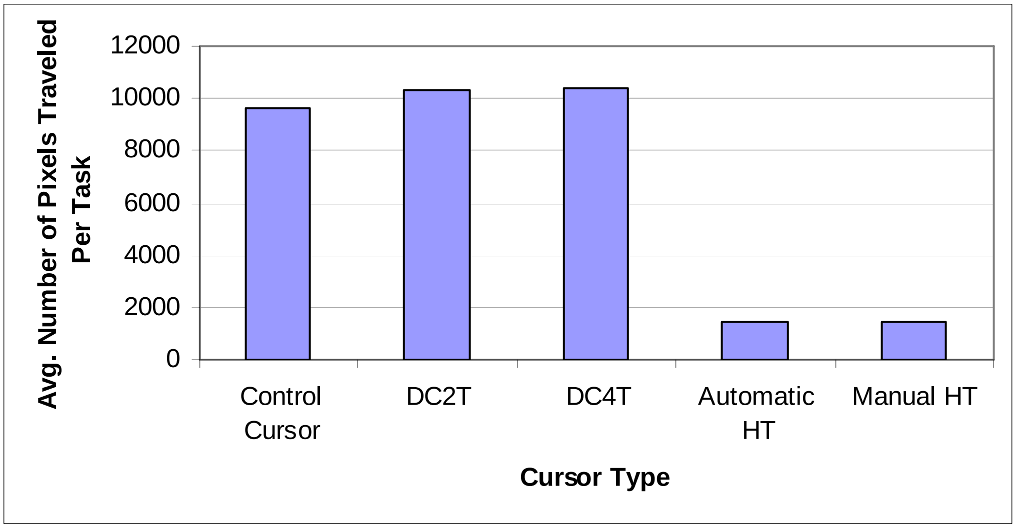


Figure 5.9: Number of pixels traveled per condition by using mouse device for each cursor type.

5.5.4 Survey Results

Although MSCs (DC2T, DC4T) possessed the lowest mean movement times only three of the participants have chosen MSC as the easiest to use design. Five of the participants have chosen ManualHT, while four of the participants have chosen AutoHT, two participants have chosen dynamic size DC4T and only one participant has chosen dynamic size DC2T as the easiest to use design.

Nine of the participants preferred HT cursor techniques, whereas only three of the participants chose dynamic cursors; although dynamic cursors have lower mean movement times than HT cursor techniques have. One of the participants who chose the dynamic cursor types stated that head tracking forces people to be conscious about their head movements at the same time their hand moves, and they stated that occasionally it is hard to accomplish that. On the other hand, one of participants who chose HT techniques stated that he no longer needed to follow the cursor, when he needed to see the cursor it was always near his focus of interest.

Participants who have chosen dynamic cursors preferred DC4T among DC2T, since they thought DC4T responded to their hand movement better in both short and long distances. ManualHT was preferred more than AutoHT, since it gives users more control over the cursor and unexpected hand-head multi-tasking problems no longer existed with it.

Moreover, participants sometimes were irritated about unexpected cursor movements due to their unintentional head moves while using AutoHT. None of the participants have chosen control cursor or HT-Only. They stated that it was very hard to move and follow the control cursor for long distances. They also criticized HT-Only about its accuracy. Due to the head 'jitter' problem, participants had problems accurately clicking the targets with HT-Only.

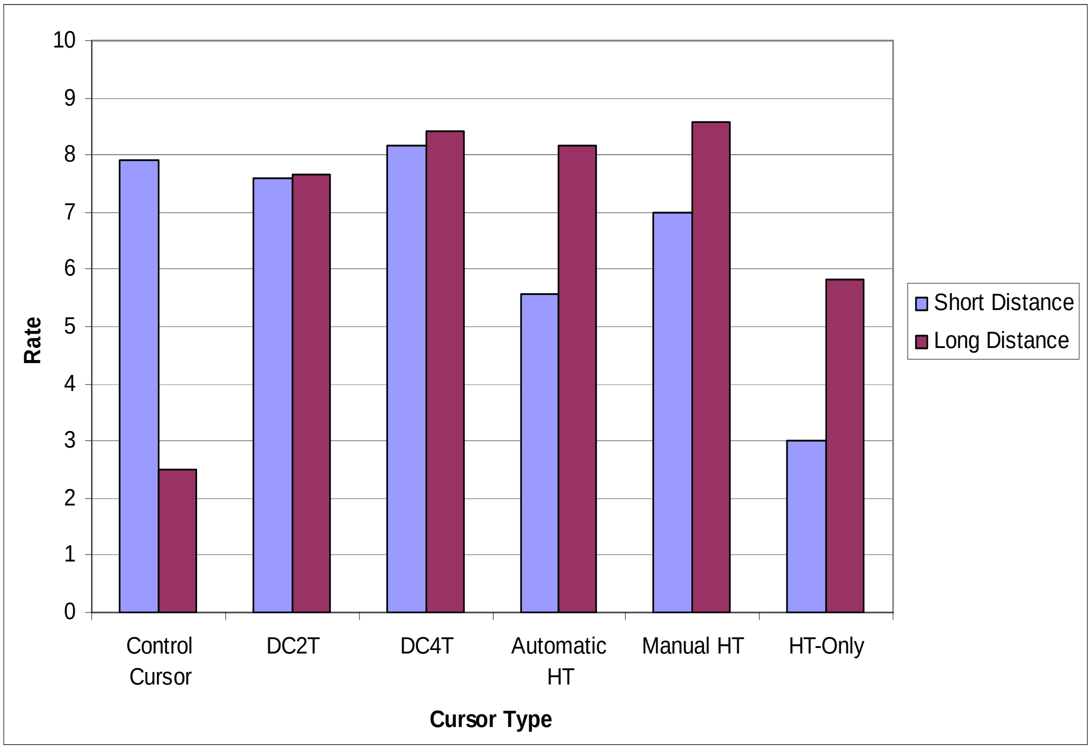


Figure 5.10: User rates of each cursor type on short and long distances.

The questionnaire after each cursor type block asked participants to rank that cursor type's ease of use in short and long distances in a scale of 10, where 0 is not effective at all, 10 is highly effective. Figure 5.10 shows that participants ranked DC2T and DC4T as highly effective on both short and long distances. Participants ranked DC2T and DC4T as highly effective as control cursor on short distances, which indicates that DC2T and DC4T did not cause overshooting problems for near targets. Even though participants ranked HT cursor as highly effective as dynamic cursors on long distances, they ranked HT cursors lower than dynamic cursors on short distances. Our belief is that usability problems of HT cursors caused this ranking difference on short distances. As indicated by our participants, automatic HT had the problem of unintentional movements, while manual HT required the usage of both hands to control the cursor, which increases the stress on user's motor skills.

5.6 Discussion

In the light of this experiment, we were able to identify that the mouse can still be as effective as absolute pointing techniques and are thus scalable to a very large, high resolution display. It is interesting that DC4T possessed a lower index of performance than DC2T. We could not detect any significant difference between DC4T and DC2T. We believe that even though we increased the number of C-D ratio levels, we came to a limit in improving the pointing performance with the C-D ratio adaptation. Therefore, we believe that more usability studies should be conducted to define the best number of C-D ratio levels and C-D ratio values of each level trying to further improve the pointing performance.

AutoHT's mean movement time was higher than ManualHT's mean movement time. Users stated that they liked the idea of having the cursor in their area of focus all the time, but that led to some usability problems like unnecessary cursor movements during the unintentional head movements. As an illustration, some of the subjects placed the cursor on top of the starting position and tried to visually search for the target position before clicking the starting target. When they move their heads to search the target position, the cursor moved according to their head movements and thus, they needed to place the cursor again on top of the starting position to click it. This problem of unnecessary cursor movements irritated users.

Although MSCs' mean movement times were lower than HT cursors' mean movement times; 9 out of 12 participants have chosen HT cursors as the easiest to use design. But the reason for choosing HT-cursors among MSCs remained unclear, since the reason behind choosing HT cursors might be the novelty of the technique instead of the performance of the technique.

HT-Only's mean movement time was significantly the worst among the cursors we compared in this experiment. Even though we tried to ease the pointing task by zooming the target area by 4x in the first click, users stated many problems in clicking the targets. This was especially the case with the smallest targets which had a 20 pixel width. The "head jitter" problem caused problems in both placing the focus area on top of a target and also clicking the target. Using only head tracking input for pointing tasks failed to provide comparably fine results.

Lastly, HT cursors managed to minimize the mouse clutching problem. Moreover, the mean movement time – target distance graph (Figure 5.5) indicates that if we increase the target distance more the HT cursors will start performing better than the MSC cursors. Despite HT cursors' advantage on less hand fatigue, it is interesting that MSC cursors had lower mean movement times than HT cursors. Users spent less time on moving the mouse and thus they were less irritated by the mouse clutching problem. However, this did not help to improve the pointing performance even in a 100 megapixel display.

Both MSC and HT cursors still have their own usability problems and it is hard to state that one should be preferred among another. Although users preferred to use HT cursors rather than MSCs, mean movement times of the MSCs were better than HT cursors' mean movement times. As in the scenario under investigation, users will be interacting with large displays for a long period of time, trying to acquire many targets. Pointing performance (mean movement times) will be more important than users' preferences (user ranks) in the scenario under investigation, because using MSC concept will bring time savings over long period of time. Therefore, we believe that due to low-cost rates of installing MSC on computers, it is the best choice for large, high resolution displays.

Chapter 6. Conclusions

6.1 Summary

Our main motivation behind these studies was to test the scalability of the mouse device to large, high resolution displays. To be able to effectively test its scalability we implemented different cursor designs which, we think, would perform better on large displays as compared to a control cursor. Indeed, some of the cursor designs that we have tested within two controlled experiments eased long distance movements by decreasing the number of mouse clutches to less than one per task and by increasing cursor visibility without obscuring targets.

We aimed to find out better cursor designs in our first experiment. We then compared best of these designs with the coarse absolute pointing techniques. We also coupled MSC concept with the coarse absolute pointing techniques to attempt to further improve the pointing performance. In the second experiment coarse absolute pointing techniques also served as a benchmark to test the scalability of the mouse device.

The results of our study indicate that the multi-scale cursor (MSC) design improves the scalability of the mouse device to large, high resolution displays because of the following set of evidences:

1. MSC design reduced the number of mouse clutching to less than one per task which provides similar usage behavior to that of small screens, even in a 100 megapixel display.
2. As the target distance gets larger the performance deterioration between consecutive distances gets smaller for multi-scale cursors more than for other cursors. This indicates that, with the multi-scale cursor implementation, the mouse device can scale up to large screen sizes better than HT and control cursors. Also, best of the multi-scale cursors had a higher index of performance [79] than both control and HT cursors. This also partially indicates that multi-scale cursor's pointing performance was better than control and HT cursors' pointing performance overall.

3. For short distance movements, for fine controls, multi-scale cursors performed similar to the typical Linux default cursor. This means that multi-scale cursors provide similar usage behaviors on small screens for small distances, which helps to overcome target overshooting problems, eases target acquisition and increases pointing accuracy, at the same time facilitating large distance movements.
4. For long distance movements, integrating the head tracking input with the mouse device input in HT cursors served as benchmarks for evaluating multi-scale cursor types. HT cursors were not faster than the best of multi-scale cursors. This validates the high-performance of the MSC on long distances and indicates that MSC is scalable to long target distances.

Therefore, the mouse can still be an effective interaction technique in a large personal workspace. The mouse device's pointing performance was improved with minor implementation modifications. Our study helped us to identify a set of design guidelines while defining the best MSC implementation.

Results of our first experiment showed that the dynamic size cursor helped users to finish the tasks faster. It increased the cursor visibility and gave visual feedback about cursor speed. As participants commented, having a bigger cursor while moving the cursor faster helped them to follow the cursor with their peripheral vision. We believe that dynamic size cursors will not only help users perform classical Fitts' law tasks faster, but also will help users to locate the cursor when they lose the cursor in their regular desktop and application level usage. Therefore, we suggest that the dynamic visualization of cursor size should be used over the standard cursor settings in large, high-resolution displays.

Dynamically changing C-D ratio according to the device movement speed resulted in significantly better performance. The control cursor's performance continued to worsen rapidly with the increase in the target distance. But with the dynamic size DC2T, the increase in mean movement time from short distances to long distances targets took from about 1 second on 200 pixels distance to about 3 seconds on 15200 pixels. That is a 3x increase in mean movement time for a 76x increase in distance. It also maintained as high performance as the control cursor in short distances. Being able to ease long distance movements by decreasing both the number of clutches and the time it takes to reach a distant target indicated the scalability of the mouse device to large displays.

We failed to detect any significant difference between performances of DC2T and DC4T in the second experiment. This result showed us that we came to a limit with dynamic size and speed cursor implementation. Thus, before trying to further improve the performance of DC4T, we believe that more usability studies have to be conducted in order to define the best number of thresholds, speed values and C-D ratio levels for a dynamic speed cursor.

In terms of configuring dynamic speed part of the multi-scale cursor, we found that having more than 2x jumps between threshold speeds decreases pointing performance. Moreover, for our display, we did not see any advantage of decreasing jumps between threshold levels to less than 2x. We also found that users preferred using automatically changing cursor speed version over manually changing cursor speed version.

The goal of combining mouse and head tracking input was to improve the overall user performance. But we failed to succeed in that, because HT cursors' mean movement time was even higher than MSCs' mean movement times. On the other hand, the amount of pixels traveled with the HT cursors was about seven times less than the amount of pixels traveled with the MSCs. HT cursors helped us to solve one usability problem of the mouse in large screens, clutching, but they caused other usability problems. While using AutoHT, some of the participants moved their heads disregarding the pointing task. This head movement caused a cursor movement, which irritated participants. While using ManualHT, users expressed frustration about being conscious of their head position, when to press the space button, and when to use the cursor for fine control.

As both HT cursor and MSC each have usability problems, it is hard to differentiate them in terms of usability; but due to lower cost rate and ease of installing MSC, we think it is the best choice for large high-resolution workspaces for the time being. Current operating systems, such as Windows XP [52], are already using the dynamic speed behavior of the MSC concept. For large, high resolution displays, MSC concept can be easily integrated with the real-world operating systems, given that we can reach their source code, by changing the speed values according to the display size. Adding the dynamic size behavior of the MSC concept will require loading larger size cursors on the screen when the device is moved faster. Effects of loading a larger cursor image should be investigated to increase the effectiveness of the dynamic size behavior of the MSC concept .

6.2 Contributions

This study helps interaction designers to better understand the scalability limits of the mouse device into large, high resolution displays. The study also helps interaction designers to better understand different design choices, while trying to improve the mouse implementation as we move from the small desktop screens to the large, high resolution displays. Specific contributions of this work are listed below.

- We discovered how the mouse device scales up to a 100 megapixel display.
- We present a dynamic size and dynamic speed interaction technique for large, high resolution displays with less mouse clutching and greater cursor visibility.
- We provide a well-designed cursor implementation to be used as a baseline for comparison against other innovative large-display interaction techniques in studies related to the Human-Computer Interaction.
- We prove that pointing performance can still be significantly improved without changing pointing amplitude or target width, but by just using the power of visualization to dynamically change cursor size.
- We prove that dynamically changing C-D ratio according to the device movement speed can significantly improve the pointing performance in large, high resolution displays.
- We show that the MSC mouse is as efficient as coarse absolute pointing techniques for long distance movements, and as effective as the control mouse for short distance movements.

Therefore, we believe that the MSC mouse is an effective way of interacting with large, high resolution displays.

6.3 Future Work

We frequently deal with the problem of losing the cursor, while using a large screen in an office set up for daily usage. Thus, we can compare future designs in a task in which users do not know the starting point of the cursor. In these experiments users will need to find the cursor first, and then they will try to click the target. We believe that dynamic size cursors will also have an advantage over static size cursors under this daily usage scenario.

Monitoring mouse device usage, counting the number of pixels that the mouse device is moved to acquire each target, in an office workspace usage behavior on large displays would give us a better insight about how to optimize mouse interaction on large displays. If the frequency of the movements for each target distance gets clear, we would have a better idea about what parts of the design space we should focus more on. We can also implement our MSC concept on a large display and monitor user performance with both the MSC and the default cursor.

There is a high rate of performance decrease between 200 pixel and 3200 pixel distance values in the performances of MSCs. We want to find out the reasons behind that and improve the design accordingly. The reason for that might be related to using discrete C-D mapping functions instead of a continuous C-D mapping functions. This causes speed jumps between acceleration levels. Making the speed transition between C-D ratio levels smoother might further improve multi-scale cursor design.

We can also improve acceleration levels of the MSC concept to achieve a continuous speed transition. In the second experiment, we increased the number of thresholds, but failed to gain any performance improvement with that. This showed that increasing number of different speed values in C-D ratio adaptation will stop providing performance improvements after some point. Even if we believe that the result would be the same while using continuous C-D mapping, we can compare the continuous C-D mapping with the current best MSC concept. While doing that we can include more C-D ratio levels and decrease the lowest acceleration level at the same time increasing the highest acceleration level.

Head tracking input served as a benchmark test in the second experiment to evaluate the performance of the MSC concept. We also want to compare MSC concept with different alternatives for pointing to large, high resolution screens to validate its effectiveness and usability. We can also improve the HT cursors such as by visualizing region of focus and compare them with MSC concept in another study.

Lastly, some users might want to be able walk and stand in front of the display without using a chair. We can also test the MSC concept on different usage scenarios to have a better idea about the effectivity of the MSC concept on different usage scenarios. We can use a gyro-mouse and/or presentation mouse to test the performance of the MSC concept on these kinds of scenarios. There are many other alternative techniques to interact with large displays on a moveable user position scenario such as using a laser pointer and hand based absolute pointing. We can compare the best of the MSC concept with other well-designed techniques in a scenario where users are free to stand up, walk and change their positions whenever they want.

References

1.Abello, J., Gansner, E.R., Koutsofios, E., and North, S. Large Scale Network Visualization*.* *Computer Graphics*, *33*,3 (1999), 13-15.

2. Arnault, L.Y., Greenstein, J.S. Is display/control gain a useful metric for optimizing an interface?, Human Factors, v.32 n.6, p.651-663, Dec. 1990

3. Ashdown, M., Oka, K., and Sato, Y. Combining head tracking and mouse input for a gui on multiple monitors. *Ext. Abstracts CHI 2005*, ACM Press (2005) 1188–1191.

4. Ball, R. and North, C. An analysis of user behavior on high-resolution tiled displays. *Proc. Interact 2005*, 350-363.

5. Ball, R. and North, C. Effects of Tiled High-Resolution Display on Basic Visualization and Navigation Tasks. In *Extended Abstracts of Human Factors in Computing Systems (CHI '05)*. 2005. Portland, OR, USA: ACM, p. 1196-1199.

6. Ball, R. and North, C. Realization of Embodied Interaction for Visual Analytics through Large Displays. *Computers & Graphics 2007*. ACM Press (2007). 31:3, 380-400.

7. Ball, R., Varghese, M., Sabri,A., Cox, E.D., Fierer, C., Peterson, M., Carstensen, B., and North, C. Evaluating the Benefits of Tiled Displays for Navigating Maps. In *IASTED International Conference on Human-Computer Interaction (IASTED-HCI 2005)*, ACM Press (2005), 66-71.

8. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., and Zierlinger, A. Drag-and-pop and drag-and-pick: techniques for accessing remote screen content on touch- and pen-operated systems. In *Proc. Interact 2003*, IOS (2003), 57-64.

9. Baudisch, P., Cutrell, E., Hinckley, K. and Gruen, R. Mouse Ether: Accelerating the Acquisition of Targets Across Multi-Monitor Displays. *Ext. Abstracts CHI 2005*, ACM Press (2004), 1379-1382.

10.Baudisch, P., Cutrell, E., and Robertson, G. High-density cursor: A visualization technique that helps users keep track of fast-moving mouse cursors. *Proc. Interact 2003*, 236-243.

11.Benko, H., and Feiner, S. Multi-monitor mouse. *Ext. Abstracts CHI 2005*, ACM Press (2005), 1208-1211.

12.Bier, E.A. and Stone, M.C. 1986. Snap-dragging. In *Proceedings of ACM SIGGRAPH 1986*. 233–240.

13.Blanch, R., Guiard, Y. Beaudouin-Lafon, M. Semantic pointing: improving target acquisition with control-display ratio adaptation, Proceedings of the SIGCHI conference on Human factors in computing systems, p.519-526, April 24-29, 2004, Vienna, Austria

14.Boff, K.R. and Lincoln, J.E. Engineering data compendium: Human perception and performance, Harry G. Armstrong Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, 1988.

15.Bowman, D., Kruijff, E., LaViola, J. and et al. *3D User Interfaces: Theory and Practice*. Pearson Education, 2005.

16.Buck, L. (1980). Motor Performance in relation to control-display gain and target width. Ergonomics, 32, 579-589.

17.Card, S. K., English, W. K., & Burr, B. J. (1978). Evaluation of mouse,rate-controlled isometric joystick, step keys, and text keys for text selection of a CRT. Ergonomics, 21, 601-613.

18.Cheng, K., Pulo, K. Direct interaction with large-scale display systems using infrared laser tracking devices. In *Proc. InfoVis 2003,* ACM Press (2003), 67-74.

19.Chromium. http://chromium.sourceforge.net/.

20.Cockburn, A. and Firth, A. Improving the acquisition of small targets. In Proc. HCI 2003, pages 181--196, 2003.

21.Czerwinski, M., Robertson, G., Meyers, B., Smith, G., Robbins, D., and Tan, D. Large Display Research Overview. In *Proc. CHI 2006,* ACM Press (2006), 69-74.

22.Czerwinski, M., Smith, G., Regan, T., Meyers, B., Robertson, G., and Starkweather, G. Toward characterizing the productivity benefits of very large displays. In *Proc. Interact 2003*, IOS (2003), 9-16.

23.Czerwinski, M., Tan, D., and Robertson, G. Women Take a Wider View. In *Proc. CHI 2002,* ACM Press (2002), 195-202.

24.Dulberg, M.S., Amant, R.S., and Zettlemoyer, L.S. An Imprecise Mouse Gesture for the Fast Activation of Controls. In *Proc. INTERACT 1999*, IOS Press (1999), 375-382.

25.Elrod, S., Gold, Goldberg, Halasz, Janssen, Lee, McCall, Pedersen, Pier, Tang and Welch. Liveboard: A Large Interactive Display Supporting Group Meetings, Presentations, and Remote Collaboration. In *Proc. CHI 1992,* ACM Press (1992), 599-607.

26.English, W.K., Engelbart, D.C., and Berman, M.L., 1967, Display-selection techniques for text manipulation. *IEEE transactions on Human factors in Electronics,* HFE-8, 1, 21-31.

27.Fitts, P.M. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology 1954*, 47:381-391.

28.Forlines, C., Vogel, D., and Balakrishnan, R. 2006. HybridPointing:fluid switching between absolute and relative pointing with a directinput device. In Proceedings of the 19th Annual ACM Symposium on Userinterface Software and Technology (Montreux, Switzerland, October 15 - 18, 2006). UIST '06. ACM Press, New York, NY, 211-220.

29.Gibbs, C.B. Controller design: Interactions of controlling limbs, time-lags, and gains in positional and velocity systems. Ergonomics, 5:385-402, 1962.

30.Global FPD Industry 2003 Executive Summary. [web site] 2003 [cited 2006 5/24/2006]; Available from: http://www.usdc.org/technical/USDCroadmap_ExcutiveSummary.htm.

31.Glut – OpenGL utility toolkit. http://www.opengl.org/resources/libraries/glut/

32.Grossman, T., and Balakrishnan, R. The bubble cursor: Enhancing target acquisition by dynamic resizing of the cursors activation area. In *Proc. CHI 2005,* ACM Press (2005), 281-290.

33. Grudin, J. Partitioning digital worlds: focal and peripheral awareness in multiple monitor use. In *Proc. CHI 2001*, ACM Press (2001), 458-465.

34. Guiard, Y., Blanch, R., and Beaudouin-Lafon, M. 2004. Object pointing: A complement to bitmap pointing in GUIs. In *Proceedings of Graphics Interface (GI'04)*. 9–16.

35. Harada, S., Landay, J.A., Malkin, J., Li, X., and Bilmes, J.A. The vocal joystick: evaluation of voice-based cursor control techniques. In *Proc. SIGACCESS 2006*. ACM Press (2006), 197-204.

36. Hess, R.A. Nonadjectival rating scales in human response experiments. Human Factors, 13(3):275-280, 1973.

37. Hinckley, K., Ramos, G., Guimbretiere, F., Baudisch, P., and Smith, M. Stitching: pen gestures that span multiple displays. *Proc. AVI 2004*, ACM Pre 23-31.

38. Jellinek, H.D., Card S. K. Powermice and user performance, Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people, p.213-220, April 01-05, 1990, Seattle, Washington, United States

39. Jenkins, W.L., and Connor, M. 8., 1949,Some design factors in making settings on a linear scale. *Journal of Applied Psychology,* 33, 395-409.

40. Jock D. Mackinlay and J. Heer. Wideband displays: Mitigating multiple monitor seams. In *Proc. CHI 2004*, ACM Press (2004), 1521–1524.

41. Johanson, B., Fox, A. and Winograd, T. The Interactive Workspaces Project: Experienceswith Ubiquitous Computing Rooms. In *Proc. IEEE Pervasive Computing 2002*, IEEE Educational Activities Department (2002), 67-74.

42. Kabbash, P. and Buxton,W. A. S. 1995. The "prince" technique: Fitts' law and selection using area cursors. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI'95)*. 273–279.

43. Keyson, D.V. Dynamic cursor gain and tactual feedback in the capture of cursor movements. Ergonomics, 12:1287--1298, 1997.

44. Kumar, M., Paepcke, A., and Winograd, T. EyePoint: Practical Pointing and Selection Using Gaze and Keyboard. In *Proceedings of CHI 2007*, ACM Press (2007), 421-430.

45. Lombardo, J., Burkom, H., Elbert, E., Magruder, S., Lewis, S.H., Loschen, W., Sari, J., Sniegoski, C., Wojcik, R., and Pavlin, J. A Systems Overview of the Electronic Surveillance System for the Early Notification of Community-Based Epidemics (ESSENCE II). *Journal of Urban Health*, *80*,(2003), 32-42.

46. MacKenzie, I.S. 1989. A note on the information-theoretic basis for Fitts' law. *J. Motor Behav.21*, 3, 323–330.

47. MacKenzie, I.S. Fitts' law as a research and design tool in human computer interaction. *Human Factors in Computing System*, ACM Press (1992), 219-226.

48. MacKenzie, I.S. and Riddersma, S. Effects of output display and control-display gain on human performance in interactive systems. Behaviour & Information Technology, 13:328--337, 1994.

49. McCormick, E.J., 1976, *Human Factors in Engineering and Design* (New York: McGrawHill).

50.McGuffin, M.J., and Balakrishnan, R. 2002. Acquisition of expanding targets. In *Proceedings ofACM Conference on Human Factors in Computing Systems (CHI'02)*. 57–64.

51.McGuffin, M.J, Balakrishnan, R. Fitts' law and expanding targets: Experimental studies and designs for user interfaces, ACM Transactions on Computer-Human Interaction (TOCHI), v.12 n.4, p.388-422, December 2005.

52.Microsoft Windows XP pointer ballistics: http://www.microsoft.com/whdc/device/input/pointer-bal.mspx

53.Munch, S. and Dillmann, R. 1997. Haptic output in multimodal user interfaces. In *Proceedingsof ACM International Conference on Intelligent User Interfaces (IUI'97)*. 105–112.

54.Myers, B.A., Bhatnagar, R., Nichols, J., Peck, C.H., Kong, D., Miller, R., and Long, A.C. Interacting at a distance: measuring the performance of laser pointers and other devices. In *Proc. CHI 2002*, ACM Press (2002), 33-40.

55.Nacenta, M.A., Sallam, S., Champoux, B., Subramanian, S., and Gutwin, C. Perspective cursor: perspective-based interaction for multi-display environments. In Proc. CHI 2006, ACM Press (2006), 289-298.

56.Ni, T., Schmidt, G., Staadt, O., Livingston, M., Ball, R., May, R. A Survey of Large High-Resolution Display Technologies, Techniques, and Applications, *IEEE Virtual Reality 2006*, pg. 223-234, March 2006.

57.Nickel, K., Stiefelhage, R. Pointing gesture recognition on 3D tracking of face, hands and head orientation. *Proc. ICMI 2003*.

58.Oakley, I., Adams, A., Brewster, S. A., and Gray, P. D. Guidelines for the design of haptic widgets. In *People and Computers XVI: British Computer Society Conference on Human Computer Interaction (HCI'02)*. ACM Press 2002.

59.Oakley, I., Brewster S. A., and Gray, P. D. 2001. Solving multi-target haptic problems in menu interaction. In *Extended Abstracts of ACM Conference on Human Factors in Computing Systems (CHI'01)*. 357–358.

60.Olsen, D.R., and Nielsen, T. Laser pointer interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 17 – 22, 2001.

61.OpenGL. http://www.opengl.org.

62.Pedersen, E.L., McCall, K., Moran, T.P., and Halasz, F.G. Tivoli: An electronic whiteboard for informal workgroup meetings. In *Proc. CHI 1993*, ACM Press (1993), 391-398.

63.Robertson, G.G., Czerwinski, M., Baudisch, P., Meyers, B., Robbins, D., Smith, G., and Tan, D. The Large-Display user experience. *IEEECG&A*, *25*, *4*, (2005), 44-51.

64.Sabri, A., Ball, R., Fabian, A., Bhatia, S., North, C. High-resolution gaming: Interfaces, notifications, and the user experience. *Proc. Interact 2007*, Elvis Science Inc. (2007), 151-166.

65.Sandstrom, T.A., Henze, C., and Levit, C. The Hyperwall. In Proc. Coordinated and Multiple Views in Exploratory Visualization (CMV), IEEE (2003), 124-133.

66. Saraiya, P., North, C., and Duca, K. Evaluation of Microarray Visualization Tools for Biological Insight. In *Proc. IEEE Symposium on Information Visualization*, IEEE (2004), 1-8.

67. Shupp, L., Ball, R., Yost, R., Booker, J., North, C. Evaluation of viewport size and curvature of large, high-resolution displays, *Proc. Graphics Interface 2006*, ACM Press (2006), 123-130.

68. Sibert, L.E., and Jacob, R.J.K. Evaluation of eye gaze interaction. In Proc. CHI 2000, ACM Press (2000), 281-288.

69. Simmons, T. What's the optimum computer display size?, *Ergonomics in Design* (2001), pp. 19–25.

70. Streitz, N.A., et al., i-LAND: An Interactive Landscape for Creativity and Innovation, In Proc. *CHI 1999*, ACM Press (1999), 120-127.

71. Sutherland, I.E. 1963. Sketchpad: A man-machine graphical communication system. In *Proceedingsof AFIPS Spring Joint Computer Conference*. 328–346.

72. Swaminathan, K., Sato, S. Interaction design for large displays, interactions, v.4 n.1, p.15-24, Jan./Feb. 1997

73. Trankle, U. and Deutschmann, D. (1991). Factors influencing speed and precision or cursor positioning using a mouse. Ergonomics, 34, 2, 161-174.

74. Tan, D., Gergle, D., Scupelli, P.G., Pausch, R. Physically large displays improve path integration in 3D virtual navigation tasks. In *Proc. CHI 2004*. ACM Press (2004), 439 – 446.

75. Tan, D., Gergle D., Scupelli, P.G., Pausch, R. With similar visual angles, larger display improve spatial performance. In *Proc. CHI 2003*, ACM Press (2003), 217–224.

76. Tan, D., Stefanucci, J.K., Proffitt, D., and Pausch, R. The infocockpit: providing location and place to aid human memory. In *Proc. PUI 2001*, 1-4.

77. Wei, B., Silva, C., Koutsofios, E., Krishnan, S., and North, S. Visualization Research with Large Displays. *IEEE Computer Graphcs and Applications*, *20*,4 (2000), 50-54.

78. Worden, A., Walker, N., Bharat, K., and Hudson, S. 1997. Making computers easier for older adults to use: Area cursors and sticky icons. In *Proceedings of ACMConference on Human Factorsin Computing Systems (CHI'97)*. 266–271.

79. Zhai, S. 2002. On the validity of throughput as a characteristic of computer input. Tech. Rep. IBM Research Report RJ 10253, IBM Almaden Research Center, San Jose, CA.

80. Zhai, S., Conversy, S., Beaudouin-Lafon, M., and Guiard, Y. 2003. Human on-line response to target expansion. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI'03)*. 177–184.

81. Zhai, S., Morimoto, C. and Ihde, S. Manual and Gaze Input Cascaded (MAGIC) Pointing. In *Proc. CHI 1999*, ACM Press (1999), 246–253.

Appendix A: Experiment 1

A.1 Informed Consent for Participant of Investigative Project

I. THE PURPOSE OF THIS RESEARCH/PROJECT

You are invited to participate in a study of interaction in large high resolution display. This research studies the different cursor types for large high resolution screens. This study involves experimentation for the purpose of evaluating and improving the user interaction techniques for large high resolution displays.

II. PROCEDURES

You will be asked to perform a set of tasks using a different cursor types. These tasks consist of clicking objects by using mouse. Your role in these tasks is to finish click the target object as fast as possible. You are helping us to evaluate our system. All information that you help us attain will remain anonymous. The time you take to do each task and other aspects of your interaction with the system will be measured. You may be asked questions during and after the evaluation, in order to clarify our understanding of your evaluation.

You may also be asked to fill out a questionnaire relating to your background with such systems.

The session will last less than two hours. The tasks are not very tiring, but you are welcome to take rest breaks as needed. One scheduled rest break will be given to you about half-way through the experiment. You may also terminate your participation at any time, for any reason.

You will be given full instructions before every task. Because we are measuring a range of abilities, some tasks will be easy and others difficult. It is important that you understand the instructions before beginning each task. If anything is unclear, be sure to ask us questions.

III. RISKS

The proposed experiments are straightforward tests of performance using large high resolution displays, and input devices. Participation involves sitting in front of the display and performing simple clicking tasks. The physical components of these tasks are not stressful, and include head turning and pointing with mouse. All light and sound intensities are well within normal ranges. There are no known mental risks. The risks associated with participation in this study is minimal.

IV. BENEFITS OF THIS PROJECT

Your participation in this project will provide information that may be used to improve the design of new mouse techniques for large high resolution displays. No guarantee of benefits has been made to encourage you to participate.

You are requested to refrain from discussing the evaluation with other people who might be in the candidate pool from which other participants might be drawn.

V. EXTENT OF ANONYMITY AND CONFIDENTIALITY

The results of this study will be kept strictly confidential. Your written consent is required for the researchers to release any data identified with you as an individual to anyone other than personnel working on the project. The information you provide will have your name removed and only a subject number will identify you during analyses and any written reports of the research.

The experiment may be videotaped. If it is taped, the tapes will be stored securely, viewed only by the experimenters (Christopher L. North, Mehmet C. Dasiyici), and erased after 3 months. If the experimenters wish to use a portion of your videotape for any other purpose, they will get your written permission before using it.

VI. COMPENSATION

Your participation is voluntary and unpaid. But there is a 15 $ reward for the fastest compilation of tasks for two different versions of the comparison of cursor types.

VII. FREEDOM TO WITHDRAW

You are free to withdraw from this study at any time for any reason.

VIII. APPROVAL OF RESEARCH

This research has been approved, as required, by the Institutional Review Board for projects involving human subjects at Virginia Polytechnic Institute and State University, and by the Department of Computer Science.

IX. SUBJECT'S RESPONSIBILITIES AND PERMISSION

I voluntarily agree to participate in this study, and I know of no reason I cannot participate. I have read and understand the informed consent and conditions of this project. I have had all my questions answered. I hereby acknowledge the above and give my voluntary consent for participation in this project. If I participate, I may withdraw at any time without penalty. I agree to abide by the rules of this project

______________________________________ ______

Signature Date

______________________________________ ______________________

Name (please print) Contact: phone or address or

____

email address (OPTIONAL)

Should I have any questions about this research or its conduct, I may contact:

Investigator: Dr. Christopher L. North Phone (540)231-2458
Professor, Computer Science Department (231-6931)
email: north@vt.edu

Review Board: David M. Moore Phone (540) 231-4991
Office 2000 Kraft Drive, Suite 2000 (0497)

A.2 Demographic and Background Questionnaire

Please help us to categorize our user population by completing the following items.

Gender (circle one): Male Female

Age: _____________

Do you wear glasses or contact lenses (circle one)?

No Glasses Contact Lenses

Are you color blind (circle one)?

Yes No

Are you (circle one)

Right-handed Left-handed Ambidextrous

Occupation (if student, indicate graduate or undergraduate):

____

Major / Area of specialization (if student): _________________________________

Rate your tiredness level today: (circle one)

•-----------------------------•---------------------------•----------------------------•

very tired somewhat tired a little tired not tired at all

Rate your expertise with computers: (circle one)

•-----------------------------•---------------------------•----------------------------•

beginner amateur intermediate advanced

How often do you use computers...

...for work? (circle the best answer)	...for fun? (circle the best answer)
a. not at all	a. not at all
b. once a month	b. once a month
c. once a week	c. once a week
d. several times a week	d. several times a week
e. daily	e. daily

Have you ever used a large display system? If so, please describe it (what type of display was used, what kind of application (e.g. game, virtual environments, geospatial) was running, how did you interact with the system, etc.).

A.3 After Each Cursor Type Questionnaire

Please complete the following questions.

1. Please, rank how much you liked this cursor technique.

1	2	3	4	5	6	7	8	9	10
Hated									Loved

2. Please, rank this cursor technique in terms of the overall ease to use.

1	2	3	4	5	6	7	8	9	10
Not Easy to Use									Very Easy to Use

3. Please, rank this cursor technique in terms of ease to use in **small distances**.

1	2	3	4	5	6	7	8	9	10
Not Effective									Highly Effective

4. Please, rank this cursor technique in terms of ease to use in **large distances**.

1	2	3	4	5	6	7	8	9	10
Not Effective									Highly Effective

5. What aspects of this cursor you liked/disliked?

6. Please feel free to add any comments, suggestions, and concerns.

A.4 Post-Experiment Questionnaire

Please complete the following questions.

1. Which cursor technique you like the best in terms of ease to use?

2. Why did you prefer that cursor technique among others?

3. Do you have any additional comments about better cursor designs for large high resolution displays? Please feel free to add any comments, suggestions, and concerns.

A.5 Statistical Analysis

Movement Times

ANOVA Results for independent variables.

General Linear Model: performance versus cursor type, target distance, target width.

Factor	Type	Levels	Values
Cursor Speed	fixed	4	control, DC1T, DC2T, manual
Cursor Size	fixed	2	static, dynamic
Target Distance	fixed	6	200, 3200, 6200, 9200, 12200, 15200
Target Width	fixed	3	20, 60, 180
Angle	fixed	2	0, 180
Gender	fixed	2	female, male

Analysis of Variance for movement time (performance):

Source	DF	Sum of Squares	F Ratio	Prob > F
CursorSpeed	3	117.4642	80.58609	<.0001
CursorSize	1	63.58555	130.8682	<.0001
CursorSpeed*CursorSize	3	15.12467	10.37625	<.0001
TargetDistance	5	2779.042	1143.934	<0.0001
CursorSpeed*TargetDistance	15	116.0043	15.9169	<0.001
CursorSize*TargetDistance	5	24.08749	9.915104	<0.001
CursorSpeed*CursorSize *TargetDistance	15	9.627741	1.321019	0.179806
TargetWidth	2	896.8636	922.9373	<0.0001
CursorSpeed*TargetWidth	6	14.36171	4.926411	<0.001
CursorSize*TargetWidth	2	4.763124	4.901598	0.00748
CursorSpeed*CursorSize *TargetWidth	6	3.156791	1.082855	0.37007
TargetDistance*TargetWidth	10	14.16259	2.914865	0.001213
CursorSpeed*TargetDistance *TargetWidth	30	18.69391	1.282492	0.139155
CursorSize*TargetDistance *TargetWidth	10	2.977539	0.61282	0.804256
CursorSpeed*CursorSize *TargetDistance*TargetWidth	30	13.08862	0.897942	0.626323
Angle	1	1.407658	2.897163	0.088814
CursorSpeed*Angle	3	1.842316	1.263917	0.285002
CursorSize*Angle	1	0.059069	0.121572	0.727354

CursorSpeed*CursorSize *Angle	3	0.331881	0.227687	0.877175
TargetDistance*Angle	5	3.929901	1.617661	0.151713
CursorSpeed*TargetDistance *Angle	15	6.154448	0.844449	0.627981
CursorSize*TargetDistance *Angle	5	5.374045	2.212112	0.05041
CursorSpeed*CursorSize *TargetDistance*Angle	15	5.354384	0.734673	0.750974
TargetWidth*Angle	2	0.47154	0.485249	0.61558
CursorSpeed*TargetWidth *Angle	6	2.591368	0.888902	0.501926
CursorSize*TargetWidth *Angle	2	0.82684	0.850878	0.427118
CursorSpeed*CursorSize *TargetWidth*Angle	6	1.064236	0.365059	0.901336
TargetDistance*TargetWidth *Angle	10	1.684341	0.346662	0.968154
CursorSpeed*TargetDistance *TargetWidth*Angle	30	7.162871	0.491407	0.990993
CursorSize*TargetDistance *TargetWidth*Angle	10	2.844934	0.585528	0.827142
CursorSpeed*CursorSize *TargetDistance*TargetWidth *Angle	30	6.060456	0.415776	0.997945
Gender	1	3.94598	8.121395	0.004397
CursorSpeed*Gender	3	4.628563	3.175417	0.023161
CursorSize*Gender	1	0.232766	0.479067	0.488886
CursorSpeed*CursorSize *Gender	3	0.621899	0.426652	0.733909
TargetDistance*Gender	5	3.243519	1.335126	0.246155
CursorSpeed*TargetDistance *Gender	15	2.401848	0.329557	0.992555
CursorSize*TargetDistance *Gender	5	1.823849	0.750749	0.585439
CursorSpeed*CursorSize *TargetDistance*Gender	15	2.274611	0.312098	0.994465
TargetWidth*Gender	2	0.398072	0.409645	0.663914
CursorSpeed*TargetWidth *Gender	6	3.150726	1.080775	0.371349
CursorSize*TargetWidth *Gender	2	0.217021	0.223331	0.79986
CursorSpeed*CursorSize *TargetWidth*Gender	6	3.145549	1.078999	0.372443
TargetDistance*TargetWidth *Gender	10	2.005012	0.41266	0.941359

CursorSpeed*TargetDistance *TargetWidth*Gender	30	10.75965	0.738164	0.848118
CursorSize*TargetDistance *TargetWidth*Gender	10	3.091627	0.636301	0.783782
CursorSpeed*CursorSize *TargetDistance*TargetWidth *Gender	30	5.694744	0.390687	0.998861
Angle*Gender	1	0.009297	0.019134	0.889991
CursorSpeed*Angle *Gender	3	0.971912	0.666779	0.572389
CursorSize*Angle *Gender	1	0.011705	0.024091	0.876662
CursorSpeed*CursorSize *Angle*Gender	3	0.991375	0.680131	0.56412
TargetDistance*Angle *Gender	5	5.068069	2.086163	0.064146
CursorSpeed*TargetDistance *Angle*Gender	15	6.774994	0.929594	0.529913
CursorSize*TargetDistance *Angle*Gender	5	0.587677	0.241905	0.943934
CursorSpeed*CursorSize *TargetDistance*Angle *Gender	15	2.080653	0.285486	0.996635
TargetWidth*Angle *Gender	2	1.151608	1.185088	0.305827
CursorSpeed*TargetWidth *Angle*Gender	6	1.516228	0.520103	0.793519
CursorSize*TargetWidth *Angle*Gender	2	0.142967	0.147123	0.863192
CursorSpeed*CursorSize *TargetWidth*Angle*Gender	6	3.140365	1.077221	0.373541
TargetDistance*TargetWidth *Angle*Gender	10	3.191728	0.656904	0.7653
CursorSpeed*TargetDistance *TargetWidth*Angle*Gender	30	11.57361	0.794005	0.779544
CursorSize*TargetDistance *TargetWidth*Angle*Gender	10	1.634605	0.336425	0.971447
CursorSpeed*CursorSize *TargetDistance*TargetWidth *Angle*Gender	30	9.573917	0.656817	0.923506

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	575	4241.505	7.376531	15.18196
Error	3974	1930.866	0.485875	0
C. Total	4549	6172.371	.	.

Comparisons for all Cursor Types using Tukey-Kramer HSD

q*	Alpha
3.03232	0.05

Cursor Speed	Cursor Size						Mean
Manual Cursor	Static	A					3.05798
DC1T	Static		B				2.77582
Control Cursor	Static		B	C			2.73349
Control Cursor	Dynamic		B	C	D		2.65341
Manual Cursor	Dynamic		B	C	D		2.64492
DC1T	Dynamic			C	D		2.5328
DC2T	Static				D		2.51552
DC2T	Dynamic					E	2.29443

Comparisons for all Cursor Speeds using Tukey-Kramer HSD

q*	Alpha
2.56997	0.005

Cursor Speed				Mean
Manual Cursor	A			2.897857888
Control Cursor		B		2.728875527
DC1T		B		2.699886523
DC2T			C	2.426551137

Number of Errors Clicks

ANOVA Results for the relationship between the target width and the number of error clicks.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Size	2	111.7845	55.89224	20.6231	<.0001
Error	113	306.25	2.710177	.	.
C. Total	115	418.0345	.	.	.

Appendix B: Experiment 2

B.1 Informed Consent for Participant of Investigative Project

I. THE PURPOSE OF THIS RESEARCH/PROJECT

You are invited to participate in a study of interaction in large high resolution display. This research studies the different cursor types for large high resolution screens. This study involves experimentation for the purpose of evaluating and improving the user interaction techniques for large high resolution displays.

II. PROCEDURES

You will be asked to perform set of tasks using different cursor types. These tasks consist of clicking objects by using mouse. Your role in these tasks is to finish click the target object as fast as possible. You are helping us to evaluate our system. All information that you help us attain will remain anonymous. The time you take to do each task and other aspects of your interaction with the system will be measured. You may be asked questions during and after the evaluation, in order to clarify our understanding of your evaluation.

You may also be asked to fill out a questionnaire relating to your background with such systems.

The session will last less than two hours. The tasks are not very tiring, but you are welcome to take rest breaks as needed. One scheduled rest break will be given to you about half-way through the experiment. You may also terminate your participation at any time, for any reason.

You will be given full instructions before every task. Because we are measuring a range of abilities, some tasks will be easy and others difficult. It is important that you understand the instructions before beginning each task. If anything is unclear, be sure to ask us questions.

III. RISKS

The proposed experiments are straightforward tests of performance using large high resolution displays, and input devices. Participation involves sitting in front of the display and performing simple clicking tasks. The physical components of these tasks are not stressful, and include head turning and pointing with mouse. All light and sound intensities are well within normal ranges. There are no known mental risks. The risks associated with participation in this study are minimal.

IV. BENEFITS OF THIS PROJECT

Your participation in this project will provide information that may be used to improve the design of new mouse techniques for large high resolution displays. No guarantee of benefits has been made to encourage you to participate.

You are requested to refrain from discussing the evaluation with other people who might be in the candidate pool from which other participants might be drawn.

V. EXTENT OF ANONYMITY AND CONFIDENTIALITY

The results of this study will be kept strictly confidential. Your written consent is required for the researchers to release any data identified with you as an individual to anyone other than personnel working on the project. The information you provide will have your name removed and only a subject number will identify you during analyses and any written reports of the research.

The experiment may be videotaped. If it is taped, the tapes will be stored securely, viewed only by the experimenters (Christopher L. North, Mehmet C. Dasiyici), and erased after 3 months. If the experimenters wish to use a portion of your videotape for any other purpose, they will get your written permission before using it.

VI. COMPENSATION

Your participation is voluntary and unpaid. But there is a 30 $ reward for the fastest compilation of tasks for any cursor types.

VII. FREEDOM TO WITHDRAW

You are free to withdraw from this study at any time for any reason.

VIII. APPROVAL OF RESEARCH

This research has been approved, as required, by the Institutional Review Board for projects involving human subjects at Virginia Polytechnic Institute and State University, and by the Department of Computer Science.

IX. SUBJECT'S RESPONSIBILITIES AND PERMISSION

I voluntarily agree to participate in this study, and I know of no reason I cannot participate. I have read and understand the informed consent and conditions of this project. I have had all my questions answered. I hereby acknowledge the above and give my voluntary consent for participation in this project. If I participate, I may withdraw at any time without penalty. I agree to abide by the rules of this project

Signature Date

Name (please print) Contact: phone or address or

email address (OPTIONAL)

Should I have any questions about this research or its conduct, I may contact:

Investigator: Dr. Christopher L. North Phone (540)231-2458
Professor, Computer Science Department (231-6931)
email: north@vt.edu

Review Board: David M. Moore Phone (540) 231-4991
Office 2000 Kraft Drive, Suite 2000 (0497)

B.2 Demographic and Background Questionnaire

Please help us to categorize our user population by completing the following items.

Gender (circle one): Male Female

Age: _____________

Do you wear glasses or contact lenses (circle one)?

No Glasses Contact Lenses

Are you color blind (circle one)?

Yes No

Are you (circle one)

Right-handed Left-handed Ambidextrous

Occupation (if student, indicate graduate or undergraduate):

____

Major / Area of specialization (if student): _________________________________

Rate your tiredness level today: (circle one)

•---------------------------•-----------------------------•----------------------------•

very tired somewhat tired a little tired not tired at all

Rate your expertise with computers: (circle one)

•-----------------------------•---------------------------•-----------------------------•

beginner amateur intermediate advanced

How often do you use computers...

...for work? (circle the best answer)	...for fun? (circle the best answer)
a. not at all	a. not at all
b. once a month	b. once a month
c. once a week	c. once a week
d. several times a week	d. several times a week
e. daily	e. daily

Have you ever used a large display system? If so, please describe it (what type of display was used, what kind of application (e.g. game, virtual environments, geospatial) was running, how did you interact with the system, etc.).

B.3 After Each Cursor Type Questionnaire

Please complete the following questions.

1. Please, rank how much you liked this cursor technique.

1	2	3	4	5	6	7	8	9	10
Hated									Loved

2. Please, rank this cursor technique in terms of the overall ease to use.

1	2	3	4	5	6	7	8	9	10
Not Easy to Use									Very Easy to Use

3. Please, rank this cursor technique in terms of ease to use in **small distances**.

1	2	3	4	5	6	7	8	9	10
Not Effective									Highly Effective

4. Please, rank this cursor technique in terms of ease to use in **large distances**.

1	2	3	4	5	6	7	8	9	10
Not Effective									Highly Effective

5. What aspects of this cursor you liked/disliked?

6. Please feel free to add any comments, suggestions, and concerns.

B.4 Post-Experiment Questionnaire

Please complete the following questions.

1. Which cursor technique you like the best in terms of ease to use?

2. Why did you prefer that cursor technique among others?

3. Do you have any additional comments about better cursor designs for large high resolution displays? Please feel free to add any comments, suggestions, and concerns.

B.5 Statistical Analysis

ANOVA Results for independent variables.

General Linear Model: performance versus cursor type, target distance, target width.

Factor	Type	Levels	Values
Cursor Type	fixed	6	control, DC2T, DC4T, Auto-HT, Manual-HT, HT-Only
Target Distance	fixed	6	200, 3200, 6200, 9200, 12200, 15200
Target Width	fixed	3	20, 60, 180
Angle	fixed	2	0, 180

Analysis of Variance for movement time (performance):

Source	DF	Sum of Squares	F Ratio	Prob > F
CursorType	5	3756.944282	748.5197184	<.0000
Size	2	989.983078	493.1014403	<.0001
CursorType*Size	10	301.5811564	30.04295849	<.0001
Angle	1	1.49232821	1.486629834	0.222863725
CursorType*Angle	5	8.598079755	1.71304969	0.128184027
Size*Angle	2	2.995614122	1.492087765	0.225117805
CursorType*Size *Angle	10	10.44530736	1.040542251	0.406123202
Distance	5	1326.007232	264.1887888	<.0001
CursorType*Distance	25	183.0418798	7.293717768	<.0001
Size*Distance	10	24.0691741	2.397726725	0.007882659
CursorType*Size *Distance	50	39.2025882	0.781057904	0.866480204
Angle*Distance	5	2.523533291	0.502779463	0.774361121
CursorType*Angle *Distance	25	28.68208031	1.142902373	0.28366111
Size*Angle*Distance	10	13.69130101	1.363902151	0.190914488
CursorType*Size *Angle*Distance	50	53.46610824	1.065239015	0.35195333

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	215	6741.495	31.35579	31.23606
Error	2326	2334.916	1.003833	0
C. Total	2541	9076.41	.	.

Comparisons for all cursor types using Tukey-Kramer HSD

q*	Alpha
2.85192019	0.05

Cursor Type					Mean
HT-Only	A				5.906792884
Control Cursor		B			3.208158245
AutoHT			C		2.732688856
ManualHT			C	D	2.589987634
DC4T			C	D	2.492421505
DC2T				D	2.42169979

Number of Error Clicks

ANOVA Results for the relationship between the cursor type and the number of error clicks.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
CursorType	5	81820.38	16364.08	37.78196	<.0001
Error	631	273297.9	433.1187	.	.
C. Total	636	355118.3	.	.	.

ANOVA Results for the relationship between the target width and the number of error clicks.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Size	2	27483.11	13741.56	26.591	<.0001
Error	634	327635.2	516.7748	.	.
C. Total	636	355118.3	.	.	.