CIRCLING INTERFACE: AN ALTERNATIVE INTERACTION METHOD FOR ON-SCREEN OBJECT MANIPULATION

by

Hyun W. Ka

B. S. in Education and Computer Science, Yonsei University, 2002

M. S. in Instructional Technology, Yonsei University, 2006

Submitted to the Graduate Faculty of

Health and Rehabilitation Sciences in partial fulfillment

of the requirements for the degree of

PhD in Rehabilitation Science

University of Pittsburgh

2013

UNIVERSITY OF PITTSBURGH

Health and Rehabilitation Sciences

This dissertation was presented

by

Hyun W. Ka

It was defended on

July 17, 2013

and approved by

Alicia M. Koontz, PhD, Associate Professor

Nancy A. Baker, ScD, Associate Professor

Edmund F. LoPresti, PhD, Adjunct Assistant Professor

Dissertation Advisor: Richard C. Simpson, PhD, Associate Professor

CIRCLING INTERFACE: AN ALTERNATIVE INTERACTION METHOD FOR ON-SCREEN OBJECT MANIPULATION

Hyun W. Ka, PhD

University of Pittsburgh, 2013

An alternative interaction method, called the circling interface, was developed and evaluated for individuals with disabilities who find it difficult or impossible to consistently and efficiently perform pointing operations involving the left and right mouse buttons. The circling interface is a gesture-based interaction technique. To specify a target of interest, the user makes a circling motion around the target. To specify a desired pointing command with the circling interface, each edge of the screen is used. The user selects a command before circling the target. Empirical evaluations were conducted with human subjects from three different groups (individuals without disability, individuals with spinal cord injury, and individuals with cerebral palsy), comparing each group's performance on pointing tasks with the circling interface to performance on the same tasks when using a mouse button or dwell-clicking software. Across all three groups, the circling interface was faster than the dwelling interface (although the difference was not statistically significant). For the single-click operation, the circling interface was slower than dwell selection, but for both double-click and drag-and-drop operations, the circling interface was faster. In terms of performance accuracy, the results were mixed: for able-bodied subjects circling was more accurate than dwelling, for subjects with SCI dwelling was more accurate than circling, and for subjects with CP there was no difference. However, if errors caused by circling on an area with no target or by ignoring circles that are too small or too fast were automatically

corrected by the circling interface, the performance accuracy of the circling interface would significantly outperform dwell selection. This suggests that the circling interface can be used in conjunction with existing pointing techniques and this combined approach may provide more effective mouse use for people with pointing problems. Consequently, the circling interface can improve clinical practice by providing an alternative pointing method that does not require physically activating mouse buttons and is more efficient than dwell-clicking. It is also expected to be useful for both computer access and augmentative communication software.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I would like to express my most sincere gratitude:

To Dr. Richard C. Simpson for being such a supportive and excellent academic advisor during my whole graduate days in PITT;

To Dr. Edmund F. LoPresti for his being always open to me and sharing his best knowledge and experience in the field of computer access with me;

To Dr. Alicia M. Koontz for her warmhearted encouragement in process of making the research design steadfast;

To Dr. Nancy A. Baker for her providing me with lots of valuable clinical insights and helpful information;

To all participants who were interested and willingly took part in this research projects;

To Young J. Chung, Hosik Kim, and Changwha Lee, for their unchanging love and support through the years;

To my brothers and sisters in Christ who have continuously prayed for me;

And a special thanks to my wife and daughter, Heeok and Irene, for their being always with me.

It is good to praise the LORD and make music to your name, O Most High, to proclaim your love in the morning and your faithfulness at night (Ps 92:1)

1.0 INTRODUCTION

1.1 PURPOSE AND SPECIFIC AIMS

The goal of this research project was to develop and evaluate an alternative interaction method, the *circling interface*, for selecting and manipulating on-screen objects based on circling the target, rather than pointing and clicking.

The following specific aims were pursued:

- Develop an alternative interaction technique that allows individuals with disabilities to circle targets to perform pointing operations that currently require using the left or right mouse buttons or dwell-clicking software.
- Conduct empirical evaluations to determine the efficacy of the circling interface by comparing the performance on pointing tasks with a circling interface to performance on the same tasks when using the existing methods.

1.2 BACKGROUND

Disability has significant influence on almost all aspects of life. Many people with disabilities do not have equal access to education, employment opportunities, and health care, do not receive the disability-related services that they require, and experience exclusion from everyday life activities. Not only do people with disabilities experience worse socioeconomic outcomes and poverty than people without disabilities, but they are also more vulnerable to psychosocial problems (World Health Organization, 2011).

"Computers" as an umbrella term encompassing all relevant devices and technologies can play a key role in addressing these issues by providing a viable means of compensating for activity and participation restrictions as well as physical impairments. Computer access increases educational opportunities for people with disabilities by eliminating physical barriers and providing better access to materials, and enhances vocational opportunities by allowing their job description to be defined by knowledge-based activities rather than heavy manual labor (Anson, 1997; Chen et al., 2006; Young, Levi, Tumanon, Desei, & Sokal, 2000). Computer access also has a positive impact on health by facilitating interaction with clinicians and peers as well as by providing access to health information (Dobransky & Hargittai, 2006). In addition, through computers, people with disabilities can be informed of the disability-related services that they require, and also proactively advocate their rights by participating in government (Bowker, 2010; Czaja & Lee, 2007). Access to a computer also allows people with disabilities to enjoy more independent living by enabling daily activities such as banking, shopping, and information retrieval (Czaja & Lee, 2007; Fox, Sohlberg, Fickas, Lemoncello, & Prideaux, 2009; Richards & Hanson, 2004).

Computer access also has positive psychosocial influences. For example, it can prevent self-consciousness and social anxiety by facilitating communication and promoting information exchange (Drainoni et al., 2004), In addition, because the Internet provides anonymity, people with disabilities can be evaluated based on the strength of their contributions rather than their physical appearance or disability, and can enjoy feelings of equality (Madara, 1997; McKenna &

Seidman, 2005). Such active participation can bring people with disabilities greater levels of self-acceptance as well as decreased feelings of isolation (McKenna & Bargh, 2000; Morahan-Martin & Schumacher, 2003).

Pointing to interact with on-screen objects (e.g., icon, button, text, hyperlink, scroll, and etc.) has been increasingly emphasized in using computers (Trewin & Pain, 1999a) as the graphical user interface (GUI) has gained in popularity. As a general rule, computer users spend one- to two-thirds of their computer work using pointing devices (Keir, Bach, & Rempel, 1999) and, in many mobile devices, text entry has become a point-and-click task. In addition, some individuals with disabilities have to rely almost exclusively on a pointing device to access their computer.

It is difficult to precisely estimate the size of the population who experience pointing problems, but a survey sponsored by Microsoft found that 19% (24.4 million) of computer users have a mild dexterity impairment and 5% (6.8 million) of computer users have a severe dexterity impairment (Stevenson & McQuivey, 2003). In particular, some individuals belonging to this population find it difficult or impossible to consistently and efficiently perform operations involving the physical buttons attached to a computer mouse (e.g., left clicking to select an item or activate a button, right clicking to display a context menu, double-clicking an icon to launch an application, clicking-and-dragging to move an icon or select text) (S. Keates, F. Hwang, P. Langdon, P. Clarkson, & P. Robinson, 2002; Trewin & Pain, 1999a; J.O. Wobbrock & K.Z. Gajos, 2008). Several alternatives to the left and right buttons on traditional computer mice already exist, but each approach has its own limitations and trade-offs.

1.3 STRUCTURE OF DISSERTATION

This dissertation is written in seven chapters. This chapter describes the objectives, background and significance of this research project. Chapter 2 describes the physical disabilities affecting pointing operations, the characteristics and specific pointing problems experienced by the target population, and reviews the existing approaches and solutions. In Chapter 3, the design and technical details of the circling interface are described. Chapters 4 through 6 cover the empirical evaluation phase of this research project. Each of these chapters evaluates the circling interface with a different user group: individuals without disability (Chapter 4), individuals with spinal cord injury (Chapter 5), and individuals with cerebral palsy (Chapter 6). Each chapter compares participants' performance on pointing tasks with the circling interface to performance on the same tasks when using the existing methods. Chapter 7 concludes by summarizing the findings from the various phases of this study and suggesting possible modifications to the current circling interface along with directions for future research. Appendices provide various research materials used in this project.

2.0 POINTING PROBLEMS AND EXISTING SOLUTIONS

2.1 PHYSICAL DISABILITIES AFFECTING POINTING OPERATIONS

There are a number of health conditions resulting in physical disability that can interfere with an individual's ability to select and manipulate objects in a GUI (Sears & Young, 2002), including amyotrophic lateral sclerosis, arthritis, cerebral palsy, locked-in syndrome, missing limbs or digits, multiple sclerosis, muscular dystrophy, Parkinson’s disease, seizures, spinal cord injury, stroke, traumatic brain injury, and tremors. In addition, age-related factors, such as macular degeneration and poor eye-hand coordination, can affect pointing operations (Arch, 2009, 2010; Becker, 2004; Holt, 2000; Ketcham, Seidler, Van Gemmert, & Stelmach, 2002; Rogers, Stronge, & Fisk, 2005; Taveira & Choi, 2009; Walker, Philbin, & Fisk, 1997). Functional limitations caused by these medical conditions include:

- difficulty grasping or lifting;
- poor muscle control;
- weakness and fatigue;
- limited range of motion;
- difficulty reaching things;
- difficulty making a rapid change in direction;
- difficulty doing complex or compound manipulations (e.g., push and turn);

- delayed response time;
- difficulty speaking, seeing, or sensing

When performing point-and-click operations, these functional limitations result in a variety of difficulties.

2.2 POINTING PROBLEMS

When performing pointing operations, individuals with limited motor functions show several different characteristics in cursor movements compared to non-disabled individuals. Hwang and colleagues investigated mouse movements of motion-impaired users, analyzing sub-movements (Hwang, Keates, Langdon, & Clarkson, 2004). They found that physically disabled individuals pause more often and for longer, require up to five times more sub-movements to complete the same task, and exhibit a positive correlation between error and peak sub-movement speed that does not exist for non-disabled individuals (Hwang et al., 2004). Similar phenomena have been found among older computer users. Their pointing performance is characterized by lower velocity, more sub-movements, a longer deceleration phase, and more frequent pauses, compared to younger computer users (Hanson, 2009; Hertzum & Hornbaek, 2010; Rogers et al., 2005; Walker et al., 1997).

The most common pointing problems encountered by individuals with limited motor functions are:

- Some individuals have difficulty grasping, lifting and repositioning a pointing device (Trewin & Pain, 1999a).

- Some individuals make uncontrolled long straight mouse movements (Choe, Shinohara, Chilana, Dixon, & Wobbrock, 2009; Smith, Sharit, & Czaja, 1999).
- Some individuals have difficulty positioning the mouse cursor inside a target (Choe et al., 2009; Smith et al., 1999), particularly if the target is small (Accot & Zhai, 2002).
- Some individuals cannot activate the buttons built into their pointing device.
- Some individuals frequently activate the mouse button unintentionally. In Keate's study (S. Keates, F. Hwang, P. Langdon, P.J. Clarkson, & P. Robinson, 2002), 17% of clicking errors occurred when the mouse cursor was far outside the target (a distance from the target's center that was between 50% and 100% of the target's radius) when the button was pressed. An additional 7% of clicking errors occurred when the mouse cursor was very far outside the target (a distance from the target's center that was greater than 200% of the target's radius).
- Some individuals often move the mouse off the target when activating buttons for click or double-click operations. In Trewin's study the mouse up position was not the same as the down position in 28.1% of mouse clicks by people with disabilities (Trewin & Pain, 1999a) and, in Keates' study of young adults, older adults and adults with Parkinson's disease, 15% of mouse errors occurred because the button was pressed when the cursor was on the target but the cursor slipped off the target before the button was released (Simeon Keates & Trewin, 2005).
- Some individuals have difficulty activating mouse button twice quickly enough to register a double click.

- Some individuals have difficulty keeping the left mouse button activated while performing drag-and-drop operations. In Trewin's study, only 45% of drags by disabled subjects were successful (Trewin & Pain, 1999a).

2.3 EXISTING SOLUTIONS

Several alternatives to the left and right buttons on traditional computer mice already exist, including commercially available hardware and software and research solutions.

2.3.1 Commercially Available Solutions

2.3.1.1 Alternative Pointing Devices

Trackball

A trackball is a pointing device consisting of a ball held by a socket containing sensors to detect rotation of the ball about two axes. The user rolls the ball with the thumb, fingers, or palm to move the mouse cursor. Thus, the trackball stays in a fixed position and has a smaller footprint than a normal mouse. By separating the action of moving the cursor from the action of pressing buttons, it prevents users from unintentionally activating buttons. However, trackballs can be inappropriate for users with poor dexterity, and they have a tendency to be slightly slower than a mouse (Casali & Chase, 1993; Douglas, Kirkpatrick, & MacKenzie, 1999).

Trackpad

A trackpad (also known as a touchpad) is a pointing device which has a conductive surface that can translate the motion and position of a user's fingers to a relative position on screen. It is a common feature of laptop computers. For clicking operations, either external buttons or tapping on the surface is used. Like a trackball, it stays at a fixed place and requires less working space than a normal mouse. In recent years, larger external trackpads have been introduced that support more complicated pointing operations.

External Switch Interface

Some pointing devices have jacks that allow any switch to emulate the left or right mouse button. Separate interfaces are also available that connect to the computer through a USB port. Like a trackball, an external switch interface reduces unintentional clicking by separating the action of moving the cursor from the action of pressing the mouse button. A switch interface also allows users to choose a more effective switch than the built-in mouse buttons.

Joystick

Not only do joysticks require less hand motion, but they also separate the action of moving the cursor from the action of pressing buttons. Some joysticks require additional software to function as a pointing device. A joystick is classified as either isotonic and isometric depending on whether it responds to angular deflection or force, although recent designs blur the distinctions between the two (Lipscomb & Pique, 1993).

An *isotonic joystick*, also called position joystick, senses the angle of deflection of the joystick handle, moving from the center position. The detected angular deflection and velocity determines the magnitude and direction of change in the cursor's position. When the joystick is released, it returns to center.

An *isometric joystick*, also called a force joystick or stiff stick, converts applied force into a proportional electrical output resulting in the magnitude and direction of change in the cursor's position. Isometric joysticks usually require more practice time to achieve expert cursor control, but when mounted on a keyboard in form of a trackpoint, it takes less homing time to switch between devices, compared to the mouse (Rutledge & Selker, 1990). However, this reduction in device switching time is usually not enough to offset the slower performance of the isometric joystick (Douglas & Mithal, 1994).

Head-Mounted Mouse Emulator

Head-mounted mouse emulators are useful for individuals who have limited arm range of motion, have limited grip strength, and have poor hand/arm control. In general, head-mounted mouse emulators translate changes in the user's head position and orientation into directly proportional movements of the mouse cursor on the screen. Some head-mounted mouse emulators provide physical switches for mouse buttons such as sip and puff or touch switches. For those who cannot use physical switches, click emulation software, such as a dwelling interface, is often used. In general, a head-mounted mouse emulator requires significantly more training compared to other pointing devices.

Target size, movement distance between targets, and control-display gain (CDG) all influence pointing performance with a head-mounted mouse emulator mouse (M. L. Lin, Radwin, & Vanderheiden, 1992; Mei, Robert, & Gregg, 1992; Schaab, Radwin, Vanderheiden, & Hansen, 1996). Both target size and movement distance are more influential than CDG. Between target size and movement distance, it was found that pointing performance with a head-mounted mouse emulator was affected by target size more than movement distance (Radwin, Vanderheiden, & Lin, 1990). In terms of movement direction, while a head-mounted mouse emulator showed slower performance in horizontal movements than a normal mouse, it demonstrated faster performance in vertical movements (Radwin et al., 1990).

Eye-Gaze Tracking Device

Eye-gaze tracking devices can be a last resort for some individuals with no control, or severely limited control, over their body movements (Hinckley, 2002; Hornof, Cavender, & Hoselton, 2004; Lankford, 2000). Either one or both eyes are tracked by a camera mounted on a computer monitor. Special software processes the camera image to determine where the user is looking (gaze point), and then the software moves the mouse cursor to the gaze point. Mouse clicks are done with either an eye blink or a dwell clicking method. Because the human eye is constantly in motion, a high sampling rate and intelligent data filtering are required to ensure the reliability of the system (Doherty, Cockton, Bloor, & Benigno, 2000; Hinckley, 2002; Patmore & Knapp, 1998).

2.3.1.2 Software Approaches

In addition to alternative pointing devices, pointing problems experienced by individuals with limited motor functions can be addressed by adjusting parameters affecting pointing performance and/or using standalone software applications or built-in features provided by most operating systems.

Make Things on the Screen Closer

Decreasing the distance the mouse cursor must travel can make pointing operations more efficient (Atwood, 2006). In current operating systems, this can be implemented by adjusting the screen resolution. Decreasing the number of pixels on the screen has the effect of making on-screen objects larger, but the number of pixels that the mouse cursor must pass through remains the same. Thus, while this approach does not change the difficulty of mouse cursor movements it does limit the maximum possible movement amplitude.

Make Things on the Screen Bigger

In general, bigger targets are more efficiently acquired and manipulated with pointing devices (Arnaut & Greenstein, 1990; Atwood, 2006; Hale, 2007; I. S. MacKenzie, 1992; Sandfeld & Jensen, 2005; Smith et al., 1999; Stafford, 2005; Walker, Meyer, & Smelcer, 1993; Wobbrock, Cutrell, Harada, & MacKenzie, 2008). Most current operating systems allow the user to change the size (in pixels) of on-screen objects without changing the distance traveled. Increasing the size of these objects can make it easier to perform pointing operations, at the cost of reduced space on the screen. Appropriate object size should be determined based on its expected frequency of use (Hale, 2007).

Adjust Control-Display Gain

Control-display gain (CDG), typically referred to as *gain* or *sensitivity*, is related to how far the mouse cursor moves in response to movement of the pointing device (Arnaut & Greenstein, 1990; Casiez, Vogel, Balakrishnan, & Cockburn, 2008). If gain is less than 1, the mouse cursor on the screen moves more slowly and a shorter distance than the pointing device, if gain is greater than 1, the cursor moves proportionally farther and faster than the pointing device. If gain is equal to 1, the cursor moves at the same distance and speed as the pointing device.

Choosing a gain value that minimizes user effort is crucial: if the value is too low, task completion time increases because the user has to make multiple ballistic movements to cover the same distance; a value that is too high may also cause task completion time to increase because of difficulty with fine motor movements; an appropriate value for gain allows users to reach targets in a single ballistic movement and make adequate fine movements to hit the target (Casiez et al., 2008; Cook, Dobbs, Warren, & McKeever, 2005; Sandfeld & Jensen, 2005; Trewin & Pain, 1999a).

Adjust Parameters for Pointing Device Configuration

Most computer operating systems allow users to configure the pointing device's behavior. Below is a list of typical parameters available for pointing device configuration, adopted from (Simpson, Koester, & Lopresti, 2010).

- Button-handedness: Controls the functions assigned to the left and right mouse buttons.
- Click method: Whether the user performs a single- or double-click to select icons.
- Double-click speed: Controls the allowable time between two clicks in a double-click operation.
- Enhanced pointer precision: The Enhance Pointer Precision (EPP) setting enables a complex algorithm controlling the velocity and acceleration of the pointing cursor.
- Snap-to-default: If this option is active, when a dialog box appears on the screen, the cursor will immediately move to the default button (e.g., “OK”).

Mouse-Keys

Most computer operating systems provide mouse-keys as a built-in alternative pointing feature, with which users can move on-screen mouse cursor using the numeric keypad on the keyboard, instead of a pointing device like a computer mouse and mouse buttons.

Click-Lock

The click-lock feature is intended for users with trouble moving the mouse while pressing the left mouse button (i.e., drag-and-drop). The click-lock feature allows users to hold the mouse button for a set amount of time to engage the click-lock, move the mouse to the new location, and then click it again. Users thus do not have to keep the mouse button depressed for entire duration of the move.

Dwell-Clicking

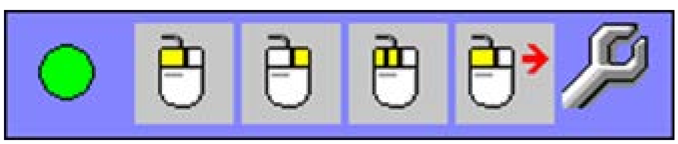

Figure 1. Tool Pallet

Dwell-clicking software, which is often used with head-mounted mouse emulators and eye-gaze systems, is an alternative to using mouse buttons. In order to specify a target of interest, users place the mouse cursor within the boundary of the target for a predefined period of time, instead of pressing mouse buttons (Soukoreff & MacKenzie, 2004). As shown in Figure 1, a tool pallet, which includes small targets activated by the same mechanism, is used to specify mouse button actions like left click, right click, and drag-and-drop.

Users need to have ability to keep the mouse cursor still on the target for the preset amount of time to successfully use dwell-clicking. It is therefore crucial to choose a dwell time that is short enough that users can successfully hover over the target but not so short that they have frequent inadvertent activations. This is especially an issue when performing right-clicks, double-clicks and drag-and-drop operations, where the software will typically perform the selected operation once and then reset back to a default of a single left-click, requiring the user to start the process over from the beginning. One approach to addressing this issue with dwell-clicking in combination with head-mounted mouse emulators is to allow the user to adjust a movement threshold to compensate for small, involuntary movements. The emulated click actions occur as long as the cursor movement stays within the set movement threshold for the dwell time. Another approach to address the same issue is to provide users with an emulated joystick mode, in which they can rest their heads in a neutral position once the cursor enters the boundary of the target (Cook et al., 2005; Evans, Drew, & Blenkhorn, 2000).

2.3.2 Research Solutions

McGuffin and Balakrishnan attempted to improve target acquisition by developing target expansion widgets, which enlarges a target when the mouse cursor was within 90% of the distance to it (McGuffin & Balakrishnan, 2005). In their experiment, in which participants selected a single isolated button, it was shown that even very small targets could be acquired efficiently with target expansion. For interfaces with multiple expanding widgets, however, problems occurred when adjacent targets overlapped when expanded.

Kabbash and Buxton investigated an alternative on-screen target acquisition technique, called the Prince Technique (also known as area cursors), in which the pointing cursor is represented by an area, rather than by a point. They claimed that selecting a small target with an area cursor could be modeled by a modified Fitts' law (Fitts, 1954; I. S. MacKenzie, 1992) in which the W term applied to the width of the cursor, rather than the width of the target (Kabbash & Buxton, 1995). In an experiment with twelve non-disabled participants, the area cursor was more accurate than the normal mouse cursor but it took longer to select a target. Based on this research, Grossman and Balakrishnan introduced an improved version of area cursors, called bubble cursors, which dynamically resized the activation area depending on the proximity of surrounding targets (Grossman & Balakrishnan, 2005). In one- and two-dimensional target acquisition tasks, the bubble cursor significantly outperformed the normal mouse cursor and object pointing, which allowed the mouse cursor to immediately jump to the nearest boundary of the nearest object based on the initiating movement direction (Guiard, Blanch, & Beaudouin-Lafon, 2004).

Sticky targets, which dynamically change the gain according to the mouse cursor position relative to the target location, have been investigated by several researchers (Blanch, Guiard, & Beaudouin-Lafon, 2004; A. Cockburn & Brewster, 2005; Andy Cockburn & Firth, 2003; Keyson, 1997). In one-dimensional pointing tasks, the average pointing time decreased from 10.9% to 27% per subject, and the error rate decreased from 31% to 62% per subject. Performance also improved in two-dimensional target acquisition tasks, but the improvements in both speed and accuracy were less dramatic.

In Worden's study, the effectiveness of area cursors and sticky targets were evaluated to determine if they improved the performance of older adults in basic selection tasks (Worden, Walker, Bharat, & Hudson, 1997). When combined, these techniques shortened target acquisition times for older adults by up to 50%.

In a series of studies, Ahlstrom and colleagues explored the possibility of force fields, which warp the screen cursor toward the target center (Ahlström, Hitz, & Leitner, 2006). In the first study, force fields reduced target acquisition time. In the second study, where the force field technique was compared with sticky targets in two realistic pointing situations involving several densely placed targets, they showed that the force fields could select targets that the sticky target technique did not.

Guiard and colleagues introduced the concept of object pointing, which allowed the mouse cursor to immediately jump to the nearest boundary of the nearest object, based on the initiating movement direction (Guiard et al., 2004). In experiments, it was shown that object pointing outperformed traditional mouse control and the benefits of object pointing increased as pointing tasks became more challenging (with a higher index of difficulty or a longer distance between objects) and as object density increased. Applying a similar concept, Ka and Simpson explored a method to improve target acquisition by inferring the user's intended target based on real-time mouse movement information and the user interface elements currently displayed on the computer screen. During preliminary testing of the prototype application, the software reduced pointing errors and the physical effort needed to operate a pointing device (Ka & Simpson, 2010).

Another group of investigators have explored crossing interfaces, both for able-bodied computer users and computer users with disabilities (Accot & Zhai, 2002; Choe et al., 2009; Moffatt & McGrenere, 2009; J.O. Wobbrock & K.Z. Gajos, 2008). In a crossing interface, the pointing cursor passes through (or crosses) a target to select it. Investigators demonstrated that participants who have difficulty using a standard mouse could select items with goal crossing faster than with area pointing, which requires positioning the mouse cursor within a confined area and then clicking it (Jacob O Wobbrock & Krzysztof Z Gajos, 2008).

Cockburn and Firth performed a comparative evaluation of three different alternative pointing methods, including expanding targets, sticky targets, and goal crossing (Andy Cockburn & Firth, 2003). Given two different pointing tasks (target acquisition and an window resizing task), goal-crossing allowed the fastest target acquisition, but produced high error rates and was unpopular with participants. Both expanding target and sticky target techniques also allowed faster target acquisition than the traditional pointing device.

Wobbrock and colleagues developed and evaluated an alternative pointing method, called Angle Mouse (Wobbrock, Fogarty, Liu, Kimuro, & Harada, 2009), which provides target-agnostic pointing assistance by continually adjusting the control-display gain based on how straight or angular the mouse movement is. When the mouse moves straight, the gain is kept high, but when the angular deviation of mouse movement is high, often near targets, the gain is dropped, making targets bigger in motor-space. The Angle Mouse improved the pointing throughput by 10.3% over the default mouse settings and 11.0% over sticky targets for participants with limited motor functions.

Trewin and colleagues developed and evaluated clicking assistance software, called Steady Clicks, to address errors due to slipping off the target and accidental clicking. Steady Clicks works by preventing cursor movement while the user is pressing the mouse buttons, preventing overlapping button presses and filtering button clicks while the pointing device is moving at a high velocity (Trewin, Keates, & Moffatt, 2006). Test participants with limited motor functions found that Steady Clicks helped them to select targets with less effort and higher accuracy.

2.3.3 Limitations and Trade-Offs

Some individuals with disabilities find that one or more of the existing solutions meets their needs, but others do not. Commercially available alternative pointing devices can involve various limitations or trade-offs: Switches and other hardware (joysticks more so than trackballs) can be expensive. External devices can also be difficult to transport and connect to multiple computers. In addition, some individuals with disabilities prefer to use a regular mouse. These individuals may share a computer with other (non-disabled) individuals or may wish to avoid the stigma associated with assistive technology (Trewin et al., 2006). In addition, most people have prior experience with the mouse and may find other devices too expensive or difficult to understand (Trewin et al., 2006). Finally, some individuals with disabilities want to be able to pull up to their desk and work, without needing assistance in positioning input devices or switches. For example, individuals who use head-mounted mouse emulators will often wear the reflective dot all day so they can come and go from their computer as they please. These individuals typically use dwell-clicking software to avoid the need for an attendant to position a switch for them.

Software approaches have their own issues as well. For example, many individuals who use head-mounted mouse emulators in combination with dwell-clicking software have difficulty finding the right balance between a dwell time that is long enough to minimize unwanted clicks but not so long that intended clicking is too difficult. Through built-in accessibility options within operating systems, the users are provided with a number of settings which can be adjusted, but performing all possible adjustments is not only a complex task but it is also not always clear what settings are available or how they can best be tuned for an individual user (Trewin & Pain, 1999a, 1999b). For example, in order for an individual to appropriately set parameter values within the Windows operating system, he or she needs to access separate

Control Panel applications with a number of tabbed panels, knowing the most appropriate values for all applicable settings.

As for the existing research solutions, most of them not only remain still in the lab, but also have a tendency to focus on a specific aspect of pointing operations. Improving target acquisition has been given much more attention than enhancing clicking issues, such as single-click, double-click, and drag-and-drop (Trewin et al., 2006). Most of the target acquisition facilitators like goal-crossing have the two challenges: (1) how to specify what operation to perform on the target once it is selected and (2) how to specify the target without affecting other targets in densely packed interfaces (referred to as the “occlusion problem”). In real world situations, these issues lead to limited suitability for applications with densely-packed targets, like on-screen keyboards and web sites with lots of links. Steady Click as a clicking assistance also has its own drawback: When the feature is active, dragging functions are limited (only a very long drag can break out of the freeze threshold) (Trewin et al., 2006).

3.0 CIRCLING INTERFACE

3.1 WHAT IS THE CIRCLING INTERFACE?

Circling has been studied for pen-based interfaces (Mizobuchi & Yasumura, 2004) but has not achieved popularity because it is less efficient than tapping (on a pen-based interface) or pointing and clicking (on a mouse-based interface) for non-disabled users (Mizobuchi & Yasumura, 2004). However, a circling interface might address some of the limitations and trade-offs associated with existing pointing solutions available for individuals with disabilities (see Section 2.2.3).

First, like many of the existing solutions, circling takes advantage of an individual's ability to accurately move the mouse cursor while eliminating the need to press physical buttons. However, circling differs from solutions like dwell-clicking in that it does not necessarily require precise targeting. For example, if more than one potential target is circled the software can present a magnified version of the circled area to allow the user to specify the desired target. This would require more actions than some other approaches, but the time saved by eliminating the need to precisely target small items on the screen may compensate for the time required to perform multiple actions.

Second, circling is better-suited for densely-packed interfaces than most research solutions (e.g., goal-crossing, area cursors) because it allows the user to explicitly identify the target. Circling can also allow the user to specify the desired operation through the act of circling. For example, circling clockwise can indicate a left mouse click and circling counter-clockwise can indicate a right mouse click and circling once can indicate a single click, circling twice can indicate a double click and circling three times can indicate click-and-drag.

Finally, as a software-based solution, circling may better meet the needs of some users than other solutions, even if it does not provide a significant performance improvement. For example, a software-based solution is less expensive, more portable, easier to install than many hardware devices and can be used with "traditional" pointing devices like mice and trackpads. A software solution also does not require assistance with mounting or positioning hardware and does not "tie" the user to the workstation with device cables.

3.2 INTERACTION DESIGN OF CIRCLING INTERFACE

We adopted a formative design approach, collaborating with potential users, assistive technology practitioners, an occupational therapist and ergonomic expert, and a software engineer. While going through the formative research, the following design issues were addressed.

3.2.1 How to Specify the Target

Two approaches were explored for users to specify the target of interest with the circling interface.

Simple Approach

In this approach, whatever is at the center of the circled area is considered the target. However, it was found that this approach required too much accuracy for motion impaired users. In particular, it showed limited suitability for specifying small and densely-packed targets.

Magnification Approach

In this approach, if more than one potential target is circled the circling interface can present a magnified version of the circled area to allow the user to specify the desired target. This requires more actions than some other approaches, but the time saved by eliminating the need to precisely target small targets on the screen can compensate for the time required to perform multiple actions. Other investigators have demonstrated a similar concept, called a zooming interface, that enhanced performance in selecting small targets (Bates & Istance, 2002; Lankford, 2000).

3.2.2 How to Specify a Pointing Command

Several design ideas were explored to allow the user to more efficiently specify a desired pointing command (e.g., single-click, double-click, and drag-and-drop) with the circling interface.

Number and direction of circles

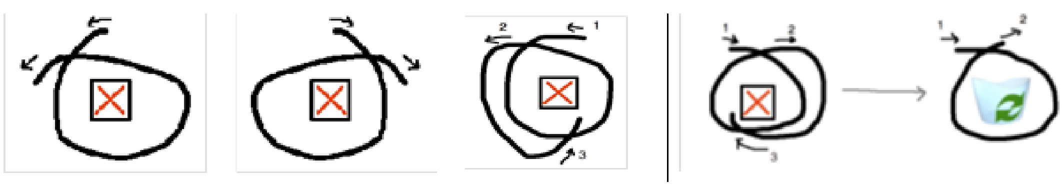

Figure 2. Number and Direction of Circles

As shown in Figure 2, the number of times the mouse cursor circles the target and the direction in which the target is circled dictates the command (e.g., counter-clockwise once for left-click; counter-clockwise twice for double-click; clockwise once for right click; clockwise twice for drag).

Tool Palette

As with dwell-clicking software, the user selects a command from a palette before circling the target.

Edge Commands

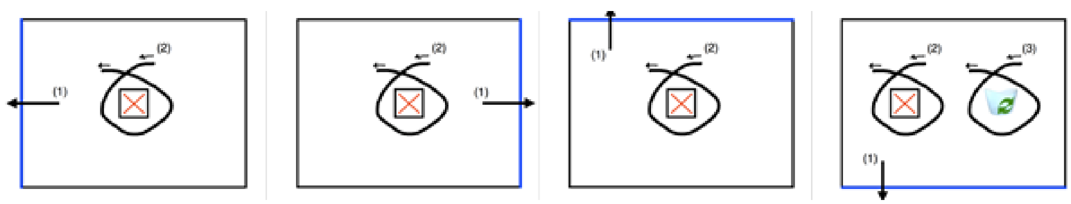

Figure 3. Edge Commands

As shown in Figure 3, each edge of the screen is used as a command which is triggered by mouse cursor (e.g., left edge for left-click; right edge for right-click; top edge for double-click; bottom edge for drag). The user selects a command before circling the target. To prevent the user from unintentionally specifying a command, a timing threshold can be considered. Locating objects along an edge decreases target selection time (Accot & Zhai, 2002; Walker et al., 1993), because the screen edge emulates a target with an infinite effective width that the mouse cursor cannot overshoot (Atwood, 2006; Hale, 2007).

The design ideas above were iteratively refined through frequent interaction with collaborators and numerous design revisions. The magnification approach was finally adopted as the means of specifying the target of interest and the edge commands approach was adopted as the means of specifying commands.

3.2.3 What Feedback Should Be Provided

It is crucial to provide some kind of intermediate feedback so that any recognition delay does not negatively affect the user’s performance (Hinckley, 2002). The circling interface provides the user with the following visual and auditory feedback:

Trail of the mouse cursor

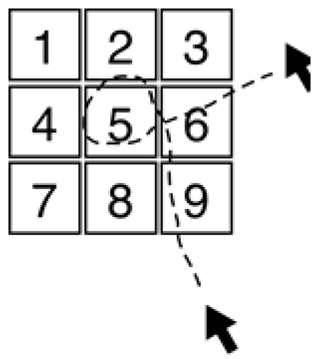

Figure 4. Trail of the Mouse cursor

As shown in Figure 4, the trail of the cursor plays an important role in the circling interface. Not only does it provide users with direct visual feedback on the cursor's movement, but it also defines the range of the available circling area. It was found that there is a trade-off between performance speed and accuracy depending on the length of the trail; while a longer trail allows users to faster make a circle, the error rate increases. On the other hand, while a shorter trail can lower the error rate, it takes longer to circle a target. It is crucial to properly adjust the length of the trail via parameter settings.

Additional Visual or auditory feedback

As additional feedback, a beeping sound is provided when the user makes a circle or specifies a pointing command by touching one of screen edges. When performing a drag-and-drop operation, a colored rectangular box appears around the item to be dragged to indicate that dragging is ready.

3.3 IMPLEMENTATION OF CIRCLING INTERFACE

3.3.1 Underlying Technologies

Two types of information are required to implement the circling interface: (1) real-time mouse movement information and (2) the properties of all user interface (UI) elements currently displayed on the screen. To get mouse movement data, a system events hooking technique for monitoring and intercepting mouse events was used. The retrieved information includes all details associated with mouse movements and operations.

To get the properties of UI elements, the accessibility application programming interface (API) provided by the Windows operating system was used. The accessibility API allows software developers to expose the properties of all standard UI elements (e.g., button, edit box, list control, combo box, icon, scroll, and etc.) to the client software module. The properties of UI elements can be represented by a hierarchical tree structure. Each node in the UI tree corresponds to a UI element on the screen. It can be used not only to indicate what kind of UI element a particular control represents (such as a button, icon, or menu), but also to determine what types of controls are available in the current context and how to interact with them.

To effectively manage the information retrieved from the above two technologies, a dynamic virtual screen object model (SOM) was devised. Each time the user moves the mouse cursor, the SOM updates its record of mouse operations. Whenever any changes in UI elements on the screen are made, the UI tree is updated in the SOM.

3.3.2 Circling Algorithm

The circling interface is a gesture-based interaction technique. In general, gesture recognition requires sophisticated machine learning techniques and large training sets. These approaches prefer to use single-stroke gestures because, as the number of sub-movements that make up a gesture increases, recognition becomes more difficult. In addition, it is also harder for the user to learn, remember, and correctly articulate each gesture (Myers, 1980; Rubine, 1991; Tappert, 1982; Zeleznik, Herndon, & Hughes, 2007). This approach may not be suitable for the circling interface because physically challenged individuals pause more often, resulting in more sub-movements within a gesture (Hwang et al., 2004; Trewin & Pain, 1999a), and the increased sub-movements and latency can compromise system performance and user satisfaction (I.S. MacKenzie & Ware, 1993; Robertson, Card, & Mackinlay, 1989). In addition, because the pointing abilities of individuals with limited motor functions vary widely, not only between individuals of the same input method but also within the same individual over time, it is not easy to ensure that the collected training sets are reliable enough.

To effectively accommodate these challenges, we developed our own software algorithm inspired by the 3D line intersection algorithm of Ronald Goldman (Goldman, 1990). The algorithm recognizes a circling gesture each time the line segments of the visual trail of the pointing cursor interact. Thus, it is not necessary for a circling gesture to be circular. Any closed curve (ranging from a simple triangle to a whole circle) made by the cursor trail can be treated as a trigger for the circling interface. Once a circling motion is recognized, the algorithm determines if the magnification function should be activated or a pointing command should be applied, using the information included in the UI tree as a point of reference.

To deal with errors made by unintentional circling due to uncontrolled body movements, the length of the cursor trail, the size of circled area, and the speed of cursor movement can be adjusted.

3.3.3 Test-bed Software

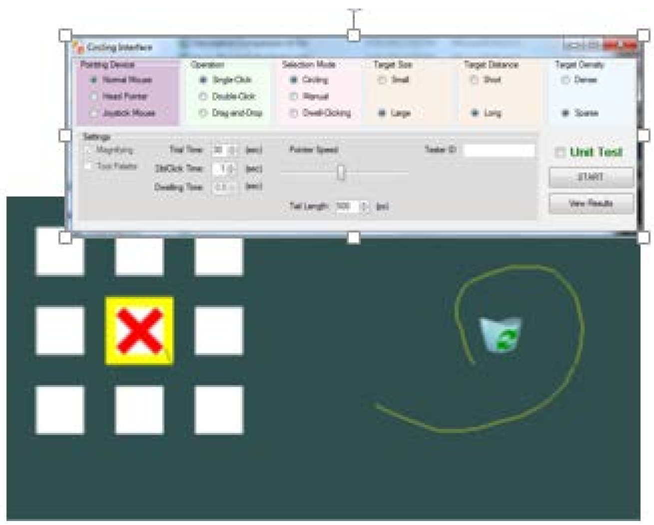

Figure 5. Test-bed Software

As shown in Figure 5, self-contained test-bed software was implemented. The software is compatible with Windows XP, Vista and 7. A dwell-clicking interface was built into the test-bed to provide complete control over the system’s behavior and the ability to record all aspects of a user’s interaction with dwell-clicking. The software is also compatible with commercially available dwell-clicking software, along with any other hardware or software mouse emulators. During task execution, the software records and time-stamps all user interface events. Data are stored in an XML file format for subsequent analysis, which includes the calculation of performance measures (Simeon Keates, Faustina Hwang, Patrick Langdon, P John Clarkson, & Peter Robinson, 2002; I. Scott MacKenzie, Kauppinen, & Silfverberg, 2001).

4.0 EMPIRICAL EVALUATION I: ABLE-BODIED SUBJECTS

4.1 OVERVIEW

An experiment was conducted with able-bodied participants to evaluate the performance of the circling interface on basic pointing tasks. Recruiting subjects without disabilities enabled us to examine multiple combinations of selection methods and input devices, which was not possible with subjects with disabilities. In addition, using able-bodied subjects made it possible to collect data from a larger homogeneous group, which facilitated statistical analyses.

4.2 METHODS

4.2.1 Hypotheses

The following hypotheses were tested:

- Performance time for pointing operations would be different for different combinations of selection method (circling interface, physical mouse buttons, and dwell-clicking software) and input device (normal computer mouse, head pointer, joystick mouse emulator).

- Performance accuracy for pointing operations would be different for different combinations of selection method and input device.
- Perceived workload would be different for different combinations of selection method and input device.

4.2.2 Subjects

Sixteen able-bodied computer users (10 Males, 6 Females; age range 21-50) were recruited. The inclusion criteria were:

- Participant should understand the purpose and the nature of all experimental procedures and tasks;
- Participant should be over 21 years of age;
- Participant should be healthy enough to perform the given pointing tasks;
- Participant should have sufficient visual acuity to perform the given pointing tasks.

Each participant's eligibility was determined based on an interview conducted before a written informed consent form was signed. All participants were recruited, and written consents were obtained, in accordance with the Institutional Review Board at the University of Pittsburgh.

All participants had used computers for at least 5 years, and their average computer usage was more than 20 hours per week. None of them had prior experience with assistive input devices and software.

4.2.3 Experimental Design

The experiment was a 3x3 within-subjects repeated-measures design. Experimental conditions were defined by the following variables:

- 3 Pointing operations
  - Single-click
  - Double-click
  - Drag-and-drop
- 3 Input devices
  - Normal Mouse
  - Head-mounted mouse emulator
  - Joystick Mouse
- 3 Selection methods
  - Manual selection (traditional mouse button)
  - Circling Interface
  - Dwell-Clicking Interface
- 2 Target densities
  - Dense (2 pixels between targets)
  - Sparse (32 pixels between targets)
- 2 Target sizes
  - Large (48 x 48 pixels)
  - Small (16 x 16 pixels)
- 2 Target distances

- Short (176 pixels);
- Long, (336 pixels)

Input device and selection method were treated as the two primary within-subjects factors. Target distance (A) and target size (W) were not treated as separate factors, since Fitts's Law shows that these factors are not independent (I. S. MacKenzie, 1992). Instead, the Index of Difficulty was used as a single factor:

$$(ID) = \log_2(A/W + 1)$$

4.2.4 Equipment

Three input devices were used (see Appendix B): a normal computer mouse[1], a head-mounted mouse emulator[2], and a joystick mouse[3]. For the head-mounted mouse emulator, participants wore a headband to which a reflective infrared dot was attached. All input devices were connected to an Intel-based Macintosh laptop computer[4] running Windows 7[5]. To provide participants with better access to on-screen objects, a 17-inch LCD monitor[6] with a 1200×800 pixel screen resolution connected to the laptop computer through a mini display port to VGA cable adapter was used. Pointing parameters for all input devices were maintained at default values throughout the experiment across all participants as shown in Table 1.

---

[1] Dell Optical Mouse (Dell Computer Corporation, Round Rock, TX
[2] HeadMouse Extreme (Origin Instruments Corporation, Grand Prairie, TX)
[3] Roller II Joystick (AMETEK Company, Ringwood, Hampshire UK)
[4] Macbook Pro, (Apple Inc., Cupertino, CA)
[5] Windows 7 (Microsoft, Redmond, WA)
[6] Acer 173 (Acer America Corporation, San Jose, CA)

Table 1. Pointing Parameters for Able-Bodied Subjects

Parameter	Value
Pointer Speed	10
Pointer Precision	on
Double-Click Time	1000 msec
Snap-To	off
Dwelling Time	800 msec
Permissible Movement	±5 pixels/dwelling time
Circling Trail Length	400 pixels
Maximal Circling Speed	< 100 pixels/tick
Minimal Circumference	> 16 pixels

4.2.5 Data Collection

During each trial, all user interface events were recorded, time-stamped and stored in an XML file format for subsequent analysis (Simeon Keates et al., 2002; I. Scott MacKenzie et al., 2001).

The following performance measures were calculated:

- Time to acquire the target - For manual selection, the time between the start of the trial and the time in which the target is first entered. For the dwell clicking and circling interfaces, the time from when the operation (single-click, double-click, drag) is specified to the time in which the target is first entered or circled.
- Time to specify the operation - For manual selection, the time from target acquisition to when the operation (single-click, double-click, drag) is specified. For dwell selection, the sum of the time between the start of the trial to the time when the operation is selected from the tool palette and the time from target acquisition to when the operation is specified on the target. For the circling interfaces, the time between the start of the trial to the time when the operation is specified by reaching one of screen edges. For the dwell-clinging and circling

interface, single click was the default and did not need to be specified unless the wrong command was active.

- Time to acquire drag goal location - The time from drag initiation to entering or circling the location where the target was to be dragged.
- Clicking errors - The number of times a click occurred outside the target.
- Dragging errors - The number of times the user terminated a drag outside of the drag goal location.
- Task completion time - The total time from the initiation of mouse cursor movement to task completion: the sum of time to acquire the target, time to specify the operation and time to acquire the drag goal location. For manual selection, the task ended as soon as the operation was specified. For dwell selection, the task ended as soon as the operation is specified on the target. For the circling interface, the task ended as soon as the target was acquired.
- Error rate – The sum of clicking errors and dragging errors.

Participants' perceived workload with each selection method and input device was measured using the National Air and Space Administration Task Load Index (NASA-TLX), which is a self-reported, survey-based, validated, multidimensional, subjective workload assessment tool (Sandra G Hart, 2006; S.G. Hart & Staveland, 1988; Noyes & Bruneau, 2007; Xiao, Wang, Wang, & Lan, 2005). The NASA-TLX provides a total workload score based on a weighted average of ratings on six subscales: mental demand, physical demand, temporal demand, performance, effort, and frustration. Each subscale is measured on a scale of 0 to 7 where zero is the lowest possible workload. The mental, physical and temporal demand subscales reflect the demands imposed on the subject and the performance, effort and frustration subscales reflect the interaction of a subject with the task (Sandra G Hart, 2006). A definition of each subscale is provided in Appendix A.1.2.

4.2.6 Experimental Procedures

Participants completed an initial questionnaire regarding basic demographic information, computer experience, and input devices that they currently use. Then, each participant was given the opportunity to practice using the test-bed software to become familiar with each selection method and device for at least 15 minutes.

Participants completed three sessions, one for each pointing operation. The order of pointing operations was the same for all participants (single-click, then double-click, then drag-and-drop). Each session consisted of 24 blocks of trials (3 selection methods x 4 IDs x 2 target densities) and each block consisted of 10 trials of a single condition. The first two trials within each block were treated as practice. The order of conditions within each session was randomized and counter-balanced across subjects.

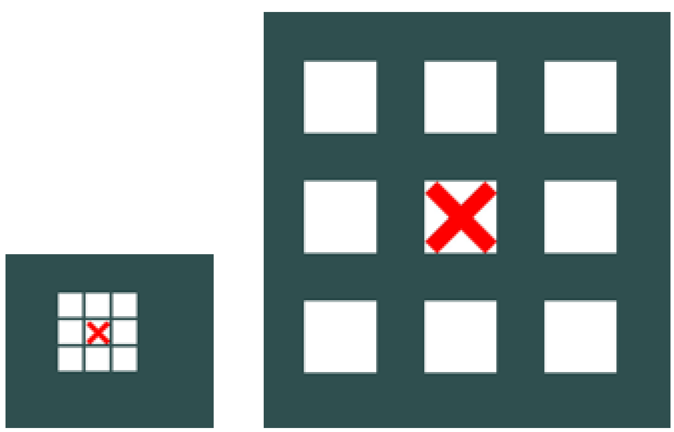

Figure 6. Target and Distractors

A single trial consisted of the acquisition of one target surrounded by eight “distractor” targets as shown in Figure 6. At the start of each trial, the mouse cursor was positioned in the center of the display and a “3–2–1” countdown flashed in the center of the screen. A trial ended when the target had been acquired, at which point the mouse cursor would return to the center of the display in preparation for the next trial. Once all ten targets had been acquired, a new block of ten trials was presented. Participants were allowed to rest between blocks of trials as desired. At the end of each session, participants were asked to indicate their perceived workload with each selection method using NASA-TLX.

4.2.7 Statistical Analysis

The data were analyzed using a multi-factor repeated-measure analysis of variance (ANOVA). Task completion time and error rate were tested for a main effect of input device or selection method. Custom contrast and post-hoc pairwise comparisons applying the Bonferroni adjustment were also performed. The interactions of factors were also tested to examine any systematic pattern of differences between each pair of factors on each dependent variable. For the circling interface, performance speed and accuracy were also examined to determine if there was a learning effect over the three test sessions. Participants’ perceived workload measures for each selection method and input device were compared using a single-factor repeated-measure ANOVA. The statistical significance level for all statistical analyses was set to .05.

4.3 RESULTS

4.3.1 Single-click Operation

4.3.1.1 Task Completion Time

Mauchly's sphericity test was used to examine the task completion time data. The assumption of sphericity for input device was met (Mauchly's W = .812, $\chi^2(2)$ = .0.088, p = .352); the assumption of sphericity for selection method was not met (Mauchly's W = .433, $\chi^2(2)$ = 8.362, p = .015); the assumption of normality was satisfied with one marginal exception of the circling interface with a normal mouse (p = .049); all other assumptions were met. Since sphericity could not be assumed, a Greenhouse-Geisser correction was applied to the test statistics.

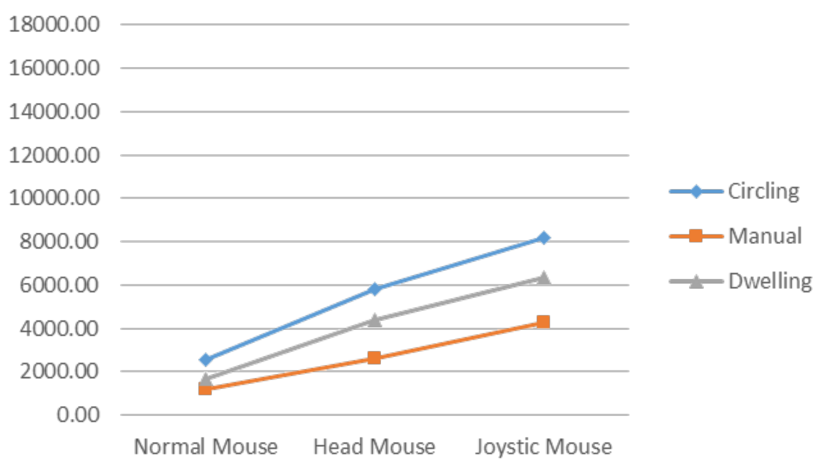

Figure 7. Profile Plots of Device and Method for Single-Click Task Completion Time

Table 2. Tests of Within Subject Effects for Single-Click Task Completion Time

Source	Mean Square	F	p	Partial Eta Squared
Input Device	248574051.583	3869.641	.000	.997
Selection Method	83749675.268	248.702	.000	.958
Input Device * Selection Method	13705408.737	42.759	.000	.795

Table 3. Estimated Marginal means of Input Device for Single-Click Task Completion Time

Input Device	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Normal Mouse	1651.024	18.499	1610.309	1691.740
Head Mouse	4136.522	56.857	4011.382	4261.663
Joystick Mouse	6471.017	49.317	6362.470	6579.563

Table 4. Estimated Marginal means of Selection Method for Single-Click Task Completion Time

Input Device	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Circling Interface	5030.285	85.173	4842.820	5217.750
Manual Selection	2710.555	47.196	2606.677	2814.433
Dwelling Interface	4517.723	73.918	4355.030	4680.416

As shown in Figure 7, Table 2 and Table 3, there was a main effect for input device ($F(1.683, 26.512) = 3869.641$, $p < .001$, $eta^2 = .997$) with a normal mouse (1651 msec), on average, having the shortest task completion time, followed by a head mounted mouse emulator (4136 msec) and a joystick mouse (6471 msec). As shown in Table 2 and Table 4, a main effect for selection method ($F(1.277, 22.043) = 248.702$, $p < .001$, $eta^2 = .958$) was also detected with the circling interface (5030 msec), on average, having the longest task completion time, followed by the dwelling interface (4517 msec) and manual selection (2710 msec) in turn.

As shown in Figure 7 and Table 2, there was an interaction between method and device ($F(2.127, 39.397) = 42.759$, $p < .001$, $eta^2 = .795$). An examination of the means revealed that, whereas the average task completion time did not differ much for a normal mouse among the different selection methods, there were statistically significant differences between each selection method for both a head-mounted mouse emulator and a joystick mouse.

4.3.1.2 Error Rate

Mauchly's sphericity test was used to examine the error rate data. The assumption of sphericity was met for input device (Mauchly's W = .742, $\chi^2(2)$ = .2.984, p = .225) and selection method (Mauchly's W = .839, $\chi^2(2)$ = 1.757, p = .415); the assumption of normality was satisfied based on the Shapiro-Wilk test. Three outliers were detected: all in the circling interface with a head mounted mouse emulator; all other assumptions were met.

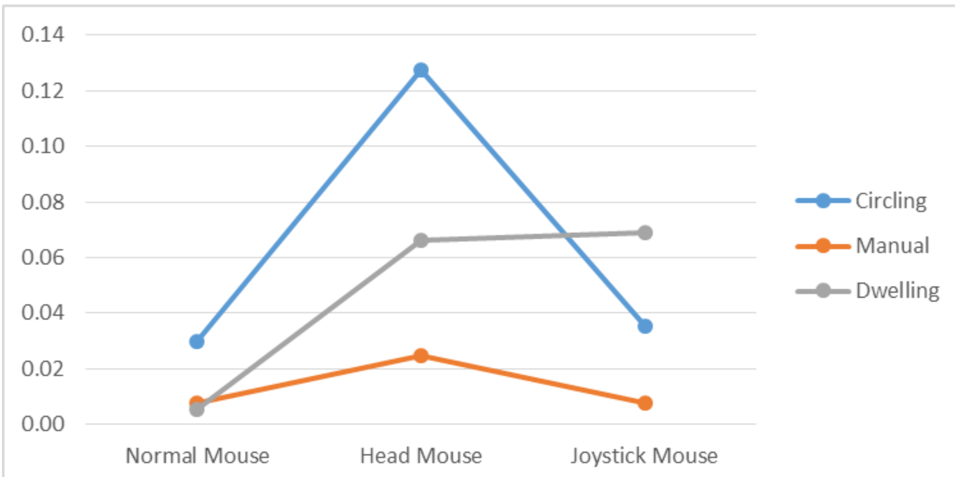

Figure 8. Profile Plots of Input Device and Selection Method for Single-Click Error Rate

Table 5. Tests of Within Subject Effects for Single-Click Error Rate

Source	Mean Square	F	p	Partial Eta Squared
Input Device	.031	35.166	.000	.762
Selection Method	.023	15.537	.000	.585
Input Device * Selection Method	.011	7.777	.000	.414

Table 6. Estimated Marginal means of Input Device for Single-Click Error Rate

Input Device	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Normal Mouse	.015	.003	.009	.022
Head Mouse	.073	.007	.058	.088
Joystick Mouse	.039	.006	.025	.053

Table 7. Estimated Marginal means of Selection Method for Single-Click Error Rate

Input Device	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Circling Interface	.065	.009	.064	.084
Manual Selection	.015	.002	.011	.019
Dwelling Interface	.048	.007	.031	.064

As shown in Figure 8, Table 5 and Table 6, there was a main effect for input device ($F(2, 30) = 35.166$, $p < .001$, $eta^2 = .762$) with a normal mouse (1.5%), on average, having the lowest error rate, followed by the joystick mouse (3.9%) and the head mounted mouse emulator (3.3%) in turn. As shown in Table 5 and Table 7, a main effect for selection method ($F(2, 30) = 15.537$, $p < .001$, $eta^2 = .585$) was also detected with the circling interface (6.5%), on average, having the highest error rate, followed by the dwelling interface (4,8%) and manual selection (1.5%).

As shown in Figure 8 and Table 5, there was also an interaction between selection method and input device ($F(4, 60) = 7.777$, $p < .001$, $eta^2 = .414$). The error rate pattern of the dwelling interface across input devices was not same as those of the other two methods: The error rate of a joystick mouse with the dwelling interface was higher than that of a head-mounted mouse emulator, whereas for the other methods with both the circling interface and a mouse button, the error rate with the joystick mouse was lower than the error rate with the head-mounted mouse emulator. While the difference in error rate across input devices for manual selection was relatively small, the error rates for the other two selection methods were significantly larger. In particular, the circling interface with a head-mounted mouse emulator had a significantly higher error rate.

4.3.1.3 Perceived Workload

The Shapiro-Wilk normality test was used to examine the perceived workload data. The assumption of normality was not satisfied, so the collected data was analyzed using Friedman's method. The statistical results are shown in Table 8.

Table 8. Statistical Significance for each comparison – Single-Click Perceived Workload

	Normal Mouse			Head Mouse			Joystick Mouse		
	Circle vs. Manual	Circle vs. Dwell	Manual vs. Dwell	Circle vs. Manual	Circle vs. Dwell	Manual vs. Dwell	Circle vs. Manual	Circle vs. Dwell	Manual vs. Dwell
Total	**.002**	**.002**	**.010**	**.003**	**.015**	.583	**.002**	**.028**	*.050*
Mental	**.003**	**.014**	**.011**	**.003**	**.034**	.066	**.008**	**.017**	.786
Physical	**.002**	**.003**	.078	**.002**	**.023**	.441	**.002**	.092	.066
Temporal	**.002**	**.003**	**.028**	**.003**	**.012**	.235	**.002**	.068	.092
Performance	**.002**	**.005**	**.017**	**.059**	.212	.667	**.005**	.112	.810
Effort	**.002**	**.005**	.074	**.045**	**.116**	.888	**.002**	**.015**	.115
Frustration	**.002**	**.003**	.109	**.003**	**.043**	.550	**.003**	.058	**.024**

(Statistically significant differences in **bold**. Marginally significant differences in *italic*.)

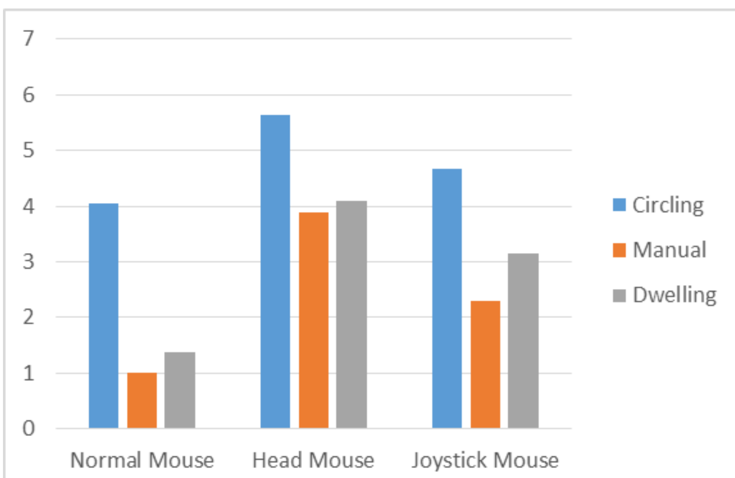

Figure 9. Single-Click Task Load Index - Total Workload

As shown in Figure 9, the circling interface imposed a significantly higher workload than manual selection and the dwelling interface for all three devices. When using the joystick mouse, manual selection imposed a marginally lower workload than the dwelling interface.

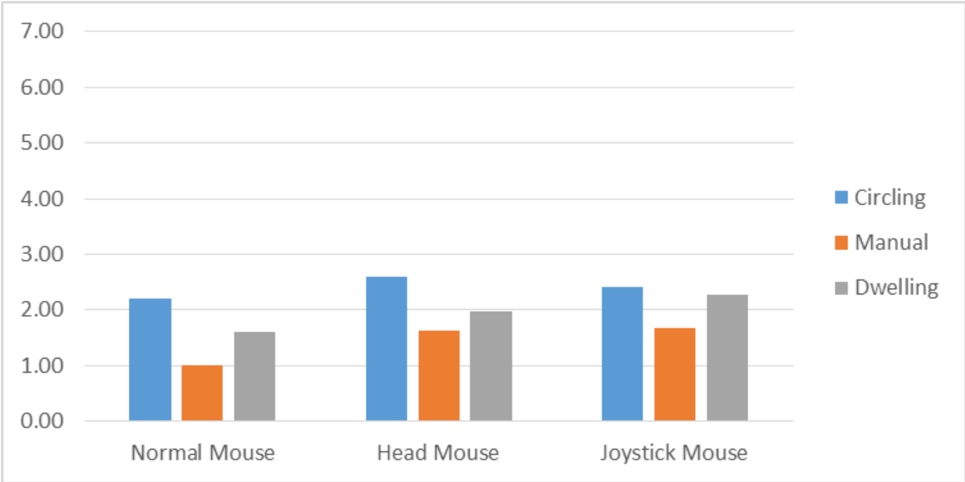

Figure 10. Single-Click Task Load Index – Mental Demand

As shown in Figure 10, the circling interface required significantly more mental effort than manual selection and the dwelling interface for all three devices. When using the normal mouse, the dwelling interface required significantly more mental efforts than manual selection.

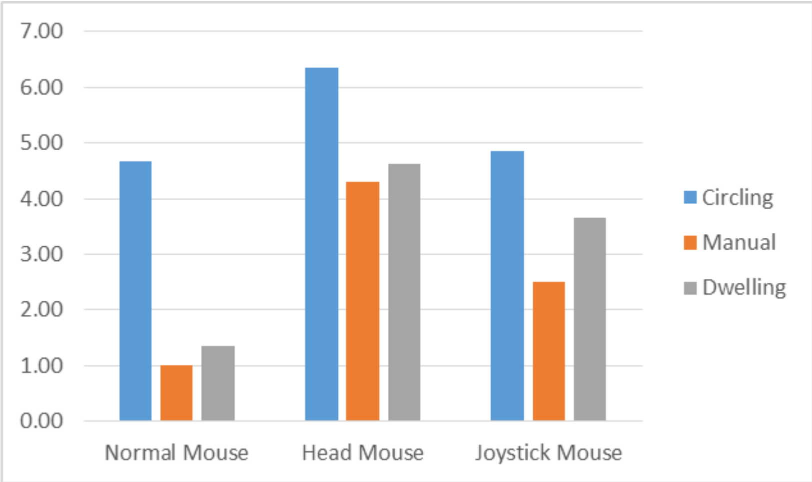

Figure 11. Single-Click Task Load Index – Physical Demand

As shown in Figure 11, the circling interface required significantly more physical effort than manual selection for all three devices and significantly more physical effort than the dwelling interface for the normal mouse and the head-mounted mouse emulator.

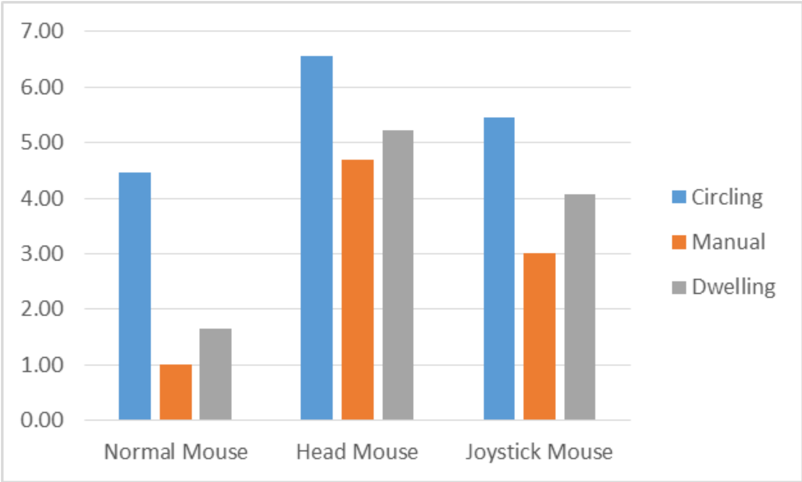

Figure 12. Single-Click Task Load Index – Temporal Demand

As shown in Figure 12, the circling interface was significantly more temporally demanding than manual selection for all three devices and the dwelling interface for the normal mouse and head-mounted mouse emulator.

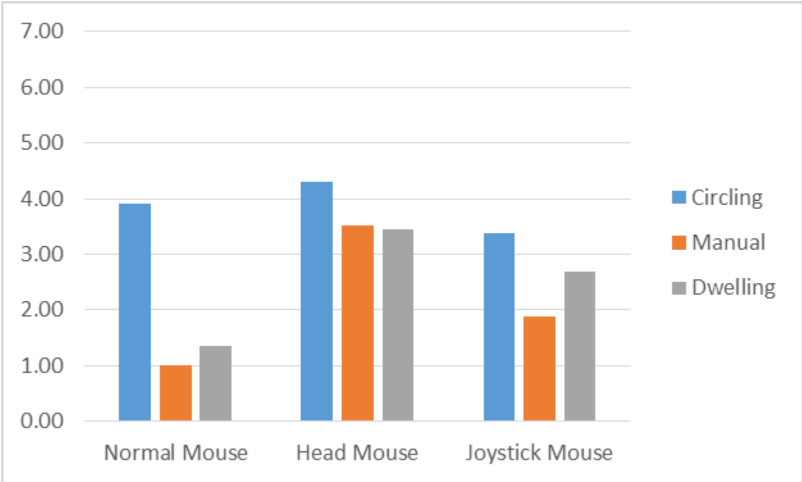

Figure 13. Single-Click Task Load Index – Performance

As shown in Figure 13, the circling interface imposed significantly more performance burden than manual selection for the normal mouse and joystick mouse. Dwell selection imposed significantly more performance burden than manual selection when using the normal mouse.

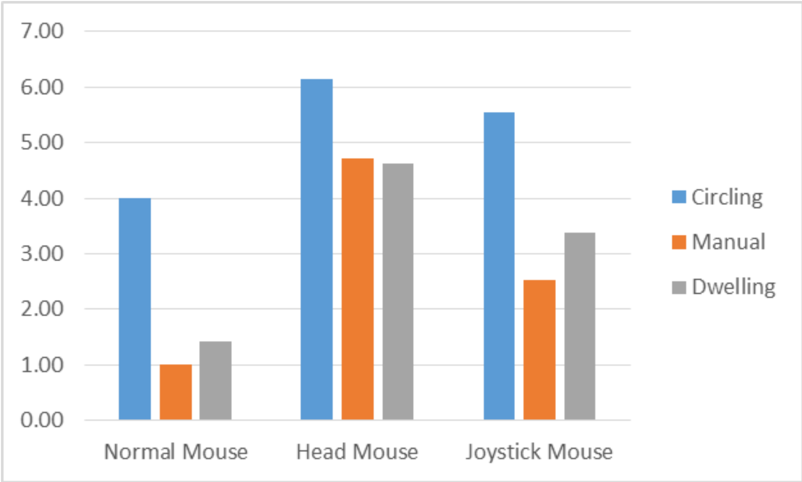

Figure 14. Single-Click Task Load Index – Effort

As shown in Figure 14, the circling interface required significantly more effort than manual selection for all three devices and more effort than dwell selection for the normal mouse and joystick mouse.

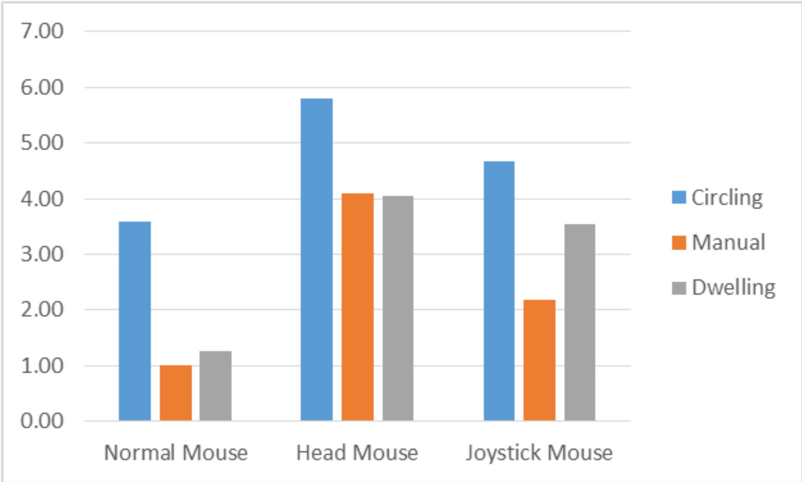

Figure 15. Single-Click Task Load Index – Frustration

As shown in Figure 15, the circling interface was significantly more stressful than manual selection for all three devices and was significantly more stressful than the dwelling interface for the normal mouse and head-mounted mouse emulator. Manual selection was significantly less stressful than the circling interface and the dwelling interface when using a joystick mouse.

4.3.2 Double-click Operation

4.3.2.1 Task Completion Time

Mauchly's sphericity test was used to examine the task completion time data. The assumption of sphericity was not met for input device (Mauchly's W = .334, $\chi^2(2)$ = 10.956., p = .004) or selection method (Mauchly's W = .638, $\chi^2(2)$ = 4.500, p = .105). The assumption of normality was satisfied based on the Shapiro-Wilk test. Two outliers were detected in the circling interface with a head-mounted mouse emulator. All other assumptions were met. Since sphericity could not be assumed, a Greenhouse-Geisser correction was applied to the test statistics.

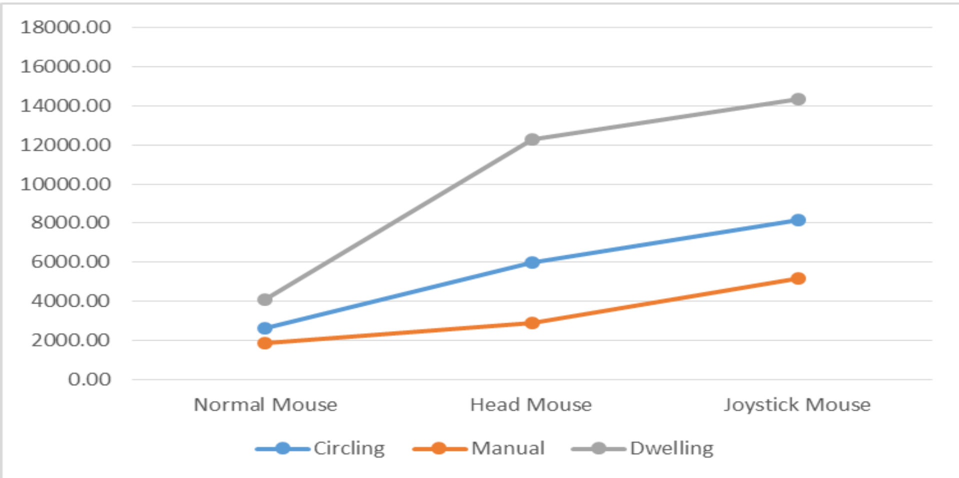

Figure 16. Profile Plots of Device and Method for Double-Click Task Completion Time

Table 9. Tests of Within Subject Effects for Double-Click Task Completion Time

Source	Mean Square	F	p	Partial Eta Squared
Input Device	504234783.117	889.065	.000	.988
Selection Method	293639259.497	560.900	.000	.981
Input Device * Selection Method	38969331.152	62.463	.000	.850

Table 10. Estimated Marginal means of Input Device for Double-Click Task Completion Time

Input Device	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Normal Mouse	2850.751	48.967	2742.977	2958.526
Head Mouse	7053.064	204.516	6602.929	7503.200
Joystick Mouse	9215.975	108.979	8976.113	9455.837

Table 11. Estimated Marginal means of Selection Method for Double-Click Task Completion Time

Input Device	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Circling Interface	5599.333	67.751	5450.214	5748.453
Manual Selection	3301.797	48.830	3194.322	3409.272
Dwelling Interface	10218.660	132.722	9926.541	10510.779

As shown in Figure 16, Table 9 and Table 10, there was a main effect for input device ($F(2, 30) = 889.065$, $p < .001$, $eta^2 = .988$) with a normal mouse (2859 msec), on average, having the shortest task completion time, followed by a head mounted mouse emulator (8636 msec) and a joystick mouse (9682 msec) in turn. As shown in Table 9 and Table 11, a main effect for selection method ($F(2, 30) = 560.900$, $p < .001$, $eta^2 = .981$) was also detected with the dwelling interface (10218 msec), on average, having the longest task completion time, followed by the circling interface (6695 msec) and manual selection (4563 msec).

As shown in Figure 16 and Table 9, there was an interaction between selection method and device ($F(4, 60) = 62.463$, $p < .001$, $eta^2 = .850$). While task completion time did not differ much for the normal mouse across the different selection methods, there were larger differences between each selection method for both the head-mounted mouse emulator and the joystick mouse. Task completion time for the dwelling interface with a head-mounted mouse emulator and a joystick mouse was significantly slower than task completion time for the circling interface and manual selection.

4.3.2.2 Error Rate

Mauchly's sphericity test was used to examine the error rate data. The assumption of sphericity was met for input device (Mauchly's W = .755, $\chi^2(2)$ = .2.805, p = .246) and selection method (Mauchly's W = .774, $\chi^2(2)$ = 2.568, p = .277). The assumption of normality was satisfied based on the Shapiro-Wilk test. One outlier was found in the dwelling interface with a head-mounted mouse emulator. All other assumptions were met.

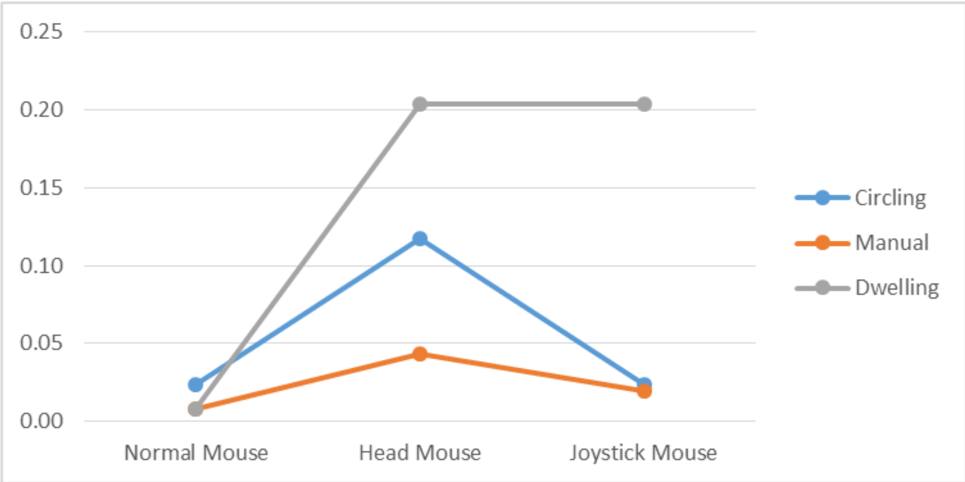

Figure 17. Profile Plots of Input Device and Selection Method for Double-Click Error Rate

Table 12. Tests of Within Subject Effects for Double-Click Error Rate

Source	Mean Square	F	p	Partial Eta Squared
Input Device	.107	74.165	.000	.871
Selection Method	.125	93.631	.000	.895
Input Device * Selection Method	.041	28.972	.000	.725

Table 13. Estimated Marginal means of Input Device for Double-Click Error Rate

Input Device	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Normal Mouse	.014	.003	.007	.022
Head Mouse	.112	.009	.102	.143
Joystick Mouse	.084	.006	.071	.096

Table 14. Estimated Marginal means of Selection Method for Double-Click Error Rate

Input Device	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Circling Interface	.056	.006	.043	.069
Manual Selection	.025	.003	.018	.032
Dwelling Interface	.139	.009	.120	.159

As shown in Figure 17, Table 12 and Table 13, there was a main effect for input device ($F(2, 30) = 74.165$, $p < .001$, $eta^2 = .871$), with the normal mouse (1.4%), having the lowest error rate, followed by the joystick mouse (8.4%) and the head-mounted mouse emulator (11.2%). As shown in Table 12 and Table 14, a main effect for selection method ($F(2, 30) = 93.631$, $p < .001$, $eta^2 = .895$) was also detected with the dwelling interface (13.9%), on average, having the highest error rate, followed by the circling interface (5.6%) and manual selection (2.5%).

As shown in Figure 17 and Table 12, there was also an interaction between selection method and input device ($F(4, 60) = 28.972$, $p < .001$, $eta^2 = .725$). The error rate with the head-mounted mouse emulator was significantly higher than with the other two devices.

4.3.2.3 Perceived Workload

The Shapiro-Wilk normality test was used to examine the perceived workload data. The assumption of normality was not satisfied, so the data was analyzed using Friedman's method. The statistical results are shown in Table 15.

Table 15. Statistical Significance for each comparison – Double-Click Perceived Workload

	Normal Mouse			Head Mouse			Joystick Mouse		
	Circle vs. Manual	Circle vs. Dwell	Manual vs. Dwell	Circle vs. Manual	Circle vs. Dwell	Manual vs. Dwell	Circle vs. Manual	Circle vs. Dwell	Manual vs. Dwell
Total	**.002**	.814	**.004**	**.002**	.099	**.002**	**.002**	**.005**	**.002**
Mental	**.007**	.496	**.007**	**.005**	.108	**.002**	**.003**	.715	**.003**
Physical	**.011**	.086	**.002**	**.002**	**.029**	**.002**	**.003**	**.003**	**.002**
Temporal	**.003**	.201	**.005**	**.005**	.285	**.003**	**.002**	.068	**.003**
Performance	**.005**	.192	**.016**	.070	.102	.074	**.005**	.722	**.002**
Effort	**.002**	.812	**.007**	*.050*	**.022**	**.002**	**.002**	**.026**	**.004**
Frustration	**.003**	.953	**.023**	**.007**	**.026**	**.002**	**.007**	**.004**	**.002**

(Statistically significant differences in **bold**. Marginally significant differences in *italic*.)

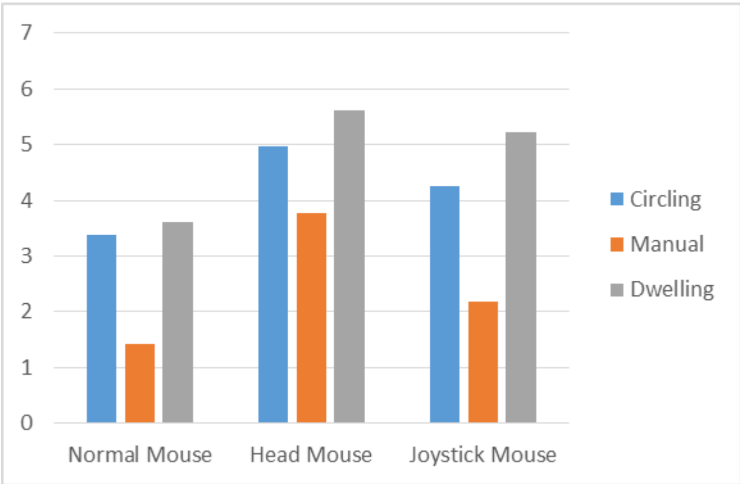

Figure 18. Double-Click Task Load Index - Total Workload

As shown in Figure 18, the circling interface imposed a significantly higher workload than manual selection for all three devices and a significantly higher workload than the dwelling interface for the joystick mouse. The dwelling interface imposed a significantly higher workload than manual selection for all three devices.

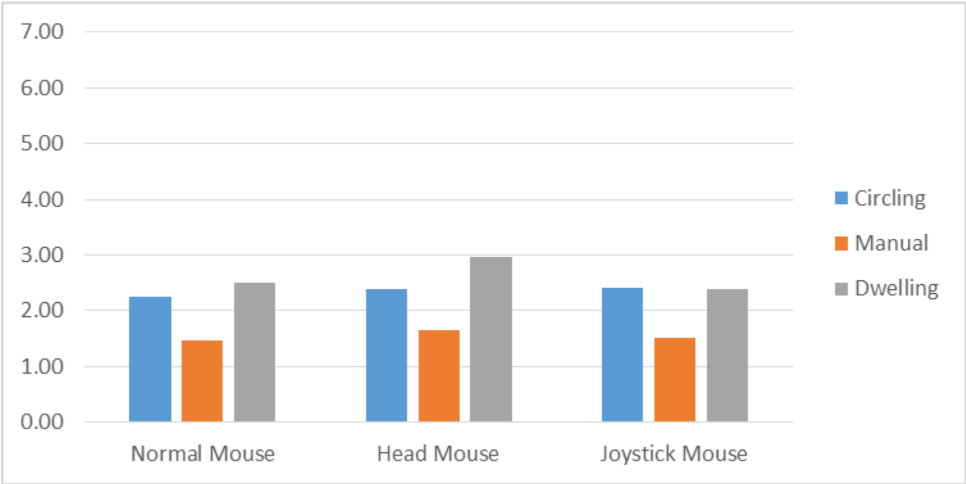

Figure 19. Double-Click Task Load Index – Mental Demand

As shown in Figure 19, the circling interface required significantly more mental effort than manual selection and the dwelling interface for all three devices.

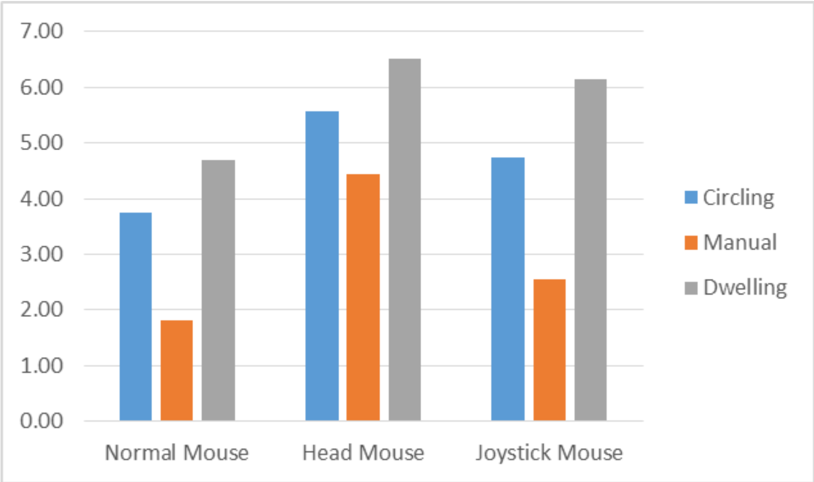

Figure 20. Double-Click Task Load Index – Physical Demand

As shown in Figure 20, the circling interface required significantly more physical effort than manual selection for all three devices. The circling interface required significantly less physical effort than the dwelling interface for the head-mounted mouse emulator and joystick mouse. The dwelling interface required significantly more physical effort than manual selection for all three devices.

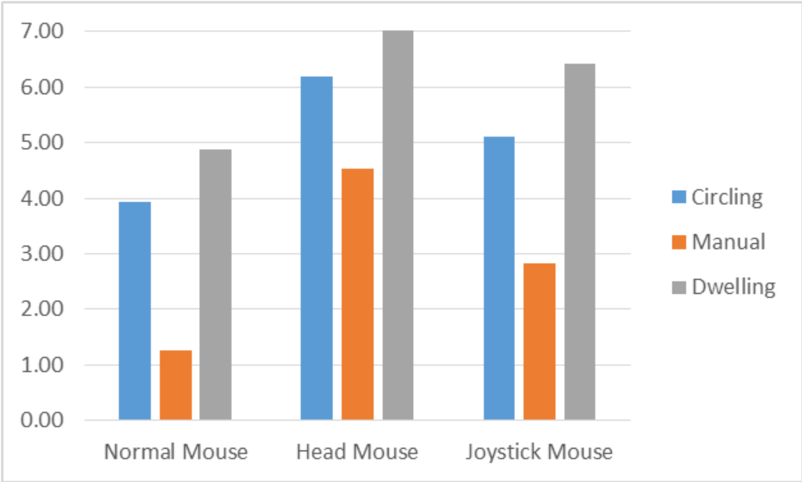

Figure 21. Double-Click Task Load Index – Temporal Demand

As shown in Figure 21, the circling interface required significantly more temporally demanding than manual selection for all three devices. The circling interface required significantly less temporally demanding than the dwelling interface for the head-mounted mouse emulator and joystick mouse. The dwelling interface required significantly more temporally demanding than manual selection for all three devices.

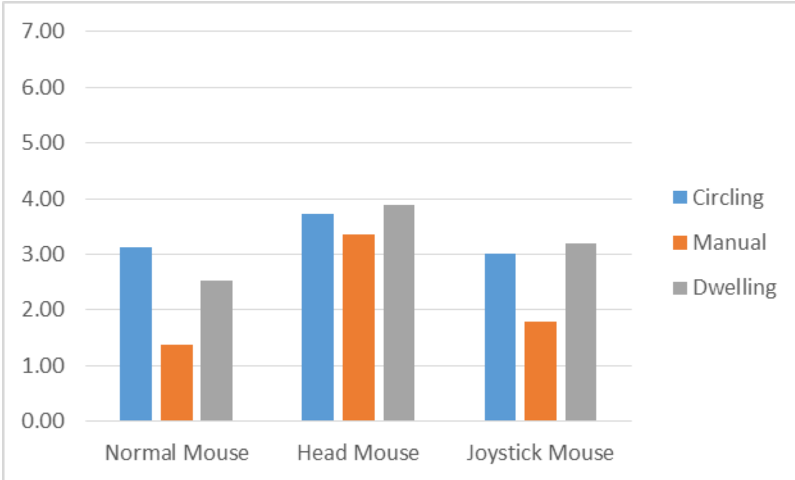

Figure 22. Double-Click Task Load Index – Performance

As shown in Figure 22, the circling interface imposed significantly more performance burden than manual selection for the normal mouse and joystick mouse. Dwell selection imposed significantly more performance burden than manual selection when using the normal mouse.

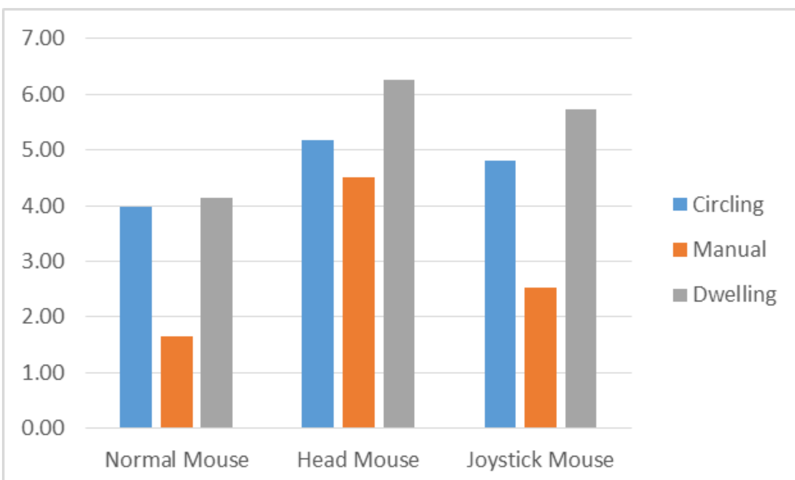

Figure 23. Double-Click Task Load Index – Effort

As shown in Figure 23, the circling interface required significantly more effort than manual selection for all three devices and less effort than dwell selection for both the head-mounted mouse emulator and joystick mouse.

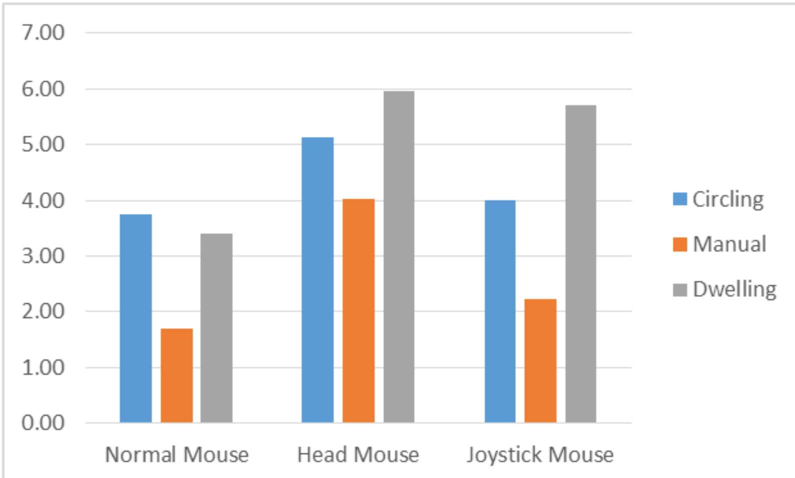

Figure 24. Double-Click Task Load Index – Frustration

As shown in Figure 24, the circling interface was significantly more stressful than manual selection for all three devices and was significantly less stressful than the dwelling interface for the head-mounted mouse emulator and joystick mouse. Manual selection was significantly less stressful than the circling interface and the dwelling interface for all input devices.

4.3.3 Drag-and-Drop Operation

4.3.3.1 Task Completion Time

Mauchly's sphericity test was used to examine the task completion time data. The assumption of sphericity for input device was met (Mauchly's W = .735, $\chi^2(2) = 3.083$, p = .214); the assumption of sphericity for selection method was not met (Mauchly's W = .580, $\chi^2(2) = 5.454$, p = .065). The assumption of normality was satisfied based on the Shapiro-Wilk test. All other assumptions were met.

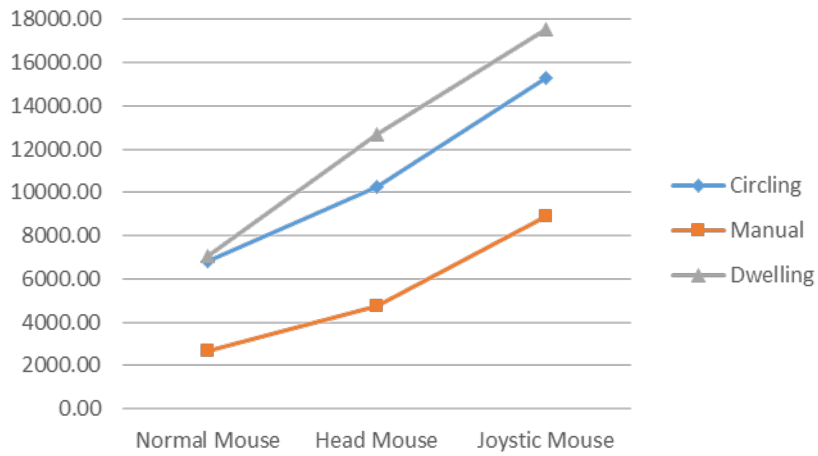

Figure 25. Profile Plots of Device and Method for Drag-and-Drop Task Completion Time

Table 16. Tests of Within Subject Effects for Drag-and-Drop Task Completion Time

Source	Mean Square	F	p	Partial Eta Squared
Input Device	639302041.169	1444.024	.000	.992
Selection Method	463141243.151	1592.484	.000	.993
Input Device * Selection Method	16524825.574	47.995	.000	.814

Table 17. Estimated Marginal means of Input Device for Drag-and-Drop Task Completion Time

Input Device	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Normal Mouse	5347.140	65.411	5203.171	5491.109
Head Mouse	9052.669	88.541	8857.793	9247.546
Joystick Mouse	13755.586	149.746	13425.997	14085.176

Table 18. Estimated Marginal means of Selection Method for Drag-and-Drop Task Completion Time

Input Device	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Circling Interface	10267.977	105.595	10035.564	10500.391
Manual Selection	5439.359	42.389	5346.062	5532.656
Dwelling Interface	12448.060	114.701	12195.604	12700.516

As shown in Figure 25, Table 16 and Table 17, there was a main effect for input device ($F(2, 30) = 1444.024$, $p < .001$, $eta^2 = .992$) with the normal mouse (5347 msec), on average, having the shortest task completion time, followed by the head-mounted mouse emulator (9052 msec) and the joystick mouse (13755 msec). As shown in Table 16 and Table 18, a main effect for selection method ($F(2, 30) = 1592.484$, $p < .001$, $eta^2 = .993$) was also detected with the dwelling interface (12448 msec), on average, having the longest task completion time, followed by the circling interface (10267 msec) and manual selection (5439 msec) in turn.

As shown in Figure 25 and Table 16, there was an interaction between method and device ($F(4, 60) = 47.995$, $p < .001$, $eta^2 = .814$). An examination of the means revealed that, whereas the average task completion time was almost same for a normal mouse between the circling interface and the dwelling interface, there were statistically significant differences for both a head-mounted mouse emulator and a joystick mouse between the circling interface and the dwelling interface.

4.3.3.2 Error Rate

Mauchly's sphericity test was used to examine the error rate data. The assumption of sphericity was met for input device (Mauchly's W = .979, $\chi^2(2) = .211$, p = .900) and selection method (Mauchly's W = .657, $\chi^2(2) = 4.201$, p = .122). The assumption of normality was satisfied based on the Shapiro-Wilk test. Two outliers were found in the circling interface with a normal mouse. All other assumptions were met.

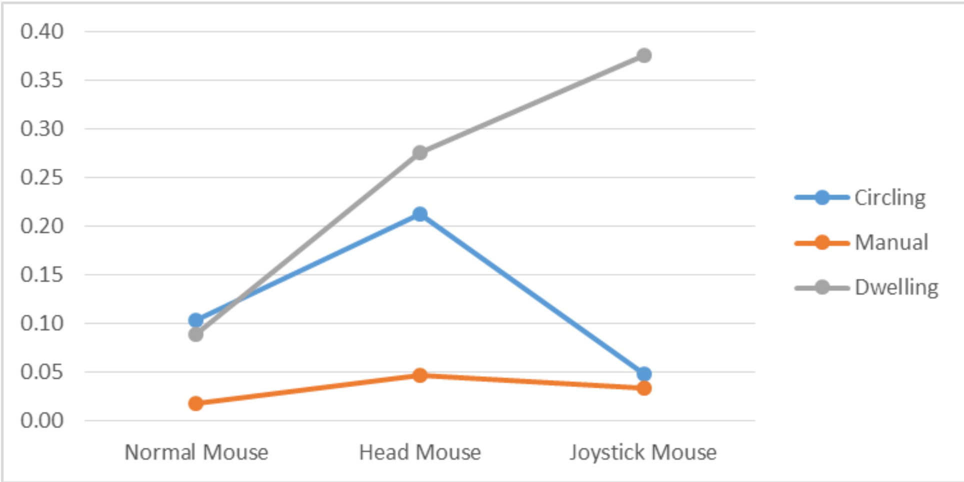

Figure 26. Profile Plots of Input Device and Selection Method for Drag-and-Drop Error Rate

Table 19. Tests of Within Subject Effects for Drag-and-Drop Error Rate

Source	Mean Square	F	p	Partial Eta Squared
Input Device	.112	45.591	.000	.806
Selection Method	.412	231.945	.000	.955
Input Device * Selection Method	.114	53.304	.000	.829

Table 20. Estimated Marginal means of Input Device for Drag-and-Drop Error Rate

Input Device	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Normal Mouse	.072	.009	.052	.092
Head Mouse	.179	.007	.163	.195
Joystick Mouse	.153	.007	.137	.170

Table 21. Estimated Marginal means of Selection Method for Drag-and-Drop Error Rate

Input Device	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Circling Interface	.122	.009	.103	.141
Manual Selection	.035	.003	.028	.041
Dwelling Interface	.248	.008	.230	.265

As shown in Figure 26, Table 19 and Table 20, there was a main effect for input device ($F(2, 30) = 45.591$, $p < .001$, $eta^2 = .806$) with a normal mouse (7.2%), on average, having the lowest error rate, followed by the joystick mouse (15.3%) and the head-mounted mouse emulator (17.9%) in turn. As shown in Table 19 and Table 21, a main effect for selection method ($F(2, 30) = 231.945$, $p < .001$, $eta^2 = .955$) was also detected with the dwelling interface (24.8%), on average, having the highest error rate, followed by the circling interface (12.2%) and manual selection (3.5%).

As shown in Figure 26 and Table 19, there was also an interaction between selection method and input device ($F(4, 60) = 53.304$, $p < .001$, $eta^2 = .829$): The error rate pattern of the dwelling interface across input devices was not the same as those of the other two methods. The error rate for the joystick mouse with the dwelling interface was higher than that of the head-mounted mouse emulator, whereas for the other methods the error rates of the joystick mouse were lower than that of the head-mounted mouse emulator. While the difference in error rate across input devices for manual selection was relatively small, those of the other two methods were significantly larger. Both the circling interface and dwelling interface with a head-mounted mouse emulator had a significantly higher error rate.

4.3.3.3 Perceived Workload

The Shapiro-Wilk normality test was used to examine the perceived workload data. The assumption of normality was not satisfied, so the collected data was analyzed using Friedman's method. The statistical results are shown in Table 22.

Table 22. Statistical Significance for each comparison – Drag-and-Drop Perceived Workload

	Normal Mouse			Head Mouse			Joystick Mouse		
	Circle vs. Manual	Circle vs. Dwell	Manual vs. Dwell	Circle vs. Manual	Circle vs. Dwell	Manual vs. Dwell	Circle vs. Manual	Circle vs. Dwell	Manual vs. Dwell
Total	**.002**	.117	**.008**	**.003**	.158	**.004**	**.002**	**.013**	**.002**
Mental	**.005**	.917	**.007**	**.002**	.832	**.003**	**.005**	.180	**.005**
Physical	**.002**	.327	**.003**	**.006**	.034	**.003**	**.002**	**.012**	**.002**
Temporal	**.004**	.422	**.007**	**.003**	.260	**.026**	**.003**	**.012**	**.002**
Performance	**.002**	.066	**.005**	**.005**	.664	.074	**.003**	.906	**.003**
Effort	**.009**	.312	**.041**	**.007**	.112	**.031**	**.003**	.11	**.003**
Frustration	**.002**	.314	**.014**	**.005**	.167	**.002**	**.007**	**.007**	**.002**

(Statistically significant differences in **bold**. Marginally significant differences in *italic*.)

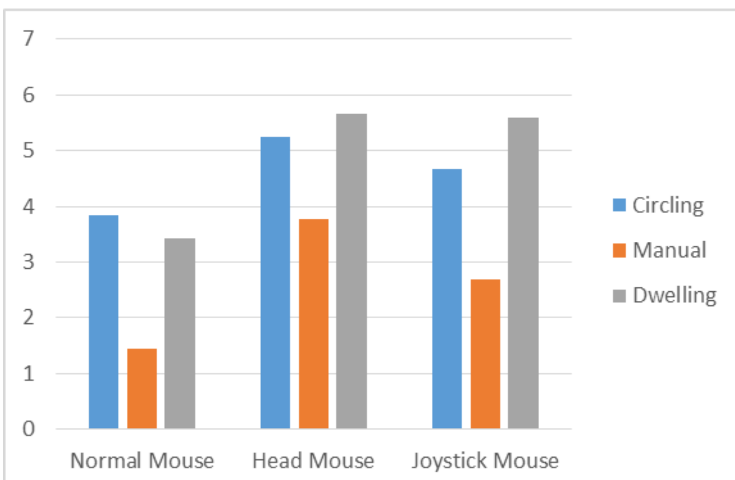

Figure 27. Drag-and-Drop Task Load Index - Total Workload

As shown in Figure 27, the circling interface imposed a significantly higher workload than manual selection for all three devices and a significantly lower workload than the dwelling interface for the joystick mouse. The dwelling interface imposed a significantly higher workload than manual selection for all three devices.

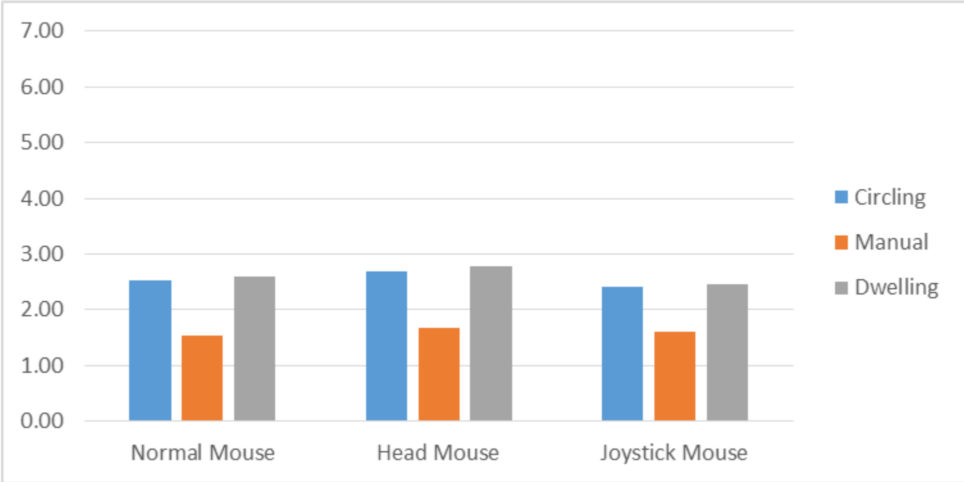

Figure 28. Drag-and-Drop Task Load Index – Mental Demand

As shown in Figure 28, manual selection required significantly less mental effort than the circling interface and dwelling interface for all three devices.

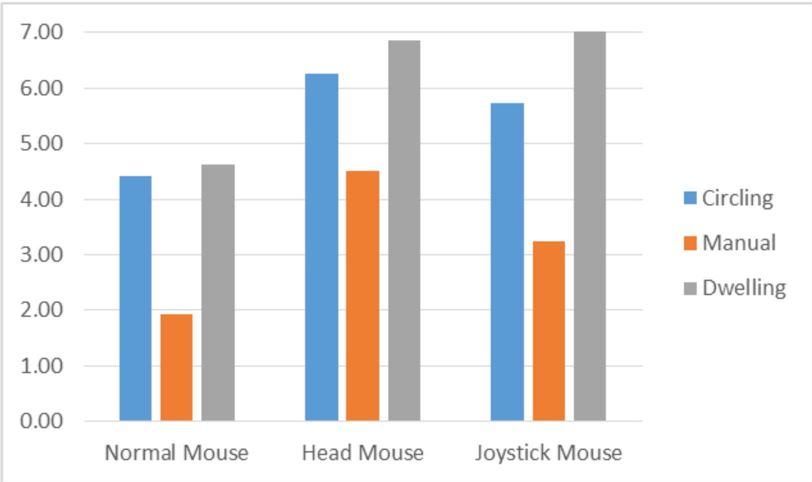

Figure 29. Drag-and-Drop Task Load Index – Physical Demand

As shown in Figure 29, the circling interface required significantly more physical effort than manual selection for all three devices. The circling interface required significantly less physical effort than the dwelling interface for the head-mounted mouse emulator and joystick mouse. The dwelling interface required significantly more physical effort than manual selection for all three devices.

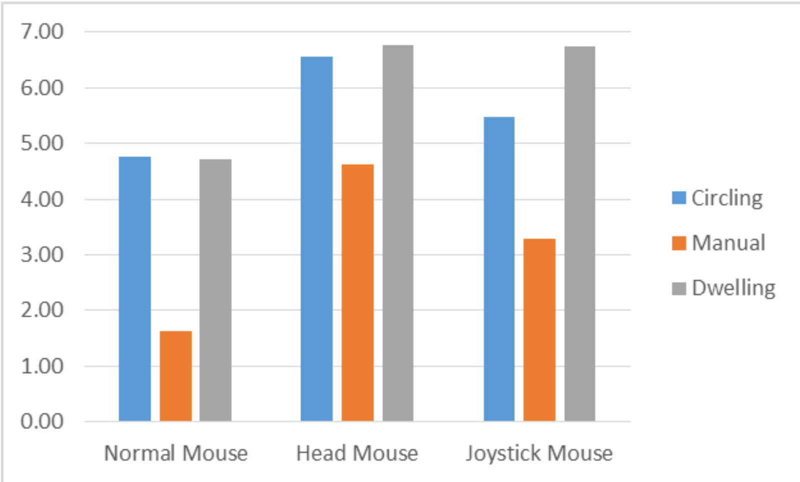

Figure 30. Drag-and-Drop Task Load Index – Temporal Demand

As shown in Figure 30, the circling interface was significantly more temporally demanding than manual selection for all three devices. The circling interface was significantly less temporally demanding than the dwelling interface for the head-mounted mouse emulator and joystick mouse. The dwelling interface was significantly more temporally demanding than manual selection for all three devices.

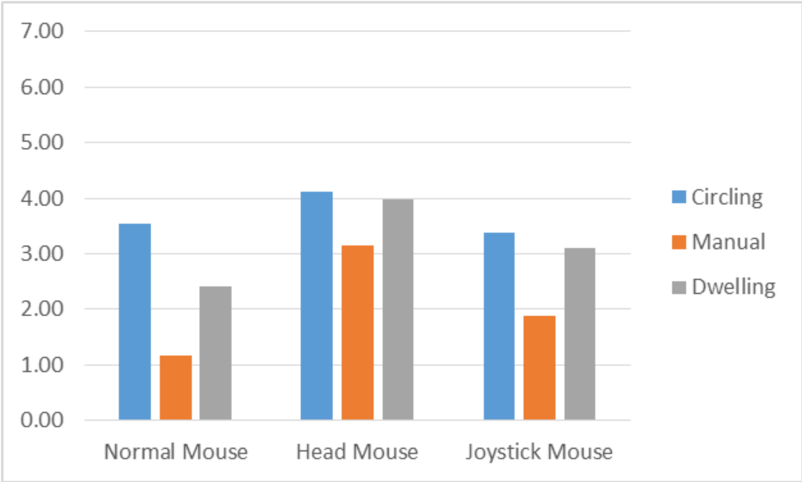

Figure 31. Drag-and-Drop Task Load Index – Performance

As shown in Figure 31, the circling interface imposed significantly more performance burden than manual selection for the normal mouse and joystick mouse. Dwell selection imposed significantly more performance burden than manual selection when using the normal mouse and joystick mouse.

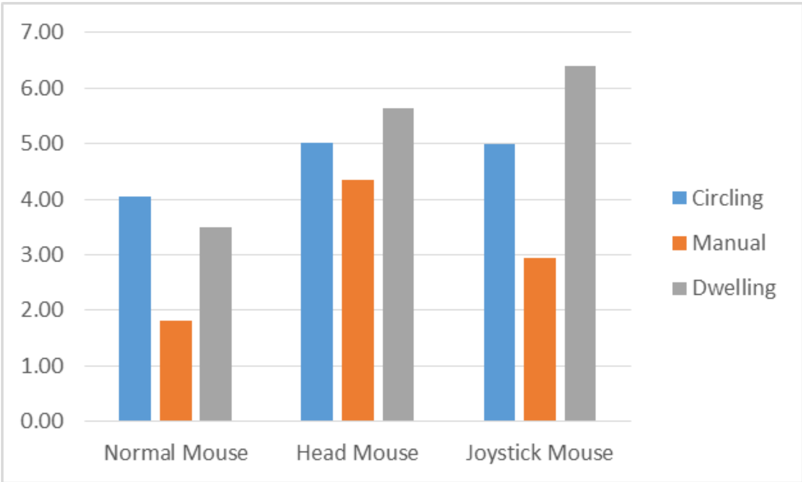

Figure 32. Drag-and-Drop Task Load Index – Effort

As shown in Figure 32, manual selection required significantly less effort than dwell selection and the circling interface for all three devices.

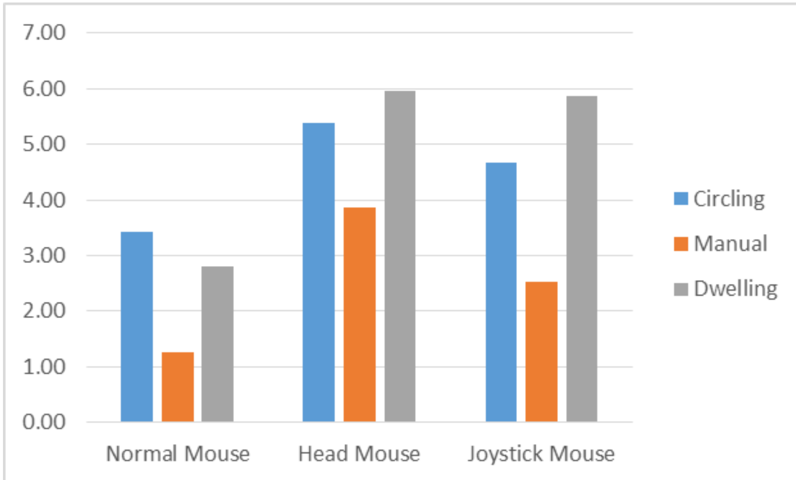

Figure 33. Drag-and-Drop Task Load Index – Frustration

As shown in Figure 33, manual selection was significantly less stressful than the circling interface and dwell selection for all three devices and the circling interface was significantly less stressful than the dwelling interface for the joystick mouse.

4.3.4 Overall

4.3.4.1 Task Completion Time

Mauchly's sphericity test was used to examine the task completion time data across all pointing operations. The assumption of sphericity was met for input device (Mauchly's W = .983, $\chi^2(2)$ = .168, p = .919) and selection method (Mauchly's W = .637, $\chi^2(2)$ = 4.516, p = .105). The assumption of normality was satisfied based on the Shapiro-Wilk test. Four outliers were detected: two in the circling interface and two in manual selection. All other assumptions were met.

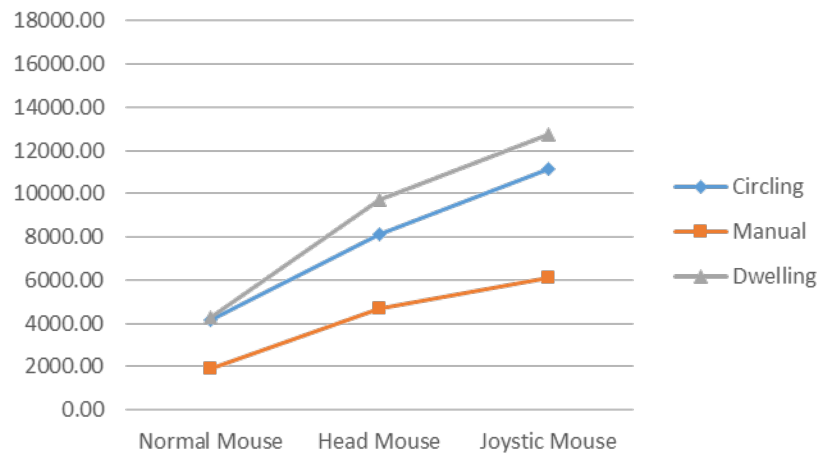

Figure 34. Profile Plots of Device and Method for Overall Task Completion Time

Table 23. Tests of Within Subject Effects for Overall Task Completion Time

Source	Mean Square	F	p	Partial Eta Squared
Input Device	407831619.03	5451.25	.000	.998
Selection Method	214309011.28	1530.07	.000	.993
Input Device * Selection Method	16907797.35	124.37	.000	0.919

Table 24. Estimated Marginal means of Input Device for Overall Task Completion Time

Input Device	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Normal Mouse	3285.758	29.932	3219.879	3351.638
Head Mouse	7373.828	51.483	7260.515	7487.141
Joystick Mouse	9961.395	61.401	9826.252	10096.538

Table 25. Estimated Marginal Means of Selection Method for Overall Task Completion Time

Input Device	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Circling Interface	7330.052	60.729	7196.388	7463.716
Manual Selection	4237.808	34.841	4161.123	4314.494
Dwelling Interface	9053.121	77.715	8882.072	9224.170

As shown in Figure 34, Table 23 and Table 24, there was a main effect for input device ($F(2, 30) = 5451.249$, $p < .001$, $eta^2 = .998$). Across all pointing operations, a normal mouse (3285 msec), on average, had the shortest completion time, followed by a head-mounted mouse emulator (7373 msec) and a joystick mouse (9961 msec). As shown in Table 23 and Table 25, a main effect for selection method ($F(2, 30) = 1530.074$, $p < .001$, $eta^2 = .993$) was also detected. Across all pointing operations, the dwelling interface (9053 msec), on average, had the longest task completion time, followed by the circling interface (7330 msec) and manual selection (4237 msec) in turn.

As shown in Figure 34 and Table 23, there was an interaction between selection method and input device ($F(4, 60) = 124.374$, $p < .001$, $eta^2 = .919$). While the average task completion time was almost same for the normal mouse between the circling interface and the dwelling interface, there were statistically significant differences for both the head-mounted mouse emulator and the joystick mouse between the circling interface and dwelling interface.

4.3.4.2 Error Rate

Mauchly's sphericity test was used to examine the error rate data across all pointing operations. The assumption of sphericity was met for input device (Mauchly's $W = .993$, $\chi^2(2) = .073$, $p = .964$) and selection method (Mauchly's $W = .878$, $\chi^2(2) = 1.305$, $p = .521$). The assumption of normality was satisfied based on the Shapiro-Wilk test. Eight outliers were found: three in manual selection with a normal mouse, three in the dwelling interface with a normal mouse, and two in the circling interface with a head-mounted mouse emulator. All other assumptions were met.

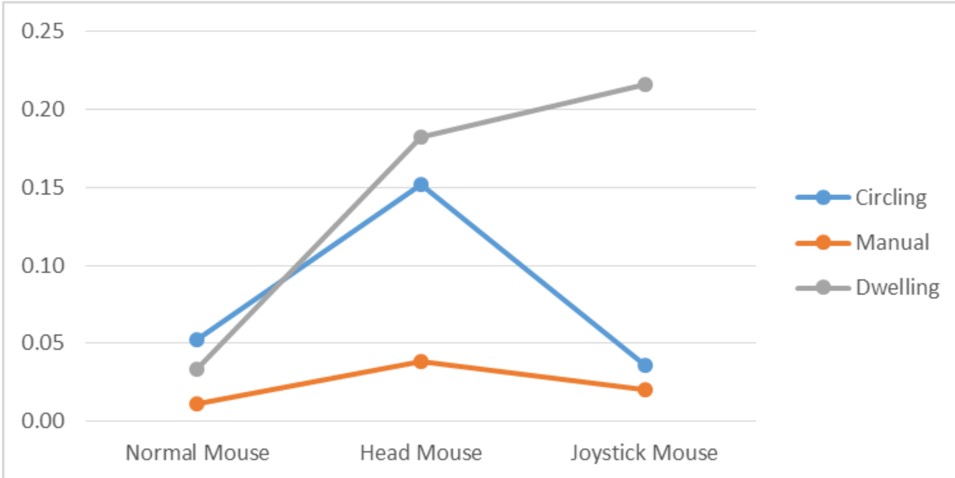

Figure 35. Profile Plots of Input Device and Selection Method for Overall Error Rate

Table 26. Tests of Within Subject Effects for Overall Error Rate

Source	Mean Square	F	p	Partial Eta Squared
Input Device	.076	149.616	.000	.932
Selection Method	.128	237.877	.000	.956
Input Device * Selection Method	.043	79.271	.000	.878

Table 27. Estimated Marginal means of Input Device for Overall Error Rate

Input Device	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Normal Mouse	.034	.003	.027	.041
Head Mouse	.125	.004	.116	.133
Joystick Mouse	.091	.004	.081	.101

Table 28. Estimated Marginal means of Selection Method for Overall Error Rate

Input Device	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Circling Interface	.081	.004	.071	.090
Manual Selection	.025	.002	.021	.028
Dwelling Interface	.144	.005	.132	.156

As shown in Figure 35, Table 26 and Table 27, there was a main effect for input device ($F(2, 30) = 149.616$, $p < .001$, $eta^2 = .932$). Across all pointing operations, a normal mouse (3.2%), on average, had the lowest error rate, followed by a joystick mouse (9.1%) and a head-mounted mouse emulator (12.5%). As shown in Table 26 and Table 28, a main effect for selection method ($F(2, 30) = 237.877$, $p < .001$, $eta^2 = .956$) was also detected. Across all pointing operations, the dwelling interface (14.4%), on average, had the highest error rate, followed by the circling interface (8.1%) and manual selection (2.5%).

As shown in Figure 35 and Table 26, there was an interaction between selection method and input device ($F(4, 60) = 79.271$, $p < .001$, $eta^2 = .878$). The error rate pattern of the dwelling interface across input devices was not same as those of the other two methods. While the error rate of the joystick mouse for the dwelling interface was higher than that of the head-mounted mouse emulator, for the other methods the error rate of the joystick mouse were lower than that of the head-mounted mouse emulator. While the difference of error rate across input devices for manual selection was relatively small, those of the other two methods were significantly larger. Both the circling interface and dwelling interface with the head-mounted mouse emulator had a significantly higher error rate.

4.3.4.3 Learning Effect

To test if there was a learning on performance time and accuracy with the circling interface, time to acquire the target and error rate during the target acquisition time were compared over three pointing task sessions (single-click, double-click, and drag-and-drop).

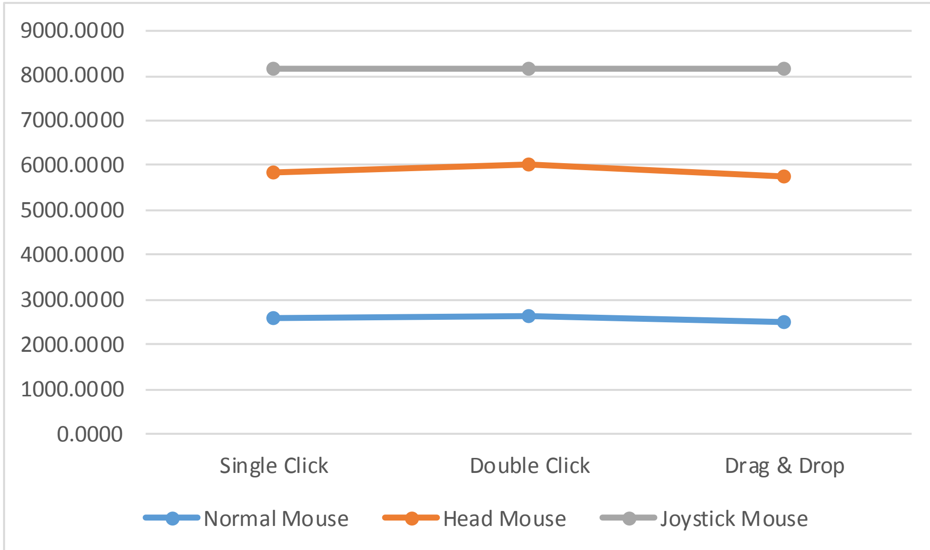

Figure 36. Learning Effect on Performance Speed

As shown in Figure 36, no statistically significant learning effect on performance time across all three input devices was detected: with a normal mouse ($p = .439$), with a head-mounted mouse emulator ($p = .294$) and with a joystick mouse ($p = .998$), respectively. Between the normal mouse and the head-mounted mouse emulator, the same pattern was detected: a slight reduction in performance time occurred between double-click and drag-and-drop operations.

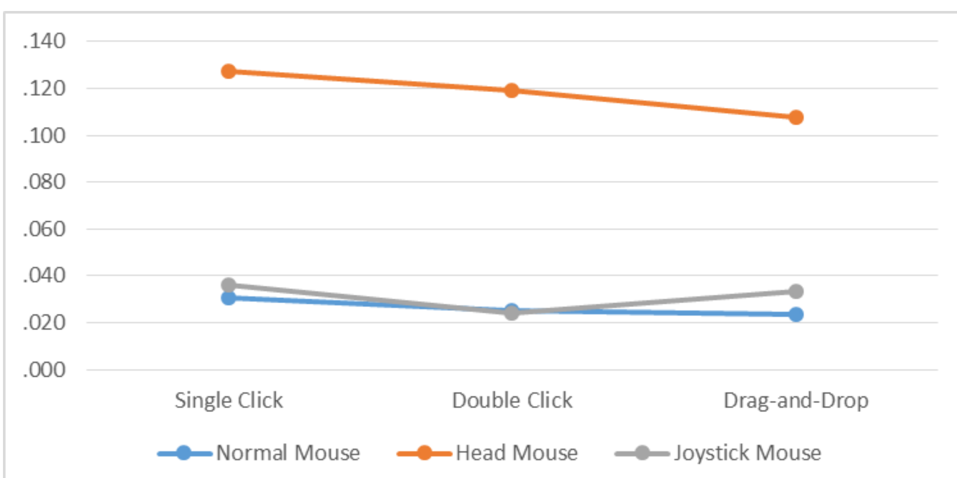

Figure 37. Learning Effect on Performance Accuracy

As shown in Figure 37, although a slight reduction in error rate was observed with both the normal mouse and the head-mounted mouse emulator over the sessions, no statistically significant learning effect on performance accuracy was detected: with the normal mouse (p = .570), with the head-mounted mouse emulator (p = .508) and with the joystick mouse (p = .235), respectively. Between the normal mouse and the head-mounted mouse emulator, the same pattern was detected: a slight yet constant reduction in error rate over the sessions.

4.3.4.4 Perceived Workload

The Shapiro-Wilk normality test was used to examine the perceived workload data. The assumption of normality was not satisfied, so the collected data was analyzed using Friedman's method. The overall statistical results are shown in Table 29.

Table 29. Statistical Significance for each comparison – Overall Perceived Workload

	Normal Mouse			Head Mouse			Joystick Mouse		
	Circle vs. Manual	Circle vs. Dwell	Manual vs. Dwell	Circle vs. Manual	Circle vs. Dwell	Manual vs. Dwell	Circle vs. Manual	Circle vs. Dwell	Manual vs. Dwell
Total	**.002**	**.028**	**.003**	**.002**	.638	**.010**	**.002**	.638	**.002**
Mental	**.003**	.185	**.003**	**.002**	.799	**.002**	**.003**	.686	**.003**
Physical	**.002**	.110	**.002**	**.002**	.656	**.002**	**.002**	.061	**.002**
Temporal	**.003**	.062	**.005**	**.002**	.656	**.003**	**.002**	.445	**.002**
Performance	**.002**	**.017**	**.012**	**.012**	.593	**.312**	**.003**	.480	**.003**
Effort	**.002**	**.019**	**.013**	**.008**	.534	**.012**	**.002**	.721	**.003**
Frustration	**.002**	*.052*	**.012**	**.003**	.533	**.003**	**.003**	.137	**.002**

(Statistically significant differences in **bold**. Marginally significant differences in *italic*.)

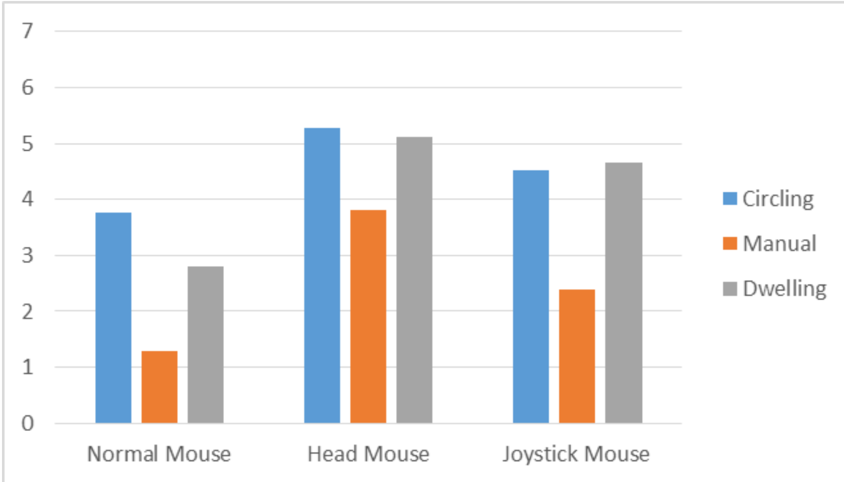

Figure 38. Task Load Index - Total Workload

As shown in Figure 38, the circling interface imposed a significantly higher workload than manual selection for all three devices and a significantly higher workload than the dwelling interface for the normal mouse. The dwelling interface imposed a significantly higher workload than manual selection for all three devices.

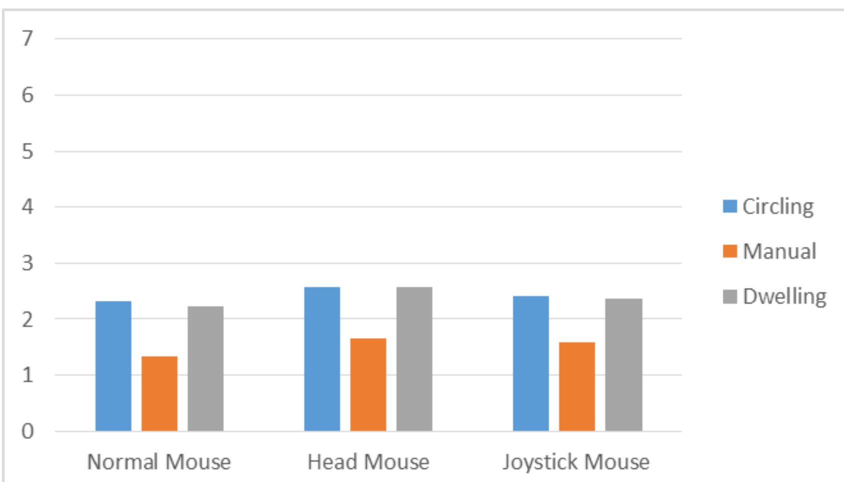

Figure 39. Task Load Index - Mental Demand

As shown in Figure 39, manual selection required significantly less mental effort than the circling interface and dwelling interface for all three devices.

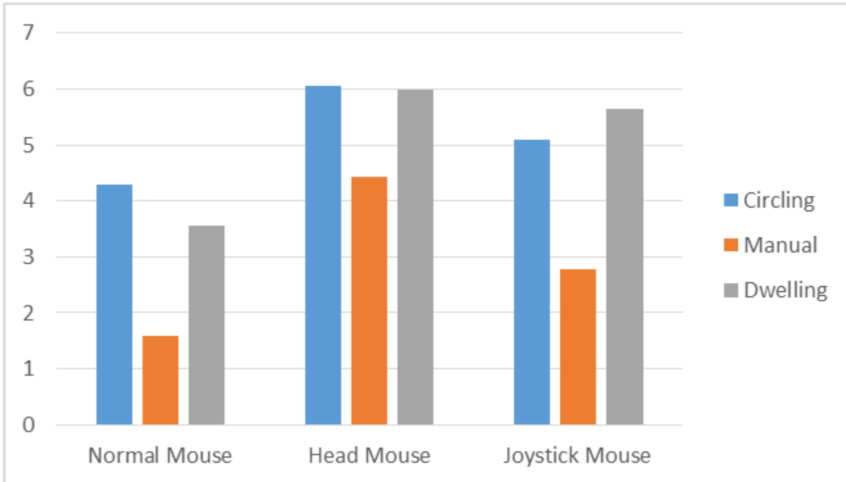

Figure 40. Task Load Index - Physical Demand

As shown in Figure 40, manual selection required significantly less physical effort than the circling interface and dwelling interface for all three devices.

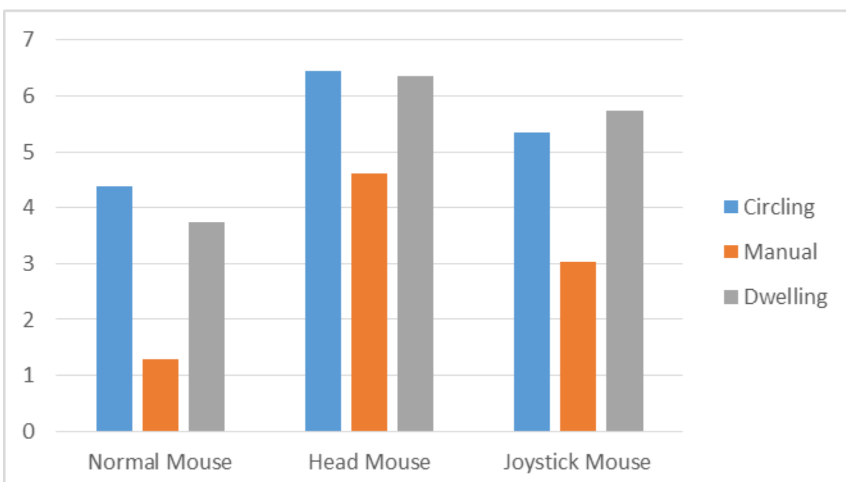

Figure 41. Task Load Index - Temporal Demand

As shown in Figure 41, manual selection was significantly less temporally demanding than the circling interface and dwelling interface for all three devices.

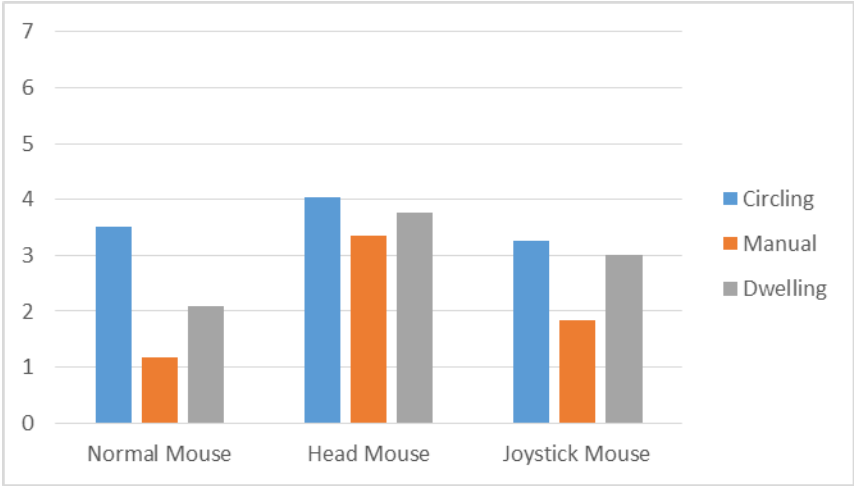

Figure 42. Task Load Index – Performance

As shown in Figure 42, the circling interface imposed significantly more performance burden than manual selection for the normal mouse and joystick mouse. Dwell selection imposed significantly more performance burden than manual selection when using the normal mouse and joystick mouse, and required significantly less performance burden than the circling interface for the normal mouse.

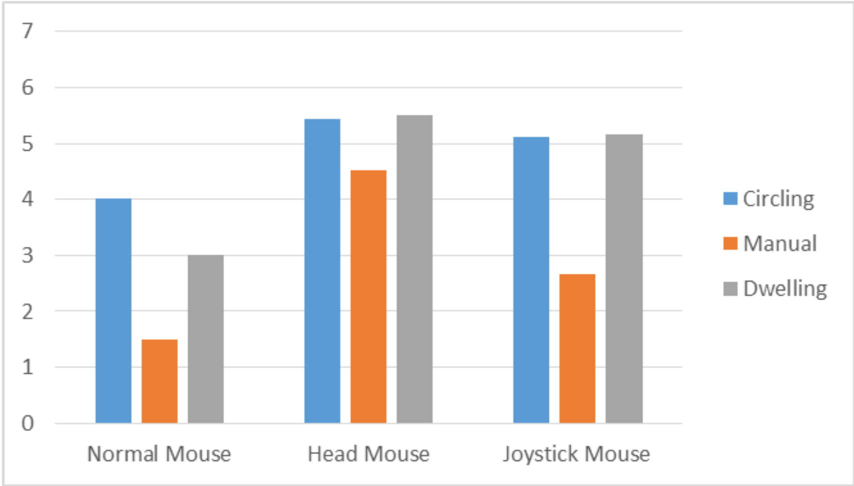

Figure 43. Task Load Index – Effort

As shown in Figure 43, manual selection required significantly less effort than the circling interface and dwell selection for all three devices. The circling interface required significantly more effort than dwell selection for the normal mouse.

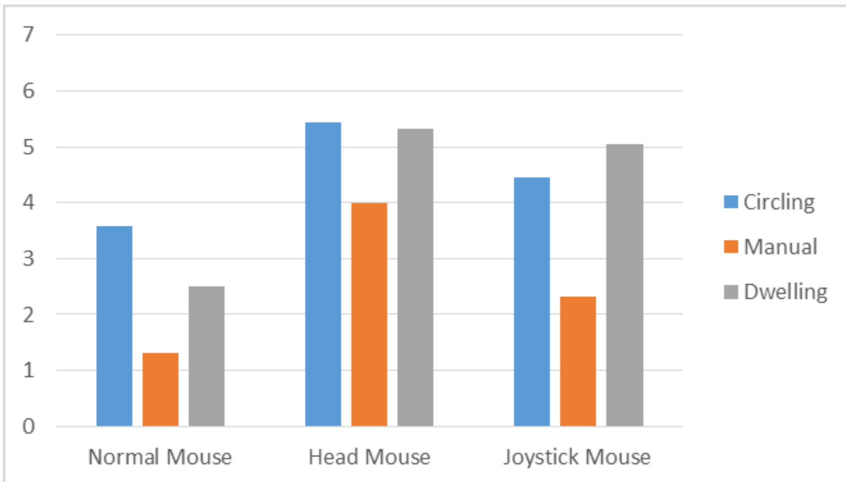

Figure 44. Task Load Index – Frustration

As shown in Figure 44, manual selection was significantly less stressful than the circling interface and dwell selection for all three devices. The circling interface was marginally more stressful than dwell selection for the normal mouse.

4.4 DISCUSSION

4.4.1 Task Completion Time

In terms of task completion time, the traditional mouse and mouse button was the fastest combination for able-bodied subjects across all tasks. For the single-click operation, the circling interface showed the least efficient performance among all three methods for all three input devices. One possibility was that the distance traveled by the mouse cursor in the circling interface was longer than those of the other methods because the mouse cursor had to go to, and then all the way around, each target. Inspecting the recorded mouse cursor trajectory data revealed that the distance traveled by the mouse cursor in the circling interface was, on average, 29.1% longer across all three input devices.

For both double-click and drag-and-drop operations, the circling interface produced equivalent or faster performance to the dwelling interface across all input devices. This was mainly because the dwelling interface required additional time to specify the operation compared to the circling interface (See Figure 45), and required a greater travel distance to correct errors. In addition to the dwelling time, the dwelling interface requires more precise pointing ability than the circling interface, which uses each edge of the screen to specify the command (e.g., left edge for left-click; right edge for right-click; top edge for double-click; bottom edge for drag).

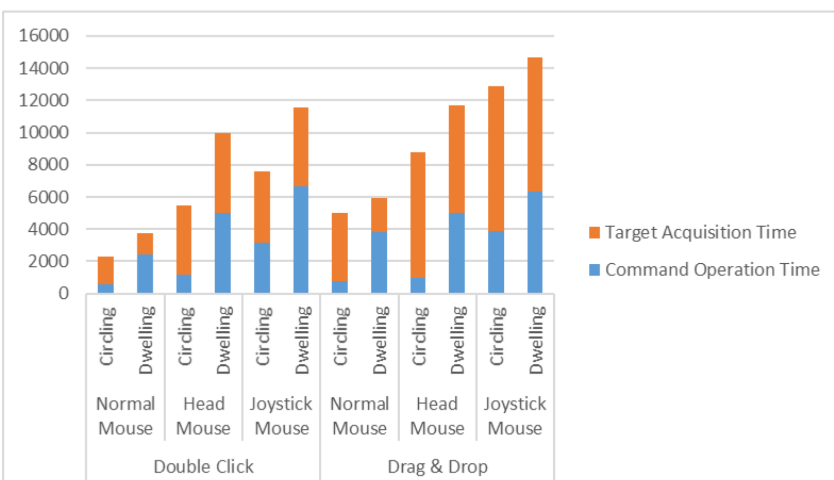

Figure 45. Command Operation Time vs. Target Acquisition Time (for error-free trials)

The traditional mouse was fastest across all tasks, followed by the head-mounted mouse emulator and the joystick mouse. This is compatible with the existing research findings (I.S.

MacKenzie, 1995; I.S. MacKenzie, Sellen, & Buxton, 1991; Radwin et al., 1990). Across all operations, the circling interface had a tendency to produce faster performance with a head pointer than with a joystick mouse.

No significant learning effect was detected for the circling interface. However, across all input devices, the same pattern was detected: a slight reduction in performance speed occurred between double-click and drag-and-drop operations.

As shown in Figure 46, both the manual selection and the dwelling interface conformed to Fitts' Law. The circling interface conformed to Fitts's law, except under sparse target density. The circling interface showed better performance with sparsely packed small targets than with sparsely packed large targets (p = .017), because smaller movements were required to circle small targets.

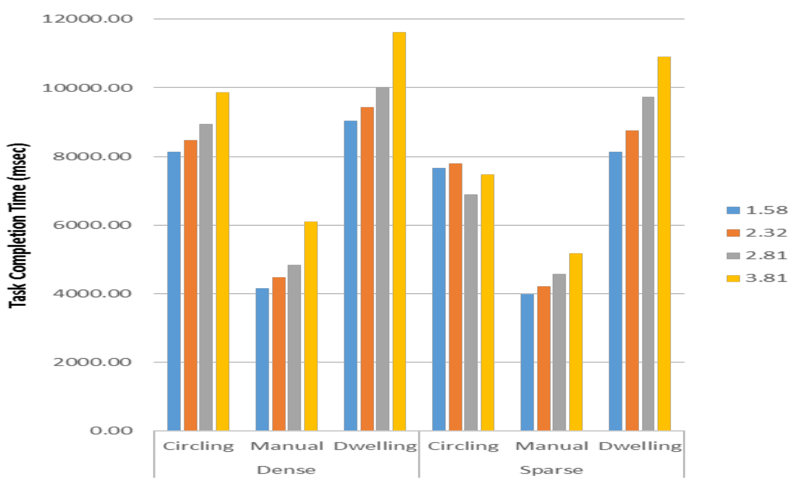

Figure 46. Task Completion Time by ID and Density

4.4.2 Error Rates

The traditional mouse and mouse button was the combination with the least errors across all pointing operations. Dwelling had fewer errors than circling for single-click operation but more errors for double-click and drag-and-drop operations across all input devices. No significant learning effect was detected for the circling interface. For both the normal mouse and the head-mounted mouse emulator, a slight yet consistent reduction in error rate occurred over the three sessions. For the joystick mouse, error rate decreased between single-click and double-click operations, but between double-click and drag-and-drop operations error rate increased again.

The possible errors that could occur when using the circling interface can be classified into the following five types:

- Circling on an area with no target
- Circling the wrong target
- Circling with the wrong command mode active
- Unintentional circling caused by jerky movements
- Unintentional circling caused by an abnormally tiny circle

To identify the portion of each type, the recorded mouse cursor event data were inspected across all pointing operations.

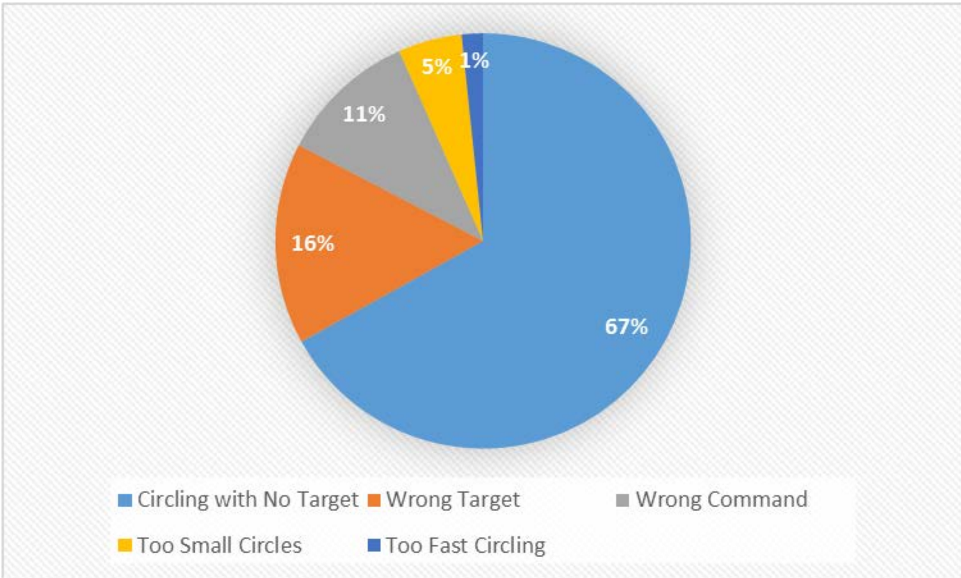

Figure 47. Portion of Each Error Type for a Normal Mouse

As shown in Figure 47, for the normal mouse, out of total 121 errors, 67% were made by circling on an area with no target (mostly during the ballistic movement phase), 16% by circling the wrong target; 11% by circling with the wrong command active, 5% by too small circles, and the remaining 1% by too fast movements.

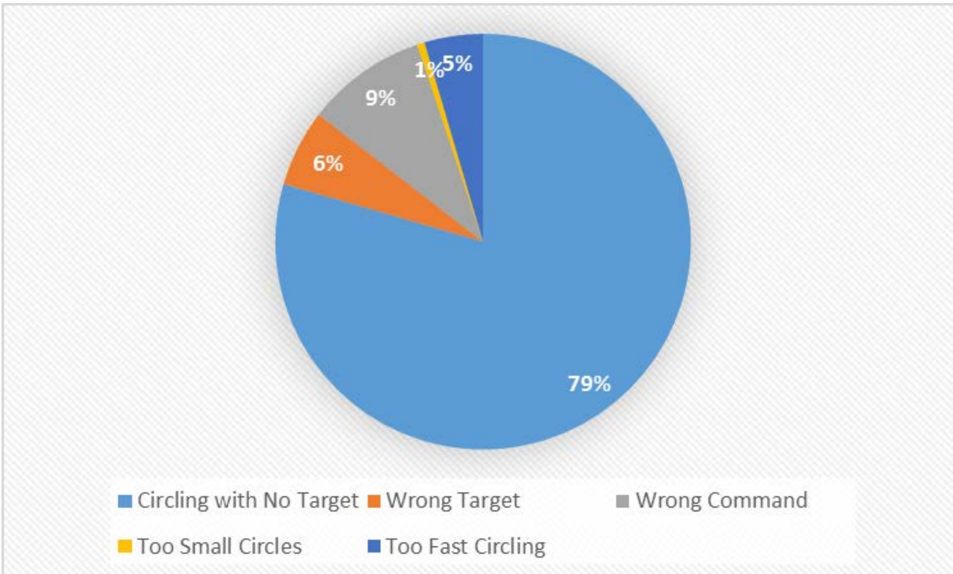

Figure 48. Portion of Each Error Type for a Head-mounted mouse emulator

As shown in Figure 48, for the head-mounted mouse emulator, out of total 351 errors, 79% were made by circling on an area with no target (mostly during the ballistic movement phase), 6% by circling the wrong target, 10% by circling in the wrong command, 5% by too fast circling, and the remaining 1% by too small circles.

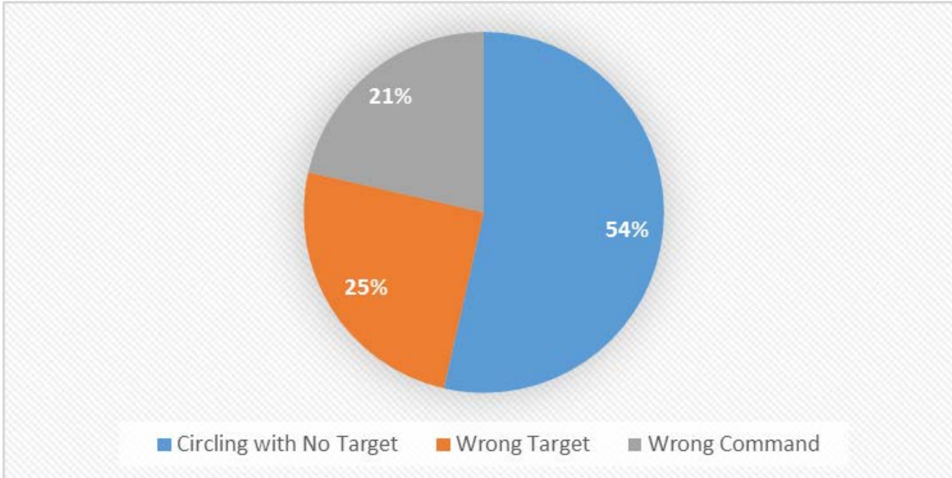

Figure 49. Portion of Each Error Type for a Joystick Mouse

As shown in Figure 49, for the joystick mouse, out of 82 errors, 54% were made by circling on an area with no target (mostly during the ballistic movement phase), 25% by circling the wrong target, and 21% by circling with the wrong command active.

The circling interface is by nature a target-gnostic approach, while the dwelling interface is target-agnostic, which may allow the circling interface to eliminate errors caused by circling on an area with no target. The circling interface works by continuously updating real-time mouse movement information along with the properties of all UI elements currently displayed on the screen, which can be used not only to indicate what kind of UI element a particular control represents (such as a button, icon, or menu), but also to determine what types of controls are available in the current context and how to interact with them. If a circling motion is recognized and the algorithm finds there is no clickable target in the circled area, that circling motion can be ignored. In addition, unintentional circling caused by jerky movements or an abnormally tiny circle can be also filtered out. Applying these error correction techniques to the collected data significantly improves the accuracy of the circling interface (See Figure 50). However, this approach might not work in practice, since people sometimes want to click in an open space (for example, when dragging and dropping). One possible way to address this issue is to make the error correction technique specific to the active application and the current context. Another option is to only eliminate circles that are too small or that are made too quickly (both of which indicate that they are unintentional).

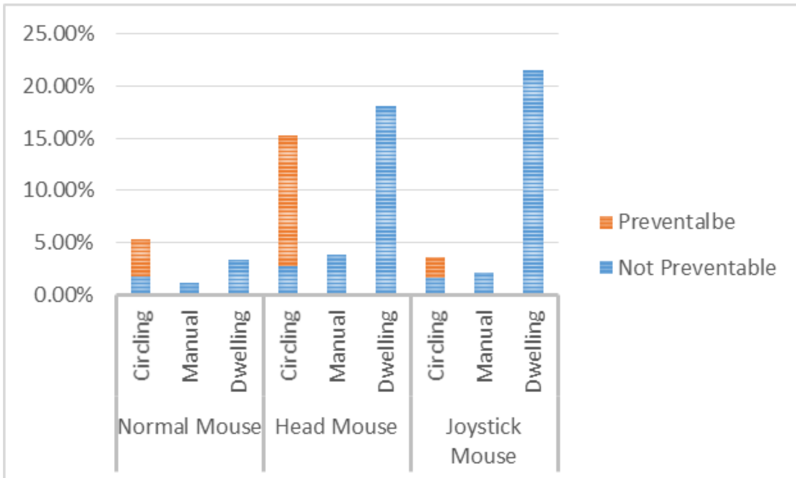

Figure 50. Overall Error Rate

4.4.3 Perceived Workload

As shown in Figure 51, the traditional mouse and manual selection had the lowest total workload. With a normal mouse the dwelling interface imposed significantly lower workload than the circling interface, but participants did not find any significant difference in difficulty between the circling and dwelling interface for both the head-mounted mouse emulator and the joystick mouse. This may be due to the fact that none of the subjects had prior experience with assistive input devices.

Regarding the subscales, participants found few differences between the circling and dwelling interface across all input devices. In both perceived performance and effort level, they felt that the circling interface imposed more burden than the dwelling interface with a normal mouse. This may be because the dwelling interface relies on a more familiar pointing-based approach than the circling interface, which uses a gesture-based approach. However, this lack of familiarity may be overcome by training and practice. As shown in Figure 52, Difference in subjective workload between circling and dwelling decreased across sessions.

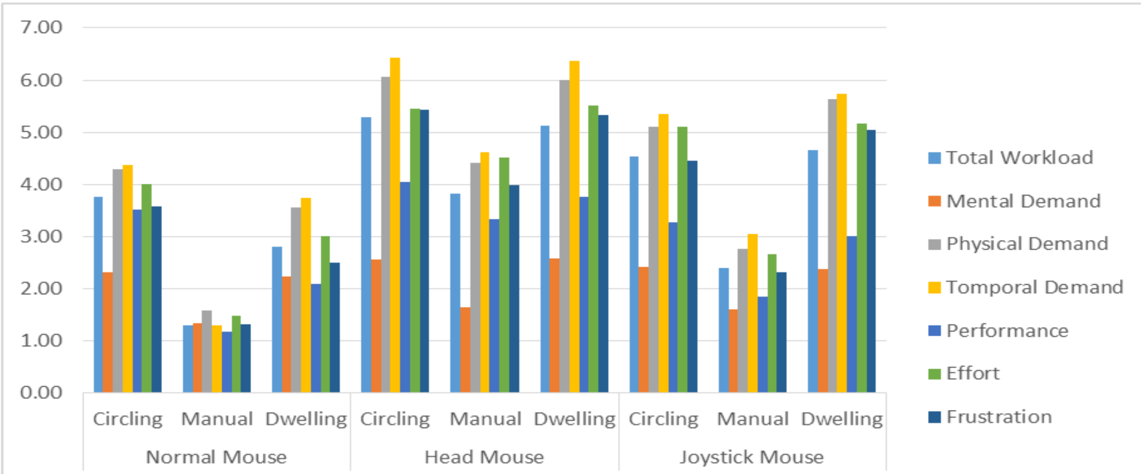

Figure 51. TLX Subscales between Methods for Each Device

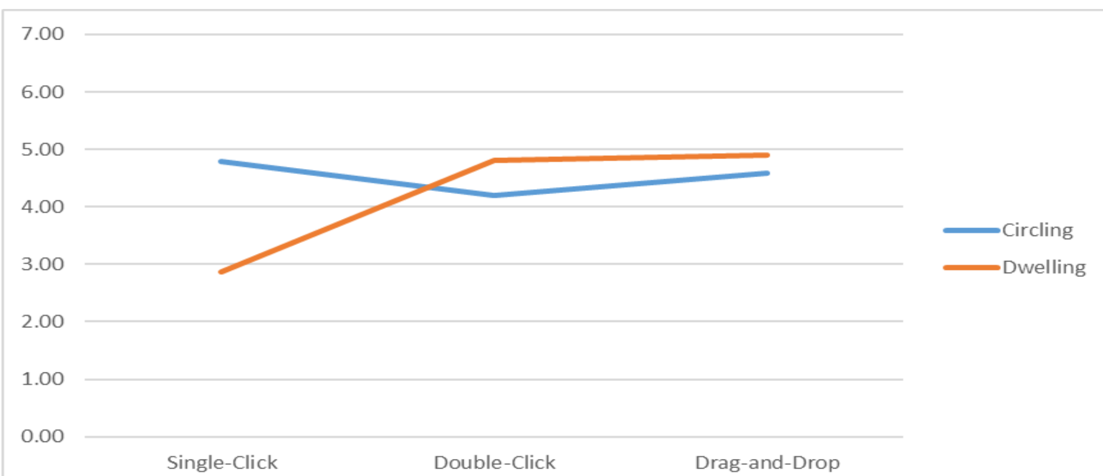

Figure 52. Difference between the Circling Interface and Dwell Selection over Pointing Operations

5.0 EMPIRICAL EVALUATION II: SUBJECTS WITH SPINAL CORD INJURY

5.1 OVERVIEW

An experiment was conducted with individuals with spinal cord injury (SCI) to evaluate the performance of the circling interface on basic pointing tasks. SCI, which refers to any injury to the spinal cord, is typically associated with major trauma from motor vehicle accidents, falls, sports injuries, and violence (Thomas, 1989). The consequences of SCI depend upon the level and completeness of the injury (V. Lin et al., 2010). The level of an injury is based upon where the spinal cord and nerve roots are affected. The completeness of an injury is often assessed using a scale defined by the American Spinal Injury Association (Marino et al., 2003). The level of SCI determines which parts of body are affected and the completeness of the injury determines the extent to which body functions will be affected. In general, injuries higher on the spinal cord result in greater impairment. Individuals with a high level and more severe SCI often use a head-mounted mouse emulator along with click emulation software, such as a dwelling interface, for their primary pointing method to access their computers.

5.2 METHODS

5.2.1 Hypotheses

The following hypotheses were tested:

- Performance time for pointing operations would be different between the circling interface and dwell-clicking software when using a head-mounted mouse emulator.
- Performance accuracy for pointing operations would be different between the circling interface and dwell-clicking software when using a head-mounted mouse emulator.
- Users' satisfaction level would be different between the circling interface and dwell-clicking software when using a head-mounted mouse emulator.

5.2.2 Subjects

Five quadriplegic SCI (C4-C5) individuals (5 Males; age range 24-46) were recruited. The inclusion criteria were:

- Participant understood the purpose and the nature of all experimental procedures and tasks;
- Participant was at least 21 years of age;
- Participant was currently using a head-mounted mouse emulator;
- Participant had sufficient visual acuity to perform the given pointing tasks.

Each participant's eligibility was determined based on an interview conducted before a written informed consent form was signed. All participants were recruited, and written consents were obtained, in accordance with the Institutional Review Board at the University of Pittsburgh.

All participants had lived with SCI over five years and were active users of head-mounted mouse emulators and dwell-clicking software. While three of them relied exclusively on a head-mounted mouse emulator with dwell-clicking software, the other two switch between dwell-clicking software and a touch switch depending on their health condition. All subjects had used computers for at least 5 years, and their average computer usage time was more than 20 hours per week.

5.2.3 Experimental Design

The experiment used a repeated-measures design. Experimental conditions were defined by the following variables:

- 3 Pointing operations
    - Single-click
    - Double-click
    - Drag-and-drop
- 1 Input device
    - Head-mounted Mouse Emulator
- 2 Selection methods
    - Circling Interface
    - Dwell-clicking Interface
- 2 Target sizes

    - Large (48 x 48 pixels)
    - Small (16 x 16 pixels)
- 2 Target distances
    - Short (176 pixels);
    - Long, (336 pixels)
- 2 Densities between Targets
    - Dense (2 pixels)
    - Sparse (32 pixels)

Selection method was a within-subjects factor. Target distance (A) and target size (W) were not treated as separate factors, since Fitts' law shows that these factors are not independent (I. S. MacKenzie, 1992). Instead, the Index of Difficulty was used as a single factor:

$$(ID) = \log_2(A/W + 1)$$

5.2.4 Equipment

All participants used the same head-mounted mouse emulator[7]. Participants either wore their own reflective dot or a headband, provided by the investigator, to which a reflective dot was attached. The head-mounted mouse emulator was connected to an Intel-based Macintosh laptop computer[8] running Windows 7[9]. To provide participants with better access to on-screen objects, a 17-inch LCD monitor[10] with a 1200×800 pixel screen resolution connected with the laptop computer through a mini display port to VGA cable adapter was used. Throughout the

[7] HeadMouse Extreme (Origin Instruments Corporation, Grand Prairie, TX)
[8] Macbook Pro, (Apple Inc., Cupertino, CA)
[9] Windows 7 (Microsoft, Redmond, WA)
[10] Acer 173 (Acer America Corporation, San Jose, CA)

experiment across all participants, pointing parameters for the head-mounted mouse emulator were set to as shown in Table 30. Participants had the option to adjust these parameters but no participants chose to do so.

Table 30. Pointing Parameters for Subjects with SCI

Parameter	Value
Pointer Speed	18
Pointer Precision	off
Snap-To	off
Dwelling Time	800 msec
Permissible Movement	±5 pixels
Circling Trail Length	400 pixels
Maximal Circling Speed	< 100 pixels/tick
Minimal Circumference	> 16 pixels

5.2.5 Data Collection

At the beginning of the study, participants completed an initial questionnaire (see Appendix A.1.1) regarding basic demographic information, computer experience, and input devices that they currently use. During each trial, all user interface events were recorded, time-stamped and stored in an XML file format for subsequent analysis (Simeon Keates et al., 2002; I. Scott MacKenzie et al., 2001).

The following performance measures were calculated:

- Time to acquire the target - For the dwell clicking and circling interfaces, the time from when the operation (single-click, double-click, drag) is specified to the time in which the target is first entered or circled.
- Time to specify the operation - For dwell selection, the sum of the time between the start of the trial to the time when the operation is selected from the tool palette

and the time from target acquisition to when the operation is specified on the target. For the circling interfaces, the time between the start of the trial to the time when the operation is specified by reaching one of screen edges. For the dwell-clinging and circling interface, single click was the default and did not need to be specified unless the wrong command was active.

- Time to acquire drag goal location - The time from drag initiation to entering or circling the location where the target was to be dragged.
- Clicking errors - The number of times a click occurred outside the target.
- Dragging errors - The number of times the user terminated a drag outside of the drag goal location.
- Task completion time - The total time from the initiation of mouse cursor movement to task completion: the sum of time to acquire the target, time to specify the operation and time to acquire the drag goal location. For dwell selection, the task ended as soon as the operation is specified on the target. For the circling interface, the task ended as soon as the target was acquired.
- Error rate – The sum of clicking errors and dragging errors.

Participants' satisfaction level with each selection method was measured using a questionnaire consisting of ten statements (see Appendix A.1.3). The first six statements measured participants' perceived workload and were adapted from the six subscales (mental demand, physical demand, temporal demand, performance, effort, and frustration) of the NASA-TLX. The last four statements measured improvement level of computer access, confidence level in using the technology, comfort level in using home, work and community, possibility of distributing the technology to others and were adapted from the Matching Persons with

Technology (MPT) instrument (Cushman & Scherer, 1996; Scherer & Cushman, 2000, 2001; Scherer & Galvin, 1996). The MPT is a validated assessment tool that measures influences on the successful use of a variety of technologies, including assistive and educational technology. Each item is measured on a scale of 0 to 7 where zero is the lowest level of agreement.

5.2.6 Experimental Procedures

Participants completed three sessions, one for each pointing operation. The order of pointing operations was the same for all participants (single-click, then double-click, then drag-and-drop). At the start of each session, participants received instructions on how to perform pointing operations using the circling interface. The participant was given the opportunity to practice using the test software to become familiar with each interaction method for at least 15 minutes.

After participants reported that they felt comfortable with the circling interface, they completed a pretest session to determine their eligibility for the subsequent experimental test procedures. The pretest session consisted of 8 trials with the circling interface, each of which represented one combination of experimental conditions (4 IDs x 2 target densities). Each condition was presented in random order and at random locations. Each participant took the pretest twice. If the average error rate was less than 20% and the participant did not make the an error twice in the same condition, then the participant was qualified for the experimental procedures. All participants passed the pretest in their first attempt.

Each session consisted of 16 blocks of trials (2 selection methods x 4 IDs x 2 target densities) and each block consisted of 10 trials of a single condition. The first two trials within each block were treated as practice. The order of conditions was randomized and counter-balanced across subjects.

A single trial consisted of the acquisition of one target surrounded by eight “distractor” targets as shown in Figure 6. At the start of each trial, the mouse cursor was positioned in the center of the display and a “3–2–1” countdown flashed in the center of the screen. A trial ended when the target had been acquired or 30 seconds had expired, at which point the mouse cursor would return to the center of the display in preparation for the next trial. Once all ten targets had been acquired, a new block of ten trials was presented. Participants were allowed to rest between blocks of trials as desired. At the end of each session, participants were asked to rate their satisfaction with each selection method using the questionnaire.

5.2.7 Statistical Analysis

The data were analyzed using the Wilcoxon signed-rank test as a nonparametric paired t-test due to the small sample size. Each performance measure was tested for an effect of selection method. Task completion time and error rate were also examined to determine if there was a learning effect over the three test sessions using Friedman’s method as a nonparametric single factor repeated measures ANOVA. Participants’ satisfaction level for each selection method were compared using the Wilcoxon signed-rank test. The statistical significance level for all statistical analyses was set to .05.

5.3 RESULTS

5.3.1 Task Completion Time

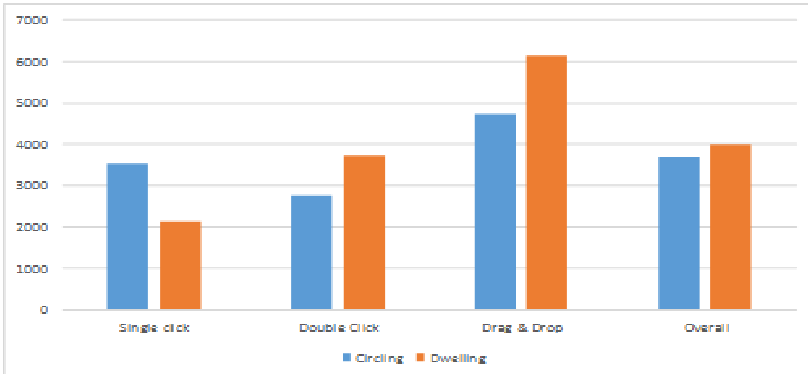

Figure 53. Task Completion time for Subjects with SCI

As shown in Figure 53, for the single-click operation, a Wilcoxon signed-rank test showed that there was a significant difference in performance time between the circling interface and dwell selection (Z = -2.032, p = .043). The circling interface (median = 3500 msec) was significantly slower than the dwelling interface (median = 2161 msec).

For the double-click operation, there was a significant difference in performance time between the circling interface and dwell selection (Z = -2.023, p = .043). The circling interface (median = 2608 msec) was significantly faster than the dwelling interface (median = 3628 msec).

For the drag-and-drop operation, there was a significant difference in performance time between the circling interface and dwell selection (Z = -2.023, p = .043). The circling interface (median = 6485 msec) was significantly faster than the dwelling interface (median = 6027 msec).

Overall, across all operations, there was not a statistically significant difference in performance time between the circling interface and dwell selection (Z = -1.753, p = .080).

However, a medium effect size of .53 implies that the circling interface (median = 3531 msec) was practically faster than the dwelling interface (median = 3881 msec).

5.3.2 Error Rate

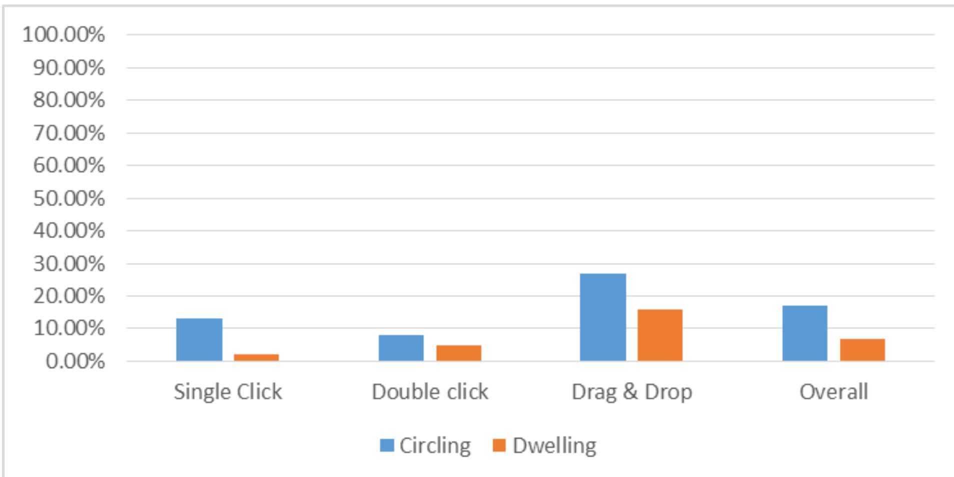

Figure 54. Error Rate for Subjects with SCI

As shown in Figure 54, for the single-click operation, a Wilcoxon signed-rank test showed that there was a significant difference in error rate between the circling interface and dwell selection (Z = -2.023, p = .042). The circling interface (median = 13%) was significantly less accurate than the dwell selection (median = 2%).

For the double-click operation, there was not a statistically significant difference in error rate between the circling interface and dwell selection (Z = --1.355, p = .176). The median value of each selection method was: circling interface = 8% and dwell selection = 5%.

For the drag-and-drop operation, there was a statistically significant difference in error rate between the circling interface and dwell selection (Z = -2.032, p = .042). The circling interface (median = 27%) was significantly less accurate than dwell selection (median = 16%).

Overall, across all pointing operations, there was a statistically significant difference in error rate between the circling interface and dwell selection (Z = -2.023, p = .043). The circling interface (median = 17%) was significantly less accurate than the dwelling interface (median = 7%).

5.3.3 Learning Effect

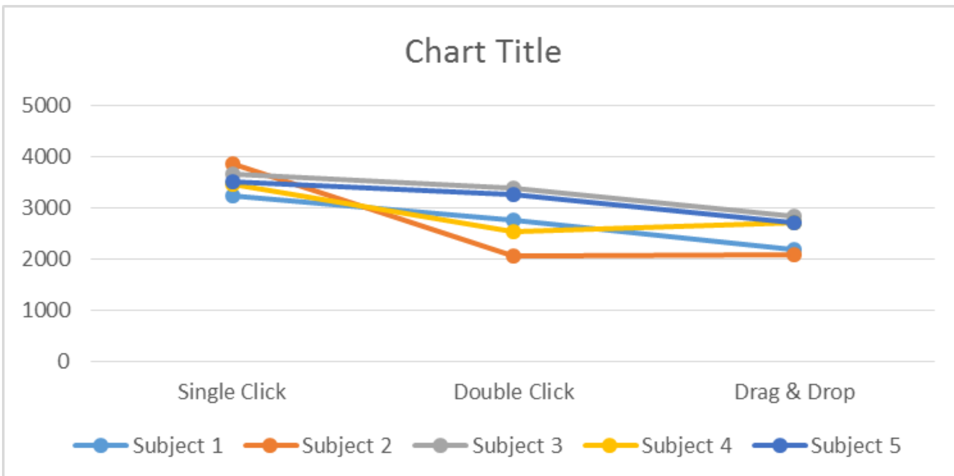

Figure 55. Learning Effect on Task Completion Time for Subjects with SCI

As shown in Figure 55, a significant learning effect on the task completion time of circling interface was detected ($\chi^2(2) = 7.600$; p = .022). A post-hoc analysis with Wilcoxon's Signed-Rank Tests was performed with a Bonferroni correction applied. While, on average, the difference between single-click and double-click operations was statistically significant (Z = -2.023, p = .043), the difference between double-click and drag-and-drop operations was not (Z = -1.214, p = .225).

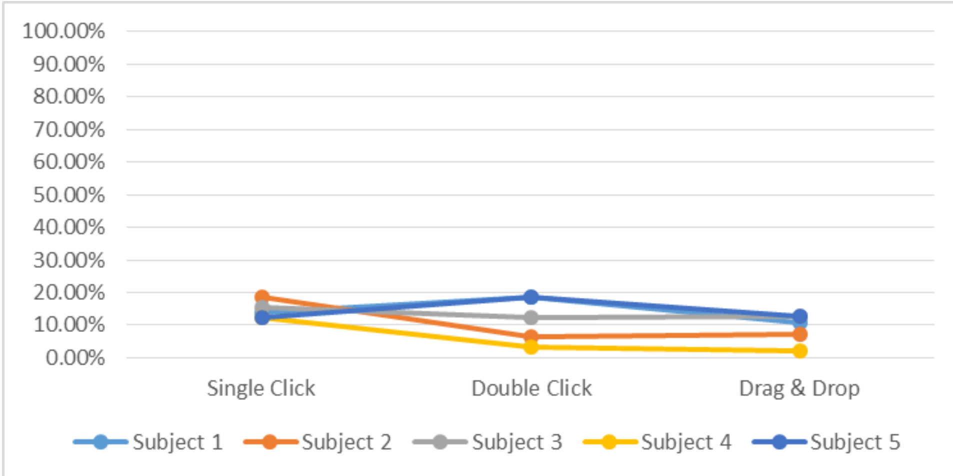

Figure 56. Learning Effect on Error Rate for Subjects with SCI

As shown in Figure 56, on average, no statistically significant learning effect for error rate with the circling interface was detected ($\chi^2(2) = 3.444$; $p = .179$). However, it was shown that the error rate decreased over the three sessions.

5.3.4 Satisfaction Level

The statistical results are shown in Table 31.

Table 31. Satisfaction Level for Subjects with SCI

	Statement	SC	DC	DD	Overall
A	The tasks or task elements required much mental and perceptual effort (e.g. thinking, deciding, calculating, remembering, looking, searching, etc.).	**.025**	.564	1.000	.063
B	The tasks or task elements required much physical effort (e.g. pushing, pulling, turning, controlling, activating, etc.)?	**.038**	.180	**.046**	.180
C	I felt time pressure due to the rate of pace at which the tasks or task elements occurred.	**.034**	.083	1.000	.180
D	It was difficult to learn how to use the technology.	.059	.083	1.000	.180
E	It was difficult to accomplish the goals of the task set by the experimenter.	.059	1.000	.157	.102
F	I felt insecure, discouraged, irritated, stressed and/or annoyed during the tasks or task elements.	**.034**	.059	.317	.066
G	This technology will benefit me and improve my computer access.	.157	1.000	.414	1.000
H	I am confident I know how to use this technology and its various features**.**	**.034**	.102	.102	**.041**
I	I will feel comfortable (and not self-conscious) using this technology at home and in public.	.059	.317	.083	.066
J	I will recommend this technology to others who may benefit.	**.025**	.564	1.000	.131

(SC: single-click; DC: double-click DD: drag-and-drop; statistically significant differences in **bold;** marginally significant differences in *italic*.)

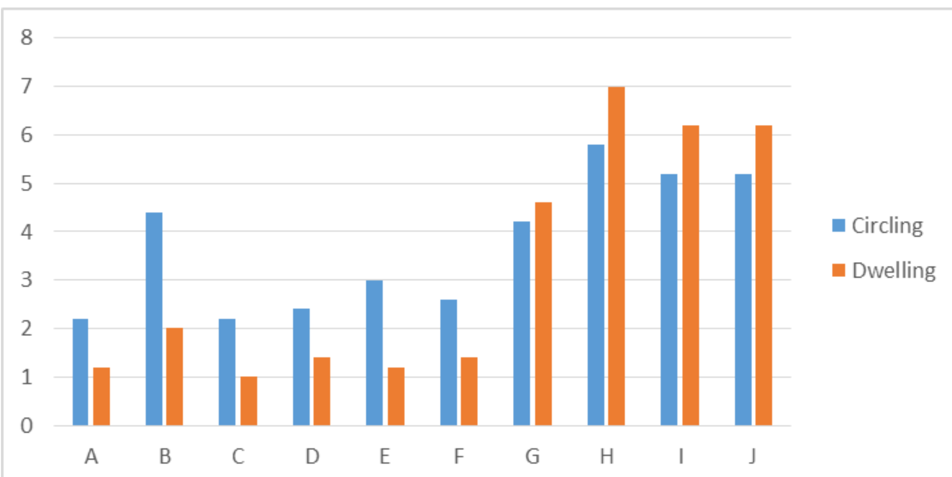

Figure 57. Single-Click Satisfaction Level for Subjects with SCI

As shown in Figure 57, for the single-click operation, the circling interface imposed significantly higher mental, physical, and temporal demands, and stress than dwell selection.

Participants had greater confidence in dwell selection and were more likely to recommend dwell selection to others than the circling interface.

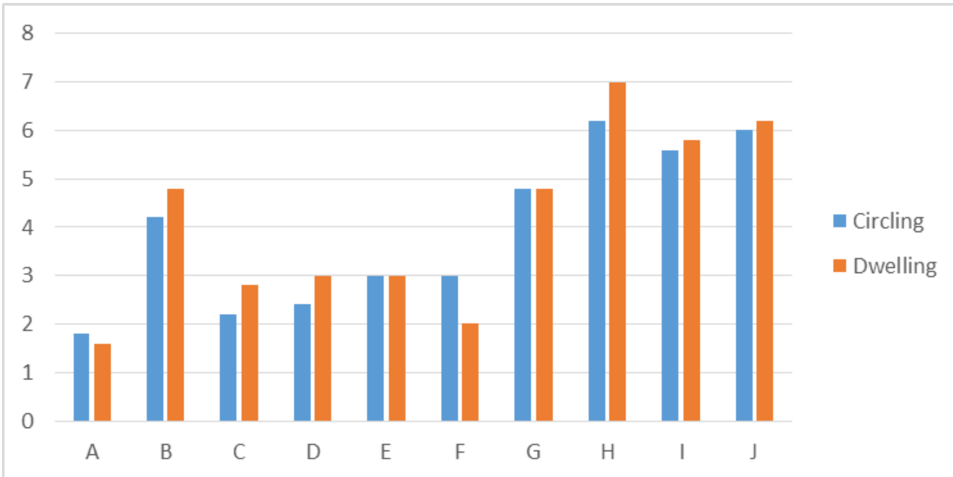

Figure 58. Double-Click Satisfaction Level for Subjects with SCI

As shown in Figure 58, for the double-click operation, no statistically significant difference between the circling interface and dwell selection was detected in any item.

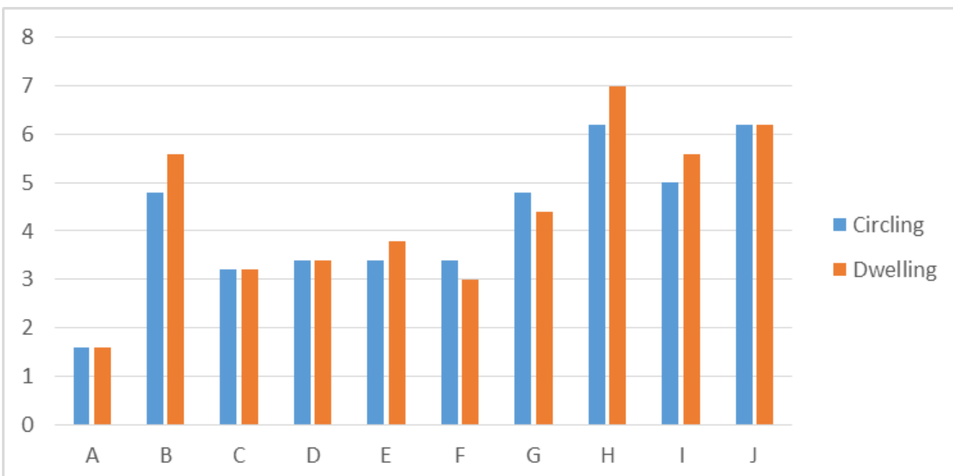

Figure 59. Drag-and-Drop Satisfaction Level for Subjects with SCI

As shown in Figure 59, for the drag-and-drop operation, the dwelling interface required a statistically significantly greater physical effort than the circling interface.

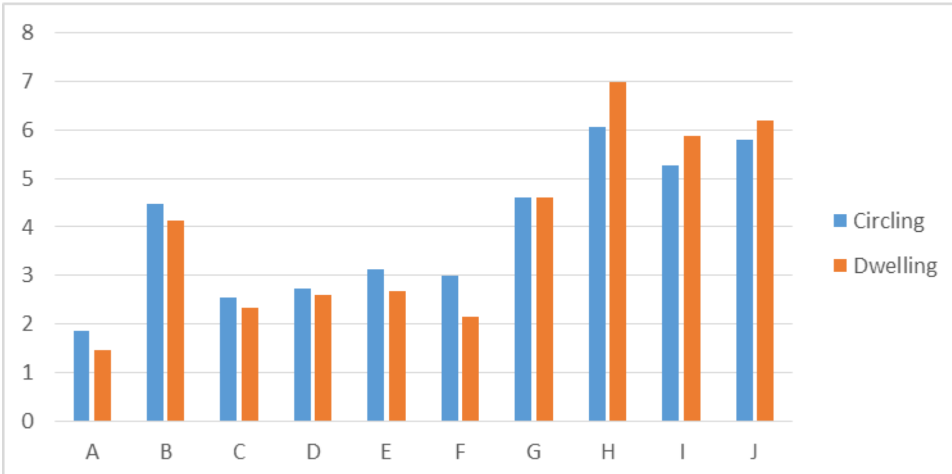

Figure 60. Overall Satisfaction Level for Subjects with SCI

As shown in Table 60, the only statistically significant difference detected was that participants had greater confidence in the dwelling interface than the circling interface (Item H).

5.4 DISCUSSION

Averaged across all tasks, the circling interface and dwell selection showed relatively equal task completion time for subjects with SCI when using a head-mounted mouse emulator. For the single-click operation, the circling interface was significantly slower than dwell selection. One possibility was that the distance traveled by the mouse cursor in the circling interface was longer than those of the other methods because the mouse cursor had to go to, and then all the way around, each target. Inspecting the recorded mouse cursor trajectory data revealed that the distance traveled by the mouse cursor in the circling interface was, on average, 25.4% longer. For both double-click and drag-and-drop operations, the circling interface produced faster performance than dwell selection. This was mainly because the dwelling interface required additional time to specify the operation compared to the circling interface (see Figure 61). In addition to the dwelling time, the dwelling interface requires more precise pointing ability than the circling interface, which uses the edges of the screen to specify the command (e.g., left edge for left-click; right edge for right-click; top edge for double-click; bottom edge for drag). Locating objects along an edge decreases target selection time (Accot & Zhai, 2002; Walker et al., 1993), because the screen edge emulates a target with an infinite effective width that the mouse cursor cannot overshoot (Atwood, 2006; Hale, 2007).

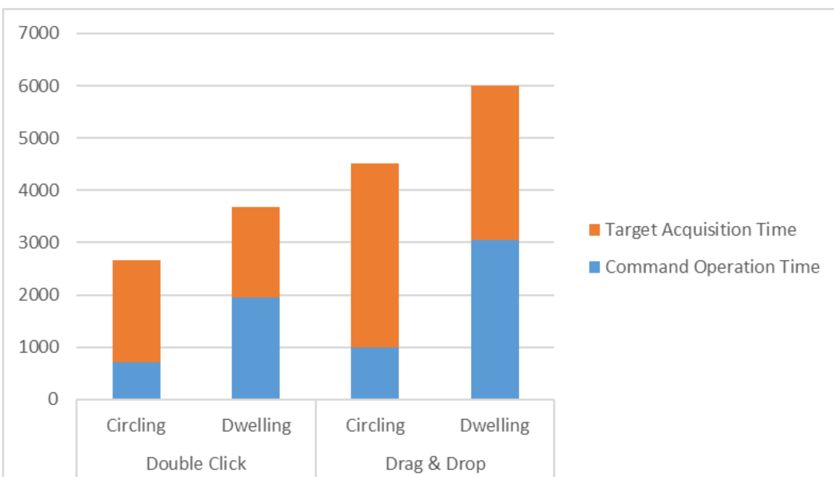

Figure 61. Command Operation Time vs. Target Acquisition Time (for error-free trials)

As shown in Figure 62, when selecting densely packed targets, both methods conform to Fitts' Law, but when selecting sparsely packed targets, the circling interface was barely affected by ID.

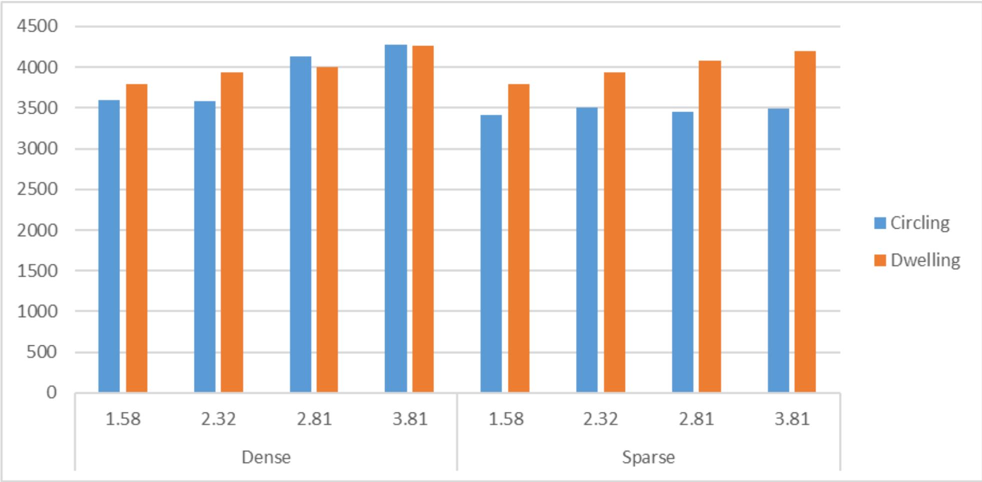

Figure 62. Task Completion Time by ID and Density

In terms of performance accuracy, dwell selection had fewer errors than the circling interface across all tasks. As a result of inspecting the recorded mouse cursor event data, out of 165 errors, 81% were made by circling on an area with no target (mostly during the ballistic movement phase), 13% by circling the wrong target, 3% by circling with the wrong command active, 2% by too fast circling and 1% by too small circles (See Figure 63).

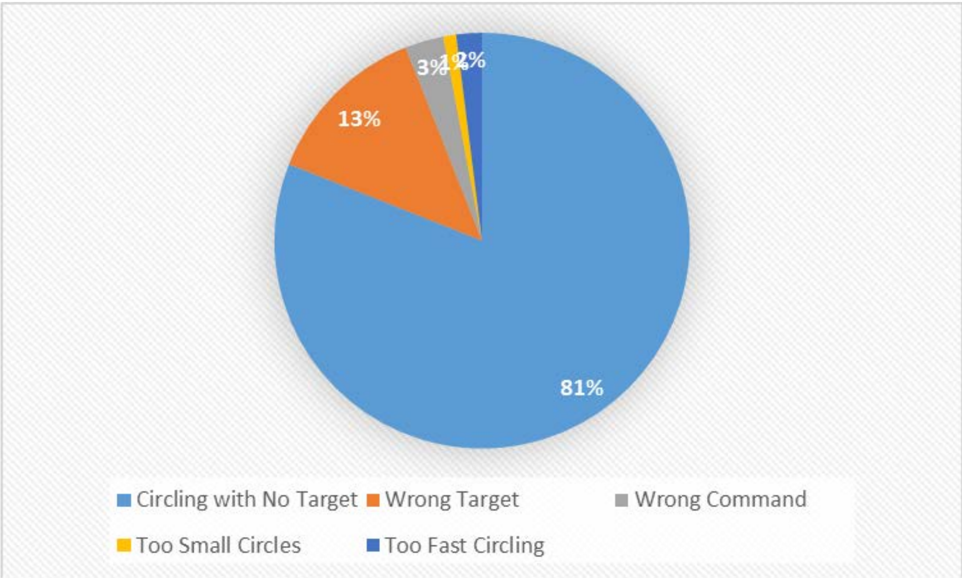

Figure 63. Portion of Each Error Type

Applying the same error correction techniques as in the previous chapter (Section 4.4.2) to the collected data significantly improves the accuracy of the circling interface (See Figure 64). However, this approach might not work in practice, since people sometimes want to click in an open space (for example, when dragging and dropping). One possible way to address this issue is to make the error correction technique specific to the active application and the current context. Another option is to only eliminate circles that are too small or that are made too quickly (both of which indicate that they are unintentional).

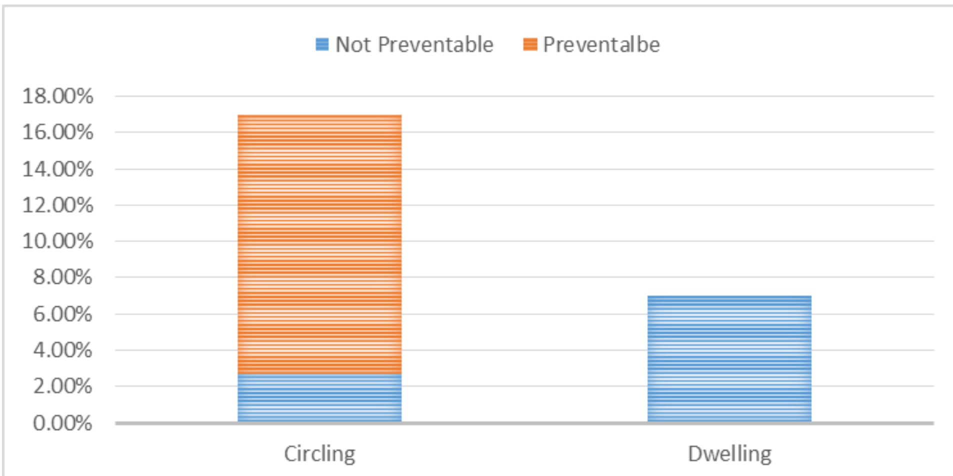

Figure 64. Overall Error Rate for Subjects with SCI

Improvement in both task completion time and error rate across all three sessions were found for all subjects, which suggests that the circling interface does not have a steep learning curve, and that more training and practice opportunities will effectively enhance user performance. Users' subjective ratings of the circling interface support this as well. For single-click operation, subjects felt that the circling interface imposed a greater workload and preferred dwell selection. However, the differences between the two methods decreased over the sessions.

6.0 EMPIRICAL EVALUATION III: SUBJECTS WITH CEREBRAL PALSY

6.1 OVERVIEW

An experiment was conducted with individuals with cerebral palsy (CP) to evaluate the performance of the circling interface on basic pointing tasks. CP is one of the most common physical disabilities in a group of disorders that results from brain injury that typically occurs before, during, or shortly after birth (Eisenberg, Glueckauf, & Zaretsky, 1999). Depending on which areas of the brain are affected, one or more of the following movement disorders can occur: stiff muscles (spasticity), uncontrollable movements (dyskinesia), or poor balance and coordination (ataxia). CP is classified into four main types: spastic, dyskinetic, ataxic, and mixed. Spastic CP is characterized by stiff muscles and weakness. Dyskinetic CP is characterized by spontaneous, slow, uncontrolled muscle activity, and abrupt and jerky movements. Ataxic CP is characterized by poor coordination, weakness, trembling and difficulty with rapid or fine movements. Mixed CP refers to symptoms of more than one type of CP. The most common type of mixed CP is spastic-dyskinetic CP. In addition to problems with movement and posture, CP can also affect speech, vision, hearing, or intelligence. Because the symptoms of CP are primarily associated with movement disorders, many individuals with CP find it difficult or impossible to consistently and efficiently perform operations involving the physical buttons attached to a computer mouse (e.g., left clicking to select an item or activate a button, right clicking to display a context menu, double-clicking an icon to launch an application, dragging to move an icon or select text) (S. Keates, Clarkson, & Robinson, 2000; Oakley, McGee, Brewster, & Gray, 2000; Radwin et al., 1990; Rao, Seliktar, & Rahman, 2000; Roy, Panayi, Erenshteyn, Foulds, & Fawcus, 1994).

6.2 METHODS

6.2.1 Hypotheses

The same hypotheses as in Section 5.2.1 were tested.

6.2.2 Subjects

Five individuals with CP (3 Males and 2 female; age range 32-56) were recruited. The same inclusion criteria described in Section 5.2.2 were applied. Each participant's eligibility was determined based on an interview conducted before a written informed consent form was documented. All participants were recruited, and written consents were obtained, in accordance with the Institutional Review Board at the University of Pittsburgh.

All participants were individuals with quadriplegic spastic-dyskinetic CP who were active head mouse users: three of them were using infrared-based head tracking system in combination with dwell selection method and the other two used a head controlled stick pointer in combination with mouse-keys and a key-guard. All participants had used computers for at least 5 years, and their average computer usage time was more than 20 hours per week.

6.2.3 Experimental Design

The same experimental design as in Section 5.2.3 was used.

6.2.4 Equipment

The same equipment as in Section 5.2.4 was used. For head controlled stick pointer users, a infrared reflective dot was attached to their own head controlled pointers. Pointing parameters for the head-mounted mouse emulator were set to as shown in Table 32. Participants had the option to adjust these parameters but no participants chose to do so.

Table 32. Pointing Parameters for Subjects with SCI

Parameter	Value
Pointer Speed	18
Pointer Precision	off
Snap-To	off
Dwelling Time	800 msec
Permissible Movement	±5 pixels
Circling Trail Length	500 pixels
Maximal Circling Speed	< 100 pixels/tick
Minimal Circumference	> 32 pixels

6.2.5 Data Collection

The same data collection strategy as in Section 5.2.5 was used.

6.2.6 Experimental Procedures

The same experimental procedures as in Section 5.2.6 were used, except for the following:

- Subjects with CP were provided with much longer (one to three hours) training and practice opportunities compared to able-bodied subjects and subjects with SCI;
- For some participants, each session was performed on a separate day.

6.2.7 Statistical Analysis

The same statistical analysis techniques as in Section 5.2.7 were used.

6.3 RESULTS

Two subjects (Subject 3 and 4), who were using a head controlled stick pointer in combination with mouse-keys and a key-guard, could not complete all experimental conditions. So, their performance data were treated separately.

6.3.1 Full Participation

6.3.1.1 Task Completion Time

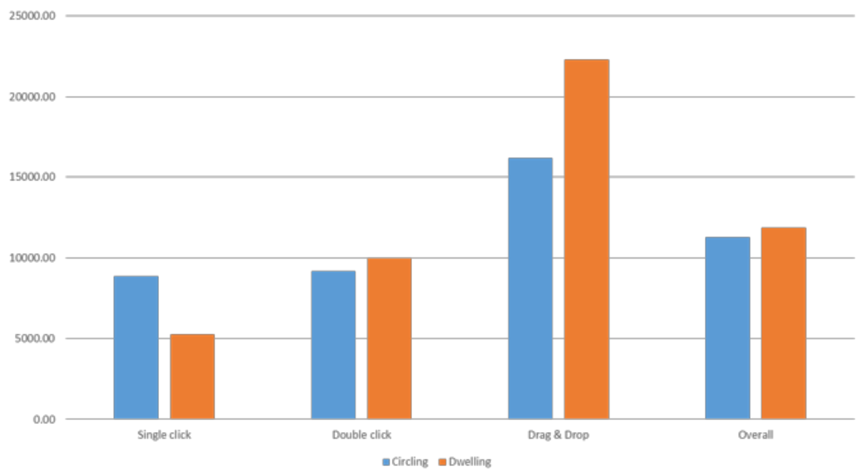

Figure 65. Task Completion time for Subjects with CP

As shown in Figure 65, for single-click operation, a Wilcoxon signed-rank test showed that there was no statistically significant difference in task completion time between the circling interface and dwell selection (Z = -12.604, p = .109). However, a large effect size of .934 implies that the circling interface (median = 8463 msec) was practically slower than the dwelling interface (median = 5385 msec).

For double-click operation, no significant difference in task completion time between the circling interface (median = 8598 msec) and dwell selection (median = 10103 msec) (Z = -1.069, p = .285) with a small effect size of .283.

For drag-and-drop operation, there was no statistically significant difference in task completion time between the circling interface and dwell selection (Z = -1.604, p = .109). However, a large effect size of .843 implieds that the circling interface (median = 15433 msec) was practically faster than the dwelling interface (median = 21184 msec).

Overall, across all operations, there was not a statistically significant difference in task completion time between the circling interface and dwell selection (Z = -1.069, p = .285). However, a medium effect size of .503 implies that the circling interface (median = 10500 msec) was equal to or faster than the dwelling interface (median = 11893 msec).

6.3.1.2 Error Rate

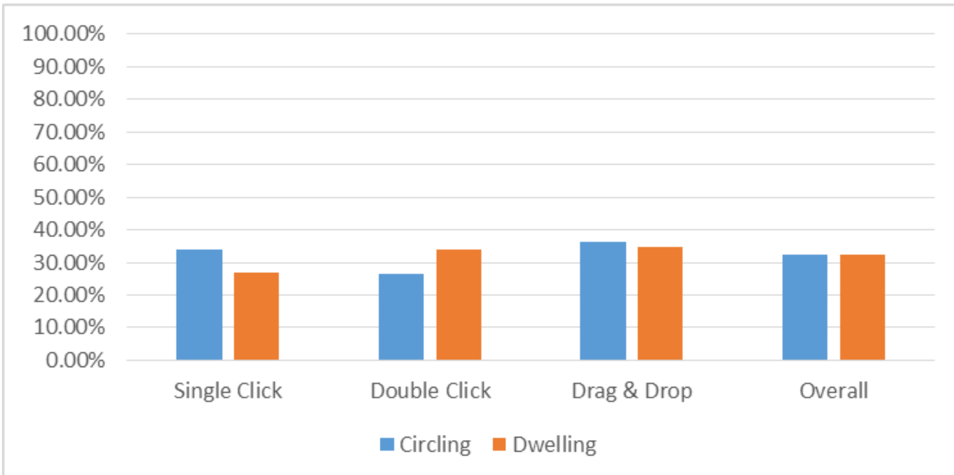

Figure 66. Error Rate for Subjects with CP

As shown in Figure 66, for single-click operation, a Wilcoxon signed-rank test showed that there was no statistically significant difference in error rate between the circling interface and dwell selection (Z = -1.604, p = .109) with a medium effect size of .51. The circling interface (median = 33%) was practically less accurate than the dwelling interface (median = 26.8%).

For double-click operation, there was not a statistically significant difference in error rate between the circling interface and dwell selection (Z = -1.604, p = .109) with a medium effect size of .53. The circling interface (median = 26.5%) was practically more accurate than the dwelling interface (median = 33.8%).

For drag-and-drop operation, there was not a statistically significant difference in error rate between the circling interface (median = 36.4%) and dwell selection (median = 34.9%) (Z = .001, p > .99).

Overall, across all pointing operations, there was not a statistically significant difference in error rate between the circling interface (median = 32.3%) and dwell selection (median = 32.2%) (Z = .001, p > .99).

6.3.1.3 Timeout

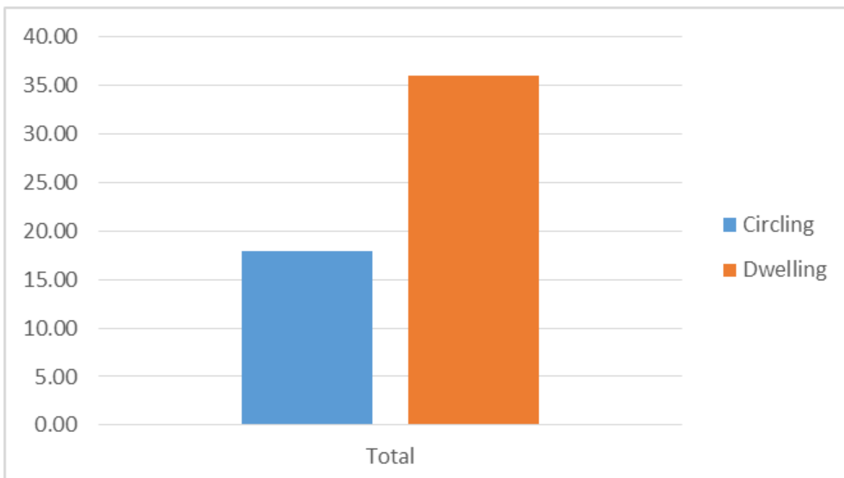

Figure 67. Timeout for Subjects with CP

Thirty second timeouts only occurred during drag-and-drop operations. Timeouts happened almost twice as often with the dwelling interface, as shown in Figure 67.

6.3.1.4 Learning Effect

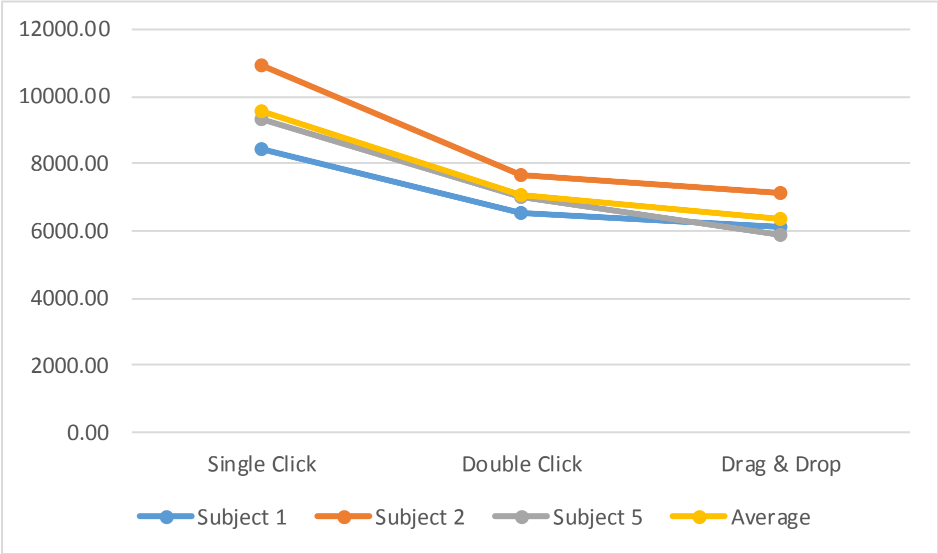

Figure 68. Learning Effect on Task Completion Time for Subjects with CP

As shown in Figure 68, a significant learning effect on task completion time with the circling interface was detected ($\chi^2(2) = 4.667$; $p = .097$). Task completion time decreased over the three sessions across all subjects.

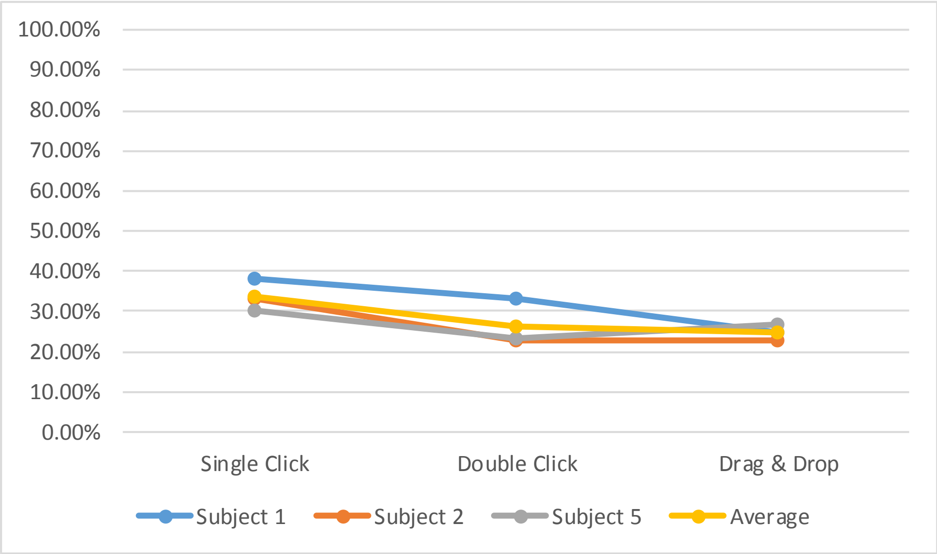

Figure 69. Learning Effect on Error Rate for Subjects with CP

As shown in Figure 69, on average, no statistically significant learning effect for error rates with the circling interface was detected ($\chi^2(2) = 4.909$; p = .086). However, the error rate did decrease consistently over the three sessions for two of the three subjects (the error rate between double-click and drag-and-drop operations slightly increased for Subject 3).

6.3.1.5 Satisfaction Level

The statistical results are shown in Table 33.

Table 33. Satisfaction Level for Subjects with CP

	Statement	SC	DC	DD	Overall
A	The tasks or task elements required much mental and perceptual effort (e.g. thinking, deciding, calculating, remembering, looking, searching, etc.).	.157	1.000	.317	1.000
B	The tasks or task elements required much physical effort (e.g. pushing, pulling, turning, controlling, activating, etc.)?	.102	.180	.157	.414
C	I felt time pressure due to the rate of pace at which the tasks or task elements occurred.	.157	.102	.317	.180
D	It was difficult to learn how to use the technology.	.317	.157	.317	.285
E	It was difficult to accomplish the goals of the task set by the experimenter.	.180	.102	.157	.276
F	I felt insecure, discouraged, irritated, stressed and/or annoyed during the tasks or task elements.	1.000	1.000	1.000	.655
G	This technology will benefit me and improve my computer access.	.102	.317	.180	1.000
H	I am confident I know how to use this technology and its various features.	.180	.083	.157	.109
I	I will feel comfortable (and not self-conscious) using this technology at home and in public.	.317	1.000	1.000	.317
J	I will recommend this technology to others who may benefit.	.317	.317	.317	.317

(SC: single-click; DC: double-click DD: drag-and-drop)

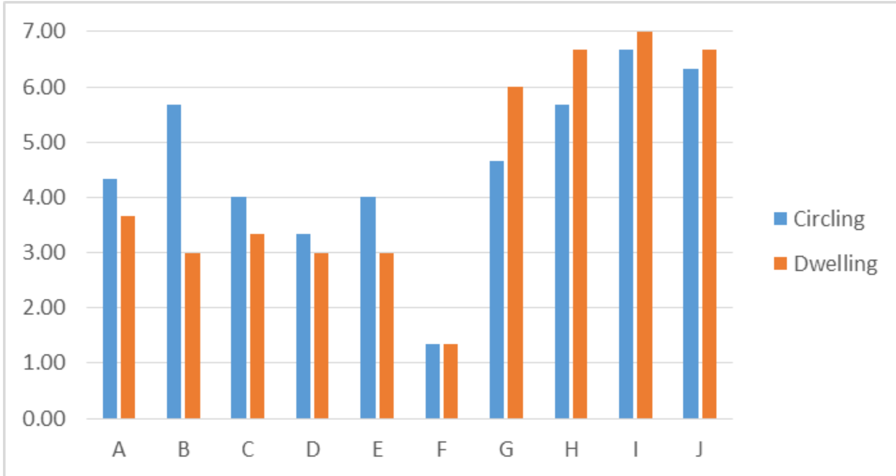

Figure 70. Single-Click Satisfaction Level for Subjects with CP

As shown in Figure 70, for single-click operation, no statistically significant difference was detected between the circling interface and dwell selection . However, the circling interface imposed practically higher mental, physical, and temporal demands, and higher performance and effort than dwell selection. As a whole, dwell selection was preferred over the circling interface in terms of improvement of computer access, the confidence level in using the technology, the comfort level in using at home, at work and in the community, and the possibility of distributing the technology to others.

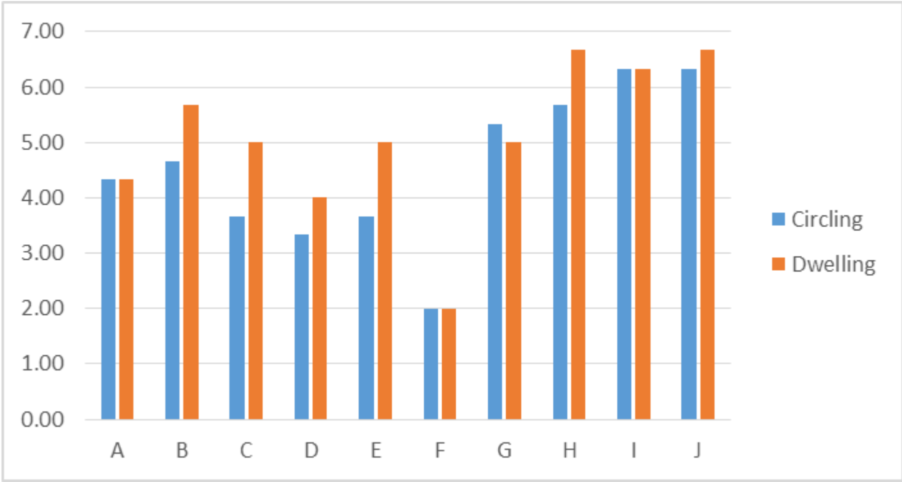

Figure 71. Double-Click Satisfaction Level for Subjects with CP

As shown in Figure 71, for the double-click operation, no statistically significant difference was detected between the circling interface and dwell selection. However, the circling interface imposed practically less physical, temporal demands, and performance and effort demands than dwell selection. And the dwelling interface was still rated more positively than the circling interface in terms of confidence level in using the technology.

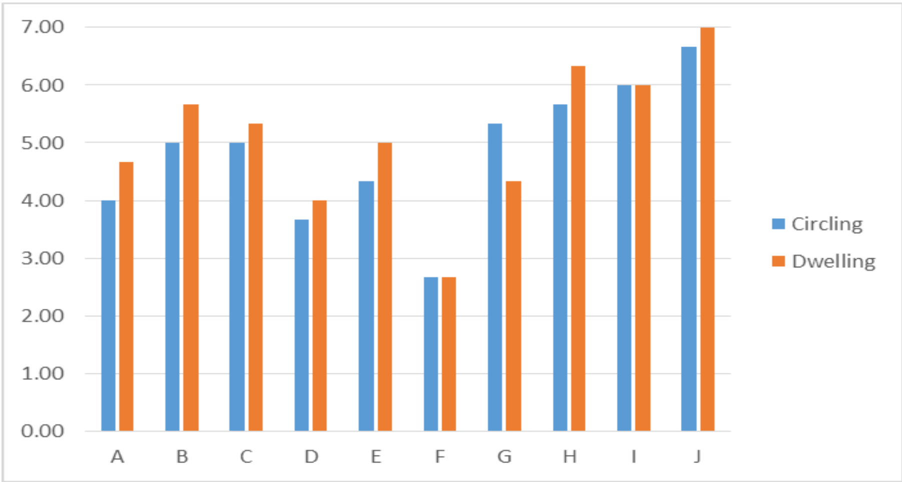

Figure 72. Drag-and-Drop Satisfaction Level for Subjects with SCI

As shown in Figure 72, for the drag-and-drop operation, no statistically significant difference was detected between the circling interface and dwell selection. However, the circling interface imposed slightly less mental, physical, temporal demands, and performance and effort demands than dwell selection.

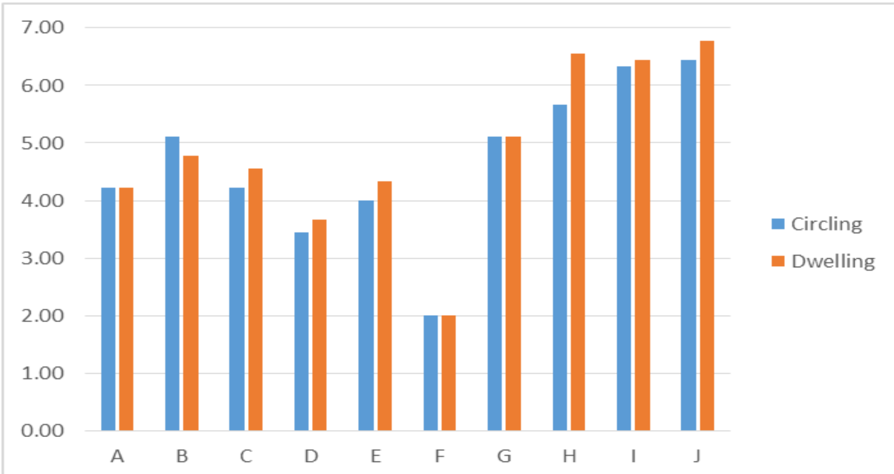

Figure 73. Overall Satisfaction Level for Subjects with CP

As shown in Table 73, no statistically significant difference was detected across in all items. However, the dwelling interface still had higher confidence levels than the circling interface.

6.3.2 Partial Participation

The task completion time data from the two subjects with CP (Subject 3 and Subject 4) who were using a head controlled stick pointer were compared using exploratory data inspection. The data were from 120 successfully-completed single click trials with the lowest ID and sparse target density.

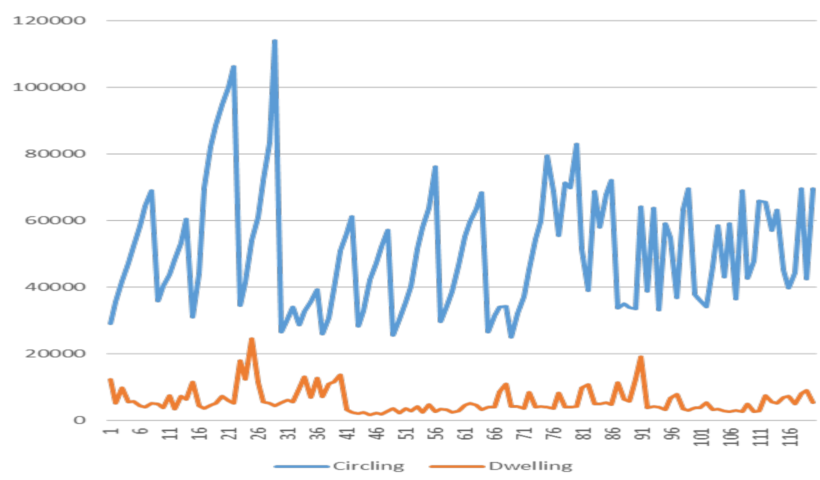

Figure 74. Data from Subject 3

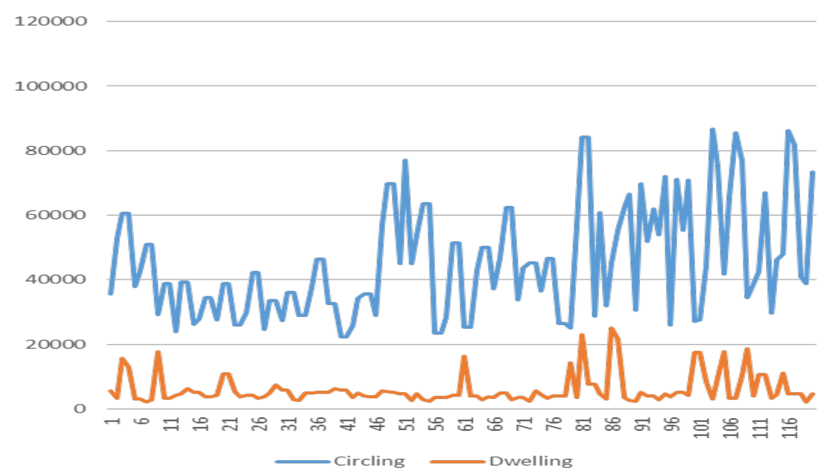

Figure 75. Data from Subject 4

As shown in Figure 74 and Figure 75, the dwelling interface was faster than the circling interface in all trials. There was no evidence that task completion time with the circling interface was improving over time. A primary reason for both participants' difficulty in using the circling interface was their unique cursor movement patterns (See Figure 76). To accommodate this issue, adjusting the screen resolution to reduce the maximum possible distance between the mouse cursor and the target and adjusting the CDG were tried, but neither was successful.

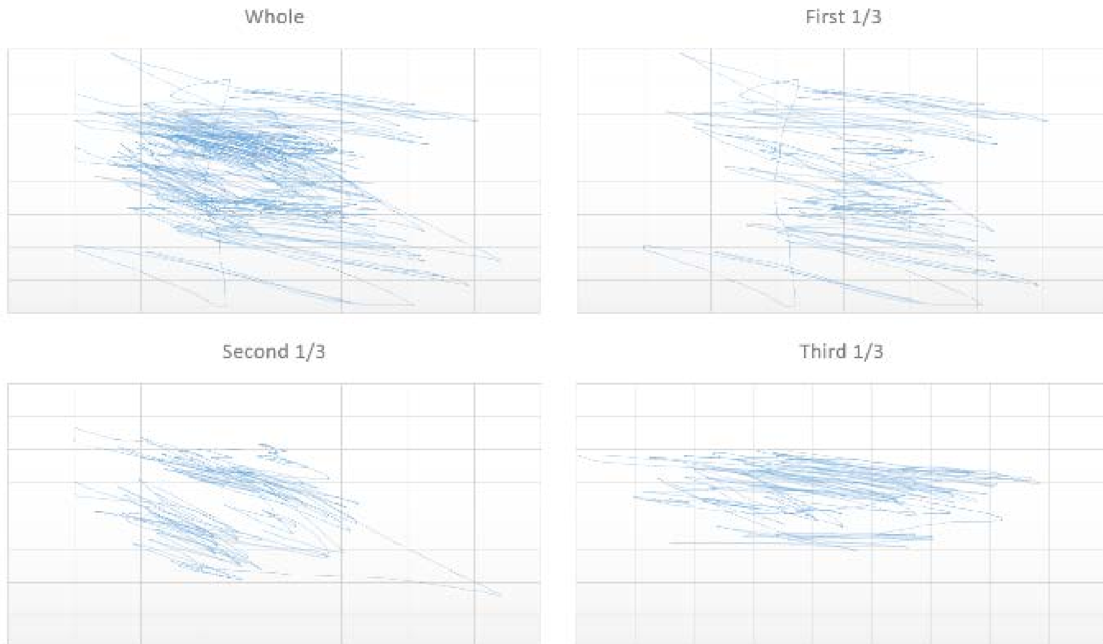

Figure 76. Plots of Cursor Trajectory

6.4 DISCUSSION

The circling interface and dwell selection had equal task completion times for subjects with CP across all tasks when using a head-mounted mouse emulator. For the single-click operation, the circling interface was slower than dwell selection (though the difference was not statistically significant). One possibility was that the distance traveled by the mouse cursor in the circling interface was longer than those of the other methods because the mouse cursor had to go to, and then all the way around, each target. Inspecting the recorded mouse cursor trajectory data revealed the distance traveled by the mouse cursor in the circling interface was, on average, 37.7% longer. For both double-click and drag-and-drop operations, the circling interface was faster than the dwelling interface (but the difference, again, was not statistically significant). This was mainly because the dwelling interface required additional time to specify the operation compared to the circling interface (Figure 77). In addition, the dwelling interface requires more precise pointing ability than the circling interface, which uses the edges of the screen to specify which command is triggered by the mouse cursor (e.g., left edge for left-click; right edge for right-click; top edge for double-click; bottom edge for drag). Locating objects along an edge decreases target selection time (Accot & Zhai, 2002; Walker et al., 1993), because the screen edge emulates a target with an infinite effective width that the mouse cursor cannot overshoot (Atwood, 2006; Hale, 2007).

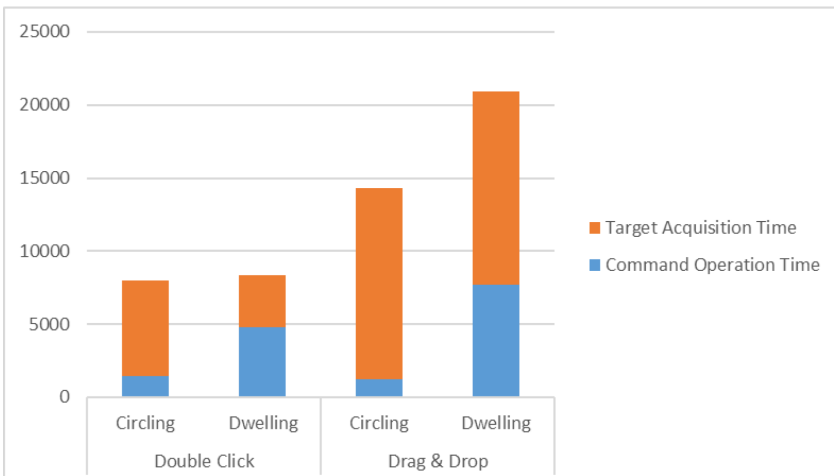

Figure 77. Command Operation Time vs. Target Acquisition Time (for error-free trials)

As shown in Figure 78, across all operations, no evidence was found that either selection method adheres to Fitts's Law. In addition, no statistically significant difference in performance time was detected based on target density in either selection method.

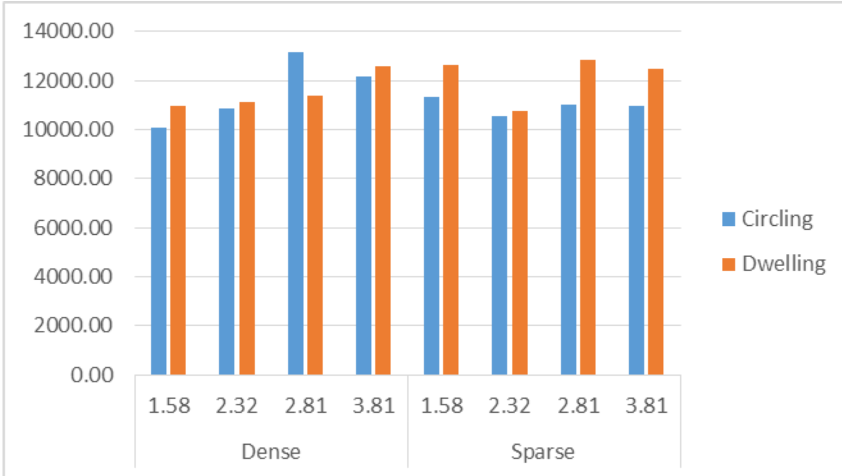

Figure 78. Task Completion Time by ID and Density

In terms of error rate, there was no difference between the circling interface and dwell selection. Out of total 180 errors, 73% were made by circling on an area with no target, 16% by circling the wrong target, 4% by circling with the wrong command specified, 6% by jerky movements, and 1% by an abnormally tiny circle (See Figure 79).

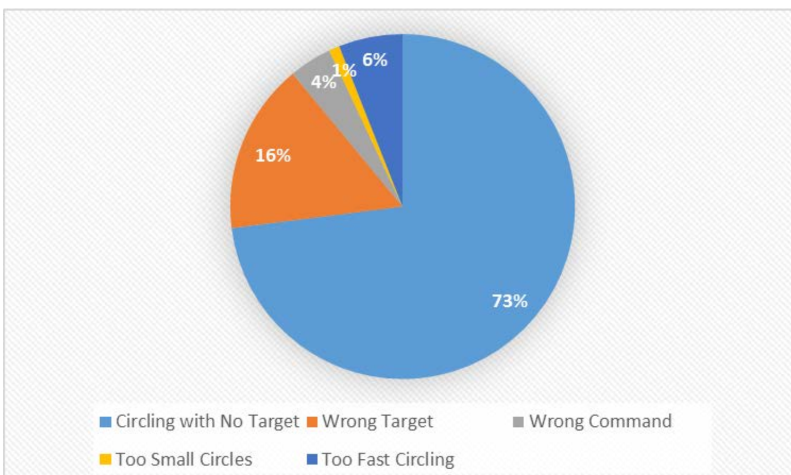

Figure 79. Portion of Each Error Type

If errors caused by circling on an area with no target and unintentional circling caused by jerky movements and an abnormally tiny circle were automatically corrected by the circling interface, the performance accuracy of the circling interface would be significantly improved (See Figure 80). However, this might not work in practice, since sometimes people want to click in an open space (for example, when dragging and dropping). One possible way to address this issue is to make the error correction technique specific to the active application and the current context. Another option is to only eliminate circles that are too small or that are made too quickly (both of which indicate that they are unintentional).

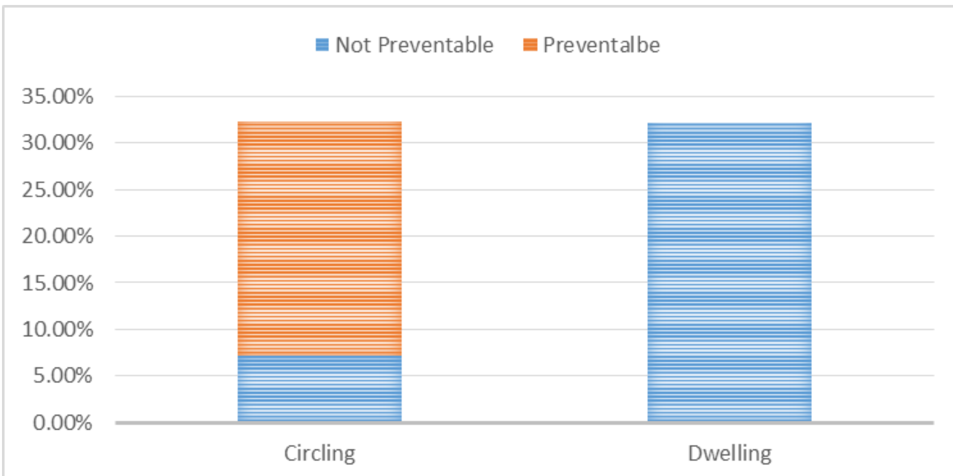

Figure 80. Overall Error Rate for Subjects with CP

Consistent improvement patterns in both task completion time and error rate with the circling interface were found, which suggests that the circling interface does not have a steep learning curve and more training and practice opportunities will effectively enhance performance. Users' subjective ratings of the circling interface support this as well. In almost all items, the differences between the two methods decreased across sessions.

Taking into account that the thirty second timeout happened almost twice as often with the dwelling interface, the usefulness of the circling interface as an alternative pointing technique for some individuals with CP may be more obvious. However, because the circling interface requires more frequent directional changes in mouse cursor movements, it will not work for some individuals who have tendency to make long straight mouse movements (Smith et al., 1999; Trewin & Pain, 1999a). It was also found that individuals with CP require a longer (25%) cursor trail to circle the target. This may be due to the difficulty with fine movements.

7.0 SUMMARY AND CONCLUSIONS

7.1.1 Summary

We developed and evaluated an alternative interaction method, called the circling interface, for selecting and manipulating on-screen objects based on circling the target rather than pointing and clicking for individuals with disabilities who find it difficult or impossible to consistently and efficiently perform pointing operations involving the left and right mouse buttons. Through a formative design approach, involving frequent interaction with collaborators (including potential users, assistive technology practitioners, an occupational therapist and ergonomic expert, and a software engineer) and numerous design revisions, essential design issues (how to specify the on-screen target; how to specify a pointing command; and what feedback should be provided) associated with the circling interface were addressed.

To specify a target of interest, the user makes a circling motion around the target. It is not necessary for the circling motion to be circular. Any closed curve (ranging from a simple triangle to a whole circle) is treated as a trigger for the circling interface. If more than one potential target is circled the circling interface presents a magnified version of the circled area to allow the user to specify the desired target.

To specify a desired pointing command (e.g., single-click, double-click, and drag-and-drop) with the circling interface, each edge of the screen is used (e.g., left edge for left-click;

right edge for right-click; top edge for double-click; bottom edge for drag). The user selects a command before circling the target.

To provide intermediate feedback so that any recognition delay does not negatively affect the user's performance, the circling interface provides the user with visual and auditory feedback. The trail of the cursor not only provides users with direct visual feedback on the cursor's movement, but it also defines the range of the available circling area. A beeping sound is provided when the user makes a circle or specifies a pointing command by reaching one of screen edges. When performing drag-and-drop operation, a colored rectangular box appears around the specified target to indicate that dragging is ready.

We implemented test-bed software that allows us to present a variety of realistic pointing tasks which can be completed by circling the target with the mouse cursor, pressing a mouse button, or using dwell-click software. With the test-bed software, we conducted empirical evaluations with human subjects from three different groups (individuals without disability, individuals with spinal cord injury, and individuals with cerebral palsy), comparing each group's performance on pointing tasks with the circling interface to performance on the same tasks when using a mouse button or dwell-clicking software.

In the experiment with able-bodied subjects, we tested the hypothesis that performance speed and accuracy for pointing operations would be different for different combinations of selection method (circling interface, physical mouse buttons, and dwell-clicking software) and input device (normal computer mouse, head-mounted mouse emulator, joystick mouse). Across all pointing operations, the traditional mouse and mouse button was the fastest and the most accurate combination. For the single-click operation, the circling interface showed slower performance than dwell selection, because the distance traveled by the mouse cursor in the circling interface was longer than that of the dwelling interface. However, for both double-click and drag-and-drop operations, the circling interface produced faster performance than the dwelling interface, because dwell selection required additional time to specify the operation as well as more precise pointing ability. While there is no evidence that the circling interface was faster than dwell selection when using a normal mouse, the circling interface showed significantly faster performance when using alternative pointing devices. Dwelling had fewer errors than circling for single-click operation but more errors for double-click and drag-and-drop operations across all input devices. When using a joystick, the circling interface made remarkably fewer errors than dwell selection. Regarding the perceived workload measured with NASA-TLX, the traditional mouse and manual selection had the lowest workload. With a normal mouse the dwelling interface imposed significantly lower workload than the circling interface, but no significant difference in difficulty between the circling and dwelling interface was found for both the head-mounted mouse emulator and the joystick mouse. Regarding the subscales, in both perceived performance and effort level, the circling interface imposed more burden than the dwelling interface with a normal mouse. This may be because the dwelling interface relies on a more familiar pointing-based approach than the circling interface. However, this lack of familiarity can be overcome by training and practice.

In the experiments with disabled subjects, we tested the hypothesis that performance speed and accuracy for pointing operations would be different between the circling interface and dwell-clicking software when using a head-mounted mouse emulator. Across all operations, for both subjects with SCI and with CP, the circling interface showed faster performance than the dwell clicking interface (although the difference was not statistically significant). For the single-click operation, the circling interface showed slower performance than dwell selection, but for both double-click and drag-and-drop operations, the circling interface produced faster performance. Subjects with CP required much longer time to complete the tasks compared to subjects with SCI. In terms of performance accuracy, while for subjects with SCI, dwell selection had fewer errors than the circling interface, for subjects with CP, there was no difference in accuracy between the two methods.

Across all three groups, the circling interface was faster than the dwelling interface (although the difference was not statistically significant). For the single-click operation, the circling interface was slower than dwell selection, but for both double-click and drag-and-drop operations, the circling interface was faster. In terms of performance accuracy, the results were mixed: for able-bodied subjects circling was more accurate than dwelling, for subjects with SCI the opposite was true, and for subjects with CP there was no difference. However, as an error correction technique, if errors caused by circling on an area with no target or by ignoring circles that are too small or too fast were automatically corrected by the circling interface, the performance accuracy of the circling interface would significantly outperform dwell selection across all three groups. However, this might not work in practice, since sometimes people want to click in an open space (for example, when dragging and dropping). One possible way to address this issue is to make the error correction technique specific to the active application and the current context. Another option is to only eliminate circles that are too small or that are made too quickly (both of which indicate that they are unintentional).

For able-bodied subjects and subjects with SCI, when selecting densely packed targets, the circling interface conformed to Fitts's Law, but when selecting sparsely packed targets, the circling interface was barely affected by ID. The circling interface showed better performance with sparsely packed small targets than with sparsely packed large targets, because smaller movements were required to circle small targets.

Consistent improvement patterns in both task completion time and error rate with the circling interface across all groups were found, which suggests that the circling interface does not have a steep learning curve and more training and practice opportunities will effectively enhance performance. Users' subjective ratings of the circling interface support this as well. In almost all items, the differences between the two methods decreased across sessions.

Our target user population consists of people who find it difficult or impossible operate standard mouse buttons, including people using head-mounted mouse emulators and other pointing devices that do not have physical buttons and people who have trouble using physical buttons on devices like mice, trackballs and joysticks. However, because the circling interface requires more frequent directional changes in mouse cursor movements, it will not work for some individuals who have a tendency to make long straight mouse movements.

7.1.2 Limitations and Future Direction

All empirical evaluations conducted in this research project were based on the basic pointing tasks modeled by Fitts's law (Fitts, 1954; I. S. MacKenzie, 1992) in a controlled laboratory setting. Researchers have found that this approach was a reliable and valid method for measuring pointing performance for able-bodied and disabled computer users (Koester, LoPresti, & Simpson, 2011; Koester, Simpson, Spaeth, & LoPresti, 2007; Rao et al., 2000). However, data from this research still have limitations in that:

- The given tasks does not directly represent real situations that people with pointing problems encounter in their day-to-day computer use;

- Data collection was performed with unfamiliar equipment in unfamiliar environments;
- Subject performance was being observed by the experimenter;
- Variability in performance might be affected by fatigue or boredom due to the simple and repetitive nature of the tasks.

Thus, it is desirable not only to replicate the results from this research study, but also to collect data from pointing tasks performed in real-world computer applications. We are planning on developing a standalone agent application that allows users to operate a number of real-world applications with the circling interface.

Although this research study demonstrated the advantage of the circling interface over dwell selection across all three groups, findings from this study should be generalized with caution, because the results from each disabled group was drawn from a small sample. In addition, the subjects with disabilities were experienced computer users and did not represent the users who recently acquired relevant disability and needed an alternative pointing method to be prescribed. In addition, this research covered just small segments of potential user population that benefits from the circling interface. Therefore, we desire not only to recruit more subjects with SCI and CP for the circling interface to have stronger clinical evidence, but also to collect data from the users who have different physical disabilities and different skill levels, since user needs and abilities are extremely diverse.

The current circling interface relies exclusively on the built-in accessibility API provided by Microsoft Windows operating system to expose the properties of all standard UI elements (e.g., button, edit box, list control, combo box, icon, scroll, and etc.) to the client software module. The major limitation of this approach is that it does not support nonstandard UI elements, which are frequently used to make a UI look more aesthetically pleasing. To overcome this issue, third party accessibility APIs should be considered, as well as techniques that can automatically recognize targets currently displayed on the screen (Hurst, 2010).

7.1.3 Conclusions

The purpose of this research project was to develop and evaluate an alternative interaction method, the *circling interface*, for selecting and manipulating on-screen objects based on circling the target, rather than pointing and clicking. The software algorithm allows individuals with disabilities to circle targets to perform pointing operations that currently require using the left or right mouse buttons or dwell-clicking software. The empirical evaluations demonstrated that for some individuals with disabilities, a circling interface would be more efficient and accurate than alternatives based on dwell-clicking and would make computers easier to use.

The primary target user population of the circling interface consists of individuals who have trouble using the standard mouse buttons and has difficulty with dwell selection. They do not need to operate physical buttons or switches; do not need to keep the cursor positioned over a target long enough to use dwell-clicking; and they can use their existing pointing devices. For the single-click operation, they may experience slower performance with the circling interface than existing solutions, but for both double-click and drag-and-drop operations, the circling interface is faster. If errors caused by circling on an area with no target and unintentional circling caused by jerky movements and an abnormally tiny circle are automatically corrected by the circling interface, their performance accuracy with the circling interface will be also expected to outperform existing solutions without a steep learning curve. This suggests that the circling interface can be used in conjunction with existing techniques and this kind of combined approach achieve more effective mouse use for people with pointing problems.

Consequently, the circling interface can improve clinical practice by providing another alternative pointing method that does not require physically activating mouse buttons and is more efficient than dwell-clicking. It is also expected to be useful for both computer access and augmentative communication software. In addition, as a software-based solution, the circling interface will be relatively easy to integrate into new and existing products.

APPENDIX A

DATA COLLECTION FORMS

A.1.1 Initial Questionnaire

* Please mark a [x] or circle the number that represents your answer, or fill in the blank with suitable words.

1. Gender:

   ① Male [ ]

   ② Female [ ]

2. Age:

   ____

3. What is the primary reason for your disability?

   ____

4. How long have you lived with your disability?

   ____

5. How long have you been using a computer?

① Less than 1 year [ ]

② Less than 2 years [ ]

③ Less than 5 years [ ]

④ More than 5 years [ ]

6. On average, how many hours per week do you use your computer?

(If you are having trouble thinking of an average, think back over the last two weeks.)

① Less than 2 hours [ ]

② Less than 5 hours [ ]

③ Less than 20 hours [ ]

④ More than 20 hours [ ]

7. What kind of pointing methods are you using now? (Please check all items applicable to you)

① Manual Computer Mouse [ ]

② Head-mounted Pointer [ ]

③ External Switch Interface [ ]

④ Dwell-clicking Software [ ]

⑤ Others: ______________________________

8. How long have you used the above pointing methods?

____

9. Have you ever changed any of the pointing device settings for your primary pointing device (movement speed, click lock, click speed)? If yes, please describe any of the settings that you prefer to use.

① Yes [ ]

② No [ ]

If yes, please describe any of the settings that you prefer to use.

____

____

A.1.2 NASA-TLX Questionnaire

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date

Mental Demand — How mentally demanding was the task?

Very Low — Very High

Physical Demand — How physically demanding was the task?

Very Low — Very High

Temporal Demand — How hurried or rushed was the pace of the task?

Very Low — Very High

Performance — How successful were you in accomplishing what you were asked to do?

Perfect — Failure

Effort — How hard did you have to work to accomplish your level of performance?

Very Low — Very High

Frustration — How insecure, discouraged, irritated, stressed, and annoyed wereyou?

A.1.3 User Satisfaction Questionnaire

Please rate each method on the following 10 items (A-J) according to the following scale (1 to 7) in the appropriate box.

1	2	3	4	5	6	7
Disagree						Agree

	Statement	Circling Interface	Dwell-Clicking Interface
	The tasks or task elements required much mental and perceptual effort (e.g. thinking, deciding, calculating, remembering, looking, searching, etc.).		
	The tasks or task elements required much physical effort (e.g. pushing, pulling, turning, controlling, activating, etc.)?		
	I felt time pressure due to the rate of pace at which the tasks or task elements occurred.		
	It was difficult to learn how to use the technology.		
	It was difficult to accomplish the goals of the task set by the experimenter.		
	I felt insecure, discouraged, irritated, stressed and/or annoyed during the tasks or task elements.		
	This technology will benefit me and improve my computer access.		
	I am confident I know how to use this technology and its various features.		
	I will feel comfortable (and not self-conscious) using this technology at home and in public.		
	I will recommend this technology to others who may benefit.		

APPENDIX B

POINTING DEVICES USED IN THIS RESEARCH

		
Normal Mouse	Head-Mounted Mouse Emulator	Joystick Mouse

BIBLIOGRAPHY

Accot, J., & Zhai, S. (2002). *More than dotting the i's - Foundations for crossing-based interfaces*. Paper presented at the Conference on Human Factors in Computing Systems, Minneapolis, MN.

Ahlström, D., Hitz, M., & Leitner, G. (2006). *An evaluation of sticky and force enhanced targets in multi target situations*. Paper presented at the 4th Nordic Conference on Human-Computer Interaction: Changing Roles, Oslo, Norway.

Anson, D. K. (1997). *Alternative Computer Access: A Guide to Selection* (1 ed.). Philadelphia, PA: F.A. Davis Company.

Arch, A. (2009). *Web accessibility for older users: successes and opportunities (keynote).* Paper presented at the Proceedings of the 2009 International Cross-Disciplinary Conference on Web Accessibililty (W4A).

Arch, A. (2010). Web accessibility for older users: A literature review.

Arnaut, L. Y., & Greenstein, J. S. (1990). Is display/control gain a useful metric for optimizing an interface? *Human Factors: The Journal of the Human Factors and Ergonomics Society, 32*(6), 651-663.

Atwood, J. (2006, August 9, 2006). Fitts' Law and Infinite Width. Retrieved May 3, 2013, 2013, from http://www.codinghorror.com/blog/2006/08/fitts-law-and-infinite-width.html

Bates, R., & Istance, H. (2002). *Zooming interfaces!: enhancing the performance of eye controlled pointing devices.* Paper presented at the Proceedings of the fifth international ACM conference on Assistive technologies.

Becker, S. A. (2004). A study of web usability for older adults seeking online health resources. *ACM Transactions on Computer-Human Interaction (TOCHI), 11*(4), 387-406.

Blanch, R., Guiard, Y., & Beaudouin-Lafon, M. (2004). *Semantic pointing: improving target acquisition with control-display ratio adaptation.* Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems.

Bowker, N. I. (2010). Understanding barriers to online experience for people with physical and sensory disabilities using discursive social psychology. *Universal Access in the Information Society, 9*(2), 121-136. doi: DOI 10.1007/s10209-009-0162-3

Casali, S. P., & Chase, J. D. (1993). *The effects of physical attributes of computer interface design on novice and experienced performance of users with physical disabilities.* Paper presented at the Proceedings of the Human Factors and Ergonomics Society Annual Meeting.

Casiez, G., Vogel, D., Balakrishnan, R., & Cockburn, A. (2008). The impact of control-display gain on user performance in pointing tasks. *Human-Computer Interaction, 23*(3), 215-250. doi: Doi 10.1080/07370020802278163

Chen, C.-L., Chen, H.-C., Cheng, P.-T., Chen, C.-Y., Chen, H.-C., & Chou, S.-W. (2006). Enhancement of Operational Efficiencies for People With High Cervical Spinal Cord Injuries Using a Flexible Integrated Pointing Device Apparatus. *Archives of Physical Medicine and Rehabilitation, 87*, 866-873.

Choe, E. K., Shinohara, K., Chilana, P. K., Dixon, M., & Wobbrock, J. O. (2009). *Exploring the design of accessible goal crossing desktop widgets*. Paper presented at the Conference on Human Factors in Computing Systems, Boston, MA.

Cockburn, A., & Brewster, S. (2005). Multimodal feedback for the acquisition of small targets. *Ergonomics, 48*(9), 1129-1150. doi: 10.1080/00140130500197260

Cockburn, A., & Firth, A. (2003). Improving the acquisition of small targets.

Cook, A. M., Dobbs, B. M., Warren, S., & McKeever, R. (2005). Measuring target acquisition utilizing Madentec's Tracker system in individuals with Cerebral Palsy. *Technology and Disability, 17*(3), 155-163.

Cushman, L. A., & Scherer, M. J. (1996). Measuring the relationship of assistive technology use, functional status over time, and consumer-therapist perceptions of ATs. *Assistive Technology, 8*(2), 103-109.

Czaja, S. J., & Lee, C. C. (2007). The impact of aging on access to technology. *Universal Access in the Information Society, 5*(4), 341-349.

Dobransky, K., & Hargittai, E. (2006). The disability divide in Internet access and use. *Information, Communication & Society, 9*(3), 313-334.

Doherty, E., Cockton, G., Bloor, C., & Benigno, D. (2000). *Mixing oil and water: transcending method boundaries in assistive technology for traumatic brain injury.* Paper presented at the Proceedings on the 2000 conference on Universal Usability.

Douglas, S. A., Kirkpatrick, A. E., & MacKenzie, I. S. (1999). *Testing pointing device performance and user assessment with the ISO 9241, Part 9 standard.* Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit.

Douglas, S. A., & Mithal, A. K. (1994). *The effect of reducing homing time on the speed of a finger-controlled isometric pointing device.* Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence.

Drainoni, M. L., Houlihan, B., Williams, S., Vedrani, M., Esch, D., Lee-Hood, E., & Weiner, C. (2004). Patterns of Internet use by persons with spinal cord injuries and relationship to health-related quality of life. *Archives of Physical Medicine and Rehabilitation, 85*(11), 1872-1879. doi: DOI 10.1016/j.apmr.2004.07.350

Eisenberg, M. G., Glueckauf, R. L., & Zaretsky, H. H. (1999). *Medical aspects of disability: A handbook for the rehabilitation professional*: Springer Publishing Company.

Evans, D. G., Drew, R., & Blenkhorn, P. (2000). Controlling mouse pointer position using an infrared head-operated joystick. *Ieee Transactions on Rehabilitation Engineering, 8*(1), 107-117. doi: Doi 10.1109/86.830955

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *J Exp Psychol, 47*(6), 381-391.

Fox, L. E., Sohlberg, M. M., Fickas, S., Lemoncello, R., & Prideaux, J. (2009). Public computing options for individuals with cognitive impairments: survey outcomes. *Disabil Rehabil Assist Technol, 4*(5), 311-320. doi: 10.1080/17483100902980943

Goldman, R. (1990). *Intersection of two lines in three-space.* Paper presented at the Graphics gems.
Grossman, T., & Balakrishnan, R. (2005). *The bubble cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area*. Paper presented at the SIGCHI Conference on Human Factors in Computing Systems, Portland, OR.
Guiard, Y., Blanch, R., & Beaudouin-Lafon, M. (2004). *Object pointing: A complement to bitmap pointing in GUIs*. Paper presented at the Graphics Interface 2004, London, Ontario, Canada.
Hale, K. (2007). Visualizing Fitts' Law. Retrieved May 9, 2013, 2013, from http://particletree.com/features/visualizing-fittss-law
Hanson, V. L. (2009). *Age and web access: the next generation.* Paper presented at the Proceedings of the 2009 International Cross-Disciplinary Conference on Web Accessibililty (W4A).
Hart, S. G. (2006). *NASA-task load index (NASA-TLX); 20 years later.* Paper presented at the Proceedings of the Human Factors and Ergonomics Society Annual Meeting.
Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Human mental workload, 1*, 139-183.
Hertzum, M., & Hornbaek, K. (2010). How Age Affects Pointing With Mouse and Touchpad: A Comparison of Young, Adult, and Elderly Users. *International Journal of Human-Computer Interaction, 26*(7), 703-734. doi: Pii 923324182
Doi 10.1080/10447318.2010.487198
Hinckley, K. (2002). Input technologies and techniques. *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*, 151-168.
Holt, B. (2000). Creating senior-friendly Web sites. *Issue Brief Cent Medicare Educ, 1*(4), 1-8.
Hornof, A., Cavender, A., & Hoselton, R. (2004). *Eyedraw: a system for drawing pictures with eye movements.* Paper presented at the ACM SIGACCESS Accessibility and Computing.
Hurst, A. (2010). Using Data About Real World Pointing Performance to Improve Computer Access with Automatic Assessment.
Hwang, F., Keates, S., Langdon, P., & Clarkson, J. (2004). *Mouse movements of motion-impaired users: a submovement analysis*.
Ka, H., & Simpson, R. (2010). *On-screen Target Acquisition through Mouse Movement Recognition.* Paper presented at the RESNA Anual Conference, Las Vegas, CA.
Kabbash, P., & Buxton, W. A. S. (1995). *The "prince" technique: Fitts' law and selection using area cursors.* Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems.
Keates, S., Clarkson, J., & Robinson, P. (2000). *Investigating the applicability of user models for motion-impaired users.* Paper presented at the Proceedings of the fourth international ACM conference on Assistive technologies.
Keates, S., Hwang, F., Langdon, P., Clarkson, P., & Robinson, P. (2002). The use of cursor measures for motion-impaired computer users. *Universal Access in the Information Society, 2*(1), 18-29.
Keates, S., Hwang, F., Langdon, P., Clarkson, P. J., & Robinson, P. (2002). *Cursor measures for motion-impaired computer users.* Paper presented at the Proceedings of the fifth international ACM conference on Assistive technologies.

Keates, S., Hwang, F., Langdon, P., Clarkson, P. J., & Robinson, P. (2002). The use of cursor measures for motion-impaired computer users. *Universal Access in the Information Society*, *2*(1), 18-29.

Keates, S., & Trewin, S. (2005). *Effect of age and Parkinson's disease on cursor positioning using a mouse*. Paper presented at the 7th International ACM SIGACCESS Conference on Computers and Accessibility, Baltimore, MD.

Keir, P. J., Bach, J. M., & Rempel, D. (1999). Effects of computer mouse design and task on carpal tunnel pressure. *Ergonomics, 42*(10), 1350-1360. doi: 10.1080/001401399184992

Ketcham, C. J., Seidler, R. D., Van Gemmert, A. W. A., & Stelmach, G. E. (2002). Age-related kinematic differences as influenced by task difficulty, target size, and movement amplitude. *Journals of Gerontology Series B-Psychological Sciences and Social Sciences, 57*(1), P54-P64.

Keyson, D. V. (1997). Dynamic cursor gain and tactual feedback in the capture of cursor movements. *Ergonomics, 40*(12), 1287-1298. doi: 10.1080/001401397187379

Koester, H., LoPresti, E., & Simpson, R. (2011). *Further Validation of The "AIM" Test for Assesing a User's Mouse Skills.* Paper presented at the RESNA 2011 Annual Conference, Toronto, ON, Canada.

Koester, H., Simpson, R., Spaeth, D., & LoPresti, E. (2007). *Reliability and Validity of Compass Software for Access Assessment*. Paper presented at the RESNA, Arlington, VA.

Lankford, C. (2000). *Effective eye-gaze input into windows.* Paper presented at the Proceedings of the 2000 symposium on Eye tracking research & applications.

Lin, M. L., Radwin, R. G., & Vanderheiden, G. C. (1992). Gain Effects on Performance Using a Head-Controlled Computer Input Device. *Ergonomics, 35*(2), 159-175. doi: Doi 10.1080/00140139208967804

Lin, V., Bono, C., Cardenas, D., Frost, F., Lindblom, L., Hammond, M., . . . Woolsey, R. (2010). *Spinal Cord Medicine: Principles & Practice*: Demos Medical.

Lipscomb, J. S., & Pique, M. E. (1993). Analog input device physical characteristics. *ACM SIGCHI Bulletin, 25*(3), 40-45.

MacKenzie, I. S. (1992). Fitts' Law as a research and design tool in human-computer interaction. *Human-Computer Interaction, 7*(1), 91-139.

MacKenzie, I. S. (1995). Input devices and interaction techniques for advanced computing. *Virtual environments and advanced interface design*, 437-470.

MacKenzie, I. S., Kauppinen, T., & Silfverberg, M. (2001). *Accuracy measures for evaluating computer pointing devices*. Paper presented at the SIGCHI Conference on Human Factors in Computing Systems, Seattle, WA.

MacKenzie, I. S., Sellen, A., & Buxton, W. (1991). *A comparison of input devices in elemental pointing and dragging tasks.* Paper presented at the Proceedings of ACM CHI '91 Conference on Human Factors in Computing Systems.

MacKenzie, I. S., & Ware, C. (1993). *Lag as a determinant of human performance in interactive systems.* Paper presented at the Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems.

Madara, E. (1997). The mutual-aid self-help online revolution. *Social Policy, 27*, 20-26.

Marino, R. J., Barros, T., Biering-Sorensen, F., Burns, S. P., Donovan, W. H., Graves, D. E., . . . Committee, A. N. S. (2003). International standards for neurological classification of spinal cord injury. *J Spinal Cord Med, 26 Suppl 1*, S50-56.

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

McKenna, K., & Bargh, J. (2000). Plan 9 from cyberspace: The implications of the Internet for personality and social psychology. *Personality and Social Psychology Review, 4*(1), 57.

McKenna, K., & Seidman, G. (2005). You, me, and we: Interpersonal processes in electronic groups. In Y. Amichai-Hamburger (Ed.), *The social net: Human behavior in cyberspace* (pp. 191-217). Oxford, UK: Oxford University Press.

Mei, L. I. L. I. N., Robert, G. R., & Gregg, C. V. (1992). Gain effects on performance using a head-controlled computer input device. *Ergonomics, 35*(2), 159-175.

Mizobuchi, S., & Yasumura, M. (2004). *Tapping vs. Circling Selections on Pen-based Devices: Evidence for Different Performance-Shaping Factors*. Paper presented at the Conference on Human Factors in Computing Systems, Vienna, Austria.

Moffatt, K., & McGrenere, J. (2009). Exploring Methods to Improve Pen-Based Menu Selection for Younger and Older Adults. *ACM Transactions on Accessible Computing, 2*(1), 1-34. doi: 10.1145/1525840.1525843

Morahan-Martin, J., & Schumacher, P. (2003). Loneliness and social use of the Internet. *Computers in Human Behavior, 19*(6), 656-671.

Myers, C. S. (1980). *A comparative study of several dynamic time warping algorithms for speech recognition.* Massachusetts Institute of Technology.

Noyes, J. M., & Bruneau, D. P. (2007). A self-analysis of the NASA-TLX workload measure. *Ergonomics, 50*(4), 514-519. doi: 10.1080/00140130701235232

Oakley, I., McGee, M. R., Brewster, S., & Gray, P. (2000). *Putting the feel in'look and feel '*. Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems.

Patmore, D. W., & Knapp, R. B. (1998). *Towards an EOG-based eye tracker for computer control.* Paper presented at the Proceedings of the third international ACM conference on Assistive technologies.

Radwin, R. G., Vanderheiden, G. C., & Lin, M. L. (1990). A Method for Evaluating Head-Controlled Computer Input Devices Using Fitts Law. *Hum Factors, 32*(4), 423-438.

Rao, R. S., Seliktar, R., & Rahman, T. (2000). Evaluation of an isometric and a position joystick in a target acquisition task for individuals with cerebral palsy. *IEEE Trans Rehabil Eng, 8*(1), 118-125.

Richards, J. T., & Hanson, V. L. (2004). *Web accessibility: a broader view.* Paper presented at the Proceedings of the 13th international conference on World Wide Web.

Robertson, G., Card, S. K., & Mackinlay, J. D. (1989). *The cognitive coprocessor architecture for interactive user interfaces.* Paper presented at the Proceedings of the 2nd annual ACM SIGGRAPH symposium on User interface software and technology.

Rogers, W. A., Stronge, A. J., & Fisk, A. D. (2005). Technology and aging. *Reviews of human factors and ergonomics, 1*(1), 130-171.

Roy, D. M., Panayi, M., Erenshteyn, R., Foulds, R., & Fawcus, R. (1994). *Gestural human-machine interaction for people with severe speech and motor impairment due to cerebral palsy.* Paper presented at the Conference companion on Human factors in computing systems.

Rubine, D. (1991). *Specifying gestures by example* (Vol. 25): ACM.

Rutledge, J. D., & Selker, T. (1990). *Force-to-motion functions for pointing.* Paper presented at the Proceedings of the IFIP TC13 Third Interational Conference on Human-Computer Interaction.

Sandfeld, J., & Jensen, B. R. (2005). Effect of computer mouse gain and visual demand on mouse clicking performance and muscle activation in a young and elderly group of experienced computer users. *Appl Ergon, 36*(5), 547-555. doi: 10.1016/j.apergo.2005.03.003

Schaab, J. A., Radwin, R. G., Vanderheiden, G. C., & Hansen, P. K. (1996). A comparison of two control-display gain measures for head-controlled computer input devices. *Hum Factors, 38*(3), 390-403.

Scherer, M. J., & Cushman, L. A. (2000). Predicting satisfaction with assistive technology for a sample of adults with new spinal cord injuries. *Psychol Rep, 87*(3 Pt 1), 981-987.

Scherer, M. J., & Cushman, L. A. (2001). Measuring subjective quality of life following spinal cord injury: a validation study of the assistive technology device predisposition assessment. *Disability and Rehabilitation, 23*(9), 387-393.

Scherer, M. J., & Galvin, J. C. (1996). An outcome perspective of quality pathways to most appropriate technology. In J. C. Galvin & M. J. Scherer (Eds.), *Evaluating, selecting and using appropriate assistive technology* (pp. 1-26). Gaithersburg, MD: Aspen Publication.

Sears, A., & Young, M. (2002). *Physical disabilities and computing technologies: an analysis of impairments.* Paper presented at the The human-computer interaction handbook.

Simpson, R., Koester, H. H., & Lopresti, E. (2010). Research in computer access assessment and intervention. *Phys Med Rehabil Clin N Am, 21*(1), 15-32. doi: 10.1016/j.pmr.2009.07.006

Smith, M. W., Sharit, J., & Czaja, S. J. (1999). Aging, motor control, and the performance of computer mouse tasks. *Hum Factors, 41*(3), 389-396.

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

Stafford, T. (2005, January 20, 2005). Size and selection times: Fitts's Law. Retrieved May 9, 2013, 2013, from http://mindhacks.com/2005/01/20/size-and-selection-times-fittss-law/

Stevenson, B., & McQuivey, J. L. (2003). The Wide Range of Abilities and Its Impact on Computer Technology (Vol. http://www.microsoft.com/enable/research/phase1.aspx). Cambridge, MA: A Research Study Commissioned by Microsoft Corporation and Conducted by Forrester Research.

Tappert, C. C. (1982). Cursive Script Recognition by Elastic Matching. *IBM Journal of Research and development, 26*(6), 765-771.

Taveira, A. D., & Choi, S. D. (2009). Review Study of Computer Input Devices and Older Users. *International Journal of Human-Computer Interaction, 25*(5), 455-474. doi: Doi 10.1080/10447310902865040

Thomas, C. L. (1989). *Taber's cyclopedic medical dictionary* (Vol. 16): Springer.

Trewin, S., Keates, S., & Moffatt, K. (2006). *Developing Steady Clicks: A method of cursor assistance for people with motor impairments*. Paper presented at the 8th International ACM SIGACCESS Conference on Computers and Accessibility, Portland, OR.

Trewin, S., & Pain, H. (1999a). Keyboard and mouse errors due to motor disabilities. *International Journal of Human-Computer Studies, 50*(2), 109-144. doi: DOI 10.1006/ijhc.1998.0238

Trewin, S., & Pain, H. (1999b). A model of keyboard configuration requirements. *Behaviour & Information Technology, 18*(1), 27-35. doi: Doi 10.1080/014492999119228

Walker, N., Meyer, D. E., & Smelcer, J. B. (1993). Spatial and temporal characteristics of rapid cursor-positioning movements with electromechanical mice in human-computer interaction. *Hum Factors, 35*(3), 431-458.

Walker, N., Philbin, D. A., & Fisk, A. D. (1997). Age-related differences in movement control: adjusting submovement structure to optimize performance. *J Gerontol B Psychol Sci Soc Sci, 52*(1), P40-52.

Wobbrock, J. O., Cutrell, E., Harada, S., & MacKenzie, I. S. (2008). *An error model for pointing based on Fitts' law.* Paper presented at the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.

Wobbrock, J. O., Fogarty, J., Liu, S.-Y. S., Kimuro, S., & Harada, S. (2009). *The angle mouse: target-agnostic dynamic gain adjustment based on angular deviation.* Paper presented at the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.

Wobbrock, J. O., & Gajos, K. Z. (2008). Goal crossing with mice and trackballs for people with motor impairments: Performance, submovements and design directions. *ACM Transactions on Access Computing, 1*(1), 1-37.

Wobbrock, J. O., & Gajos, K. Z. (2008). Goal crossing with mice and trackballs for people with motor impairments: Performance, submovements, and design directions. *ACM Transactions on Accessible Computing (TACCESS), 1*(1), 4.

Worden, A., Walker, N., Bharat, K., & Hudson, S. (1997). *Making computers easier for older adults to use: area cursors and sticky icons.* Paper presented at the Proceedings of the ACM SIGCHI Conference on Human factors in computing systems.

World Health Organization. (2011). World Report on Disability. *The Lancet, 377*(9782).

Xiao, Y. M., Wang, Z. M., Wang, M. Z., & Lan, Y. J. (2005). [The appraisal of reliability and validity of subjective workload assessment technique and NASA-task load index]. *Zhonghua Lao Dong Wei Sheng Zhi Ye Bing Za Zhi, 23*(3), 178-181.

Young, M. A., Levi, S., Tumanon, R. C., Desei, M., & Sokal, J. O. (2000). Independence for people with disabilities. A physician's primer on assistive technology. *Md Med, 1*(3), 28-32.

Zeleznik, R. C., Herndon, K. P., & Hughes, J. F. (2007). *SKETCH: an interface for sketching 3D scenes.* Paper presented at the ACM SIGGRAPH 2007 courses.