Universität Konstanz
Fachbereich Informatik & Informationswissenschaft
Master-Studiengang Information Engineering

Masterarbeit

Laserpointer and Eye Gaze Interaction Design and Evaluation

von

Hans-Joachim Bieg

Zur Erlangung des akademischen Grades eines
Master of Science (M. Sc.)

Studienfach: Information Engineering
Schwerpunkt: Computer Science
Themengebiet: Angewandte Informatik

Matr.-Nr.: 01/618524
Erstgutachter: Prof. Dr. Harald Reiterer
Zweitgutachter: Prof. Dr. Oliver Deussen

Abstract

Interacting with modern graphical user interfaces usually requires a pointing device like for instance the mouse. The pointing speed and accuracy that can be achieved with a pointing device has a considerable influence on the usability of these systems. In this thesis a comprehensive introduction is given to the methods for the evaluation of speed and precision of pointing devices based on the specifications of the international standard ISO 9241-9. Furthermore, several improvements of these methods are proposed and discussed, that should improve the comparability of future evaluations.

Effects of practice are one particular issue in the assessment of pointing device performance. Due to time and cost restrictions, only little time is typically scheduled for training with each device in comparative pointing device evaluations. This has the disadvantage that alterations in performance due to practice remain unconsidered. In order to examine the effects of practice on pointing performance, a longitudinal experiment was conducted with four participants. For each participant several sessions were scheduled on five consecutive days. In each of these sessions participants carried out a simple tapping tests with a novel laserpointer interaction system that was developed by the Human-Computer Interaction Group of the University of Konstanz. The experiment was carried out at the Powerwall at the University of Konstanz, a large high-resolution display (5.20 × 2.15 m). The results of the experiment showed that practice has a considerable effect on the pointing performance of the laserpointer. Performance increases were discernible until session four. This motivates a more careful consideration of practice effects in forthcoming comparative evaluations. Furthermore, a novel type of test based on a discrete tapping task is presented that was employed in the experiment. The playful character of the test should maintain participants' motivation across several sessions.

In the last section of the thesis, the prospects of gaze interaction are discussed by giving an overview of the core issues and existing systems. Furthermore, three novel gaze interaction techniques are presented. Finally, first experiences and observations on gaze interaction for large high-resolution displays with a mobile eye-tracker are reported.

Zusammenfassung

Für die Interaktion mit modernen grafischen Benutzeroberflächen werden meistens Zeigegeräte wie beispielsweise die Maus benötigt. Die Präzision und Geschwindigkeit mit der ein Zeigegerät benutzt werden kann hat einen erheblichen Einfluss auf die Gebrauchstauglichkeit dieser Systeme. In der vorliegenden Arbeit wird eine umfassende Einführung in die Methode zur Evaluation der Präzision und Geschwindigkeit von Zeigegeräten auf Basis der Vorgaben des internationalen Standards ISO 9241-9 gegeben. Im Weiteren werden mehrere Verbesserungen dieser Methoden vorgeschlagen und diskutiert, die zu einer größeren Vergleichbarkeit der Ergebnisse zukünftiger Evaluationen führen sollen.

Ein spezielles Problem bei der Beurteilung der Leistung von Zeigegeräten sind Übungseffekte. Wegen erhöhtem Zeit und Kostenaufwand werden Vergleichsstudien von Zeigegeräten typischerweise nur mit wenig Trainingszeit pro Gerät durchgeführt. Dies hat den Nachteil, dass übungsbedingte Leistungsveränderungen unberücksichtigt bleiben. Um den Einfluss von Übungseffekten auf die Zeigeleistung zu untersuchen wurde in dieser Arbeit ein Langzeitexperiment mit vier Teilnehmern durchgeführt. Dazu wurden für jeden Teilnehmer mehrerer Sitzungen an fünf aufeinanderfolgenden Tagen abgehalten. In jeder Sitzung führten die Teilnehmer einfache Anticktests mit einem neuartigen Laserpointer Interaktionssystem durch das von der Arbeitsgruppe Mensch-Computer Interaktion der Universität Konstanz entwickelt wurde. Das Experiment wurde an der Powerwall der Universität Konstanz durchgeführt, einem 5.20 × 2.15 m großen hochauflösenden Display. Die Ergebnisse des Experiments zeigen, dass Übungseffekte einen wesentlichen Einfluss auf die Zeigeleistung des Laserpointers hatten. Leistungssteigerungen waren bis zur vierten Sitzung erkennbar. Dies motiviert eine sorgfältigere Berücksichtigung von Übungseffekten in zukünftigen Vergleichsexperimenten. Darüberhinaus wird ein eine neue Art von Test basierend auf einer diskreten Antickaufgabe vorgestellt, die in dem Experiment eingesetzt wurde. Der spielerische Charakter dieses Tests soll dazu Beitragen die Motivation der Teilnehmer über mehrere Sitzungen aufrecht zu erhalten.

Im letzten Abschnitt der Arbeit werden die Möglichkeiten der Blickgesteuerten Interaktion diskutiert indem ein Überblick über die Kernprobleme und existierende Systeme gegeben wird. Darüberhinaus werden drei neue blickbasierte Interaktionstechniken vorgestellt. Abschließend wird über erste Erfahrungen und Beobachtungen zur blickgesteuerten Interaktion für große hochauflösende Displays mit einem mobilen Eye-tracker berichtet.

Contents

1 Introduction

1.1 Motivation

The following example illustrates the pervasiveness of pointing in today's user interfaces. Table 1 shows the sum of continuous mouse movement, the total distance the pointer was moved, and the clicks made during 11 regular days of programming, web or word-processor use. The data was recorded using Workrave[1] , a small utility that tracks computer usage, scheduling regular breaks or "micro pauses" to reduce the risk of RSI (repetitive strain injury). On average, the mouse was continuously moved 46 minutes a day, traveling a mean distance of 495 m and bearing 1566 clicks. With 52 weeks a year, 5 working days each plus 0.5 of spare-time computer use and 20 days of computer-free vacation, this amounts to the enormous sum of 12236 minutes or 203 hours of continuous mouse movement. Thus, average information workers likely spend around eight and a half days of continuous mouse movement a year.

Table 1: Mouse Usage

Day	MT (min.)	Distance (m)	Clicks
1	40	342.1	1438
2	35	289.0	912
3	34	261.8	1009
4	45	377.3	1379
5	41	297.5	1260
6	54	410.5	1930
7	39	313.4	1421
8	43	282.8	1367
9	69	475.1	2565
10	66	495.2	2549
11	42	285.6	1397
Mean	46	495.2	1566

MT = Continuous mouse movement time

As can be seen from this example, pointing with a pointing device such as the mouse plays a fundamental role in today's highly interactive graphical user interfaces. Pointing is an "(...) act of communication with the system (...)" (Douglas & Mithal, 1997). The pointing device is used to select items that have some meaning to the user and the computer, for instance, an icon which symbolizes a specific document or a button which stands for a certain action. These basic selections can be seen as the "words" that make up the dialogue with the computer. In contrast to the command-line interfaces of the early days of computing, "talking" to a computer by selecting items is much more user-friendly. Instead of having to remember which words are understood by the computer, the user can simply pick a word from the set that is offered to him through the graphical display. In fact, "dialogue" seems to be entirely inappropriate to describe the interac-

---

[1] http://www.workrave.org/

tion experience with modern *direct-manipulation* user interfaces. Instead of being a partner in a dialogue about a world of things that are of interest to the user, the interface "(...) is itself a world where the user can act" (Hutchins, Hollan, & Norman, 1985). The objects of this world can be manipulated in a similar fashion to objects in the real world and pointing devices are usually the tools that extend users' hands from the real world to the world of the interface.

1.2 Outline

The pointing speed and accuracy that can be achieved with a pointing device has a considerable influence on the usability of the system as a whole. A common method to assess the performance of a pointing device is to carry out an experiment with simple one- and multidirectional pointing and selection tests. These tests are proposed by the international standard ISO 9241 part 9 and are based on experimental paradigms that were originally used by researchers seeking to understand human motor behavior.

In the first chapter of this thesis a brief introduction is given about computer input and pointing devices, followed by an overview about Fitts' Law, a model of human motor behavior in chapter two. Fitts' Law is closely associated with pointing device performance tests and also used in the field of human-computer interaction to assess and predict task times of interactive systems or system concepts. A summary is provided covering methods and procedures required to carry out pointing device tests that are compliant to the ISO standard 9241-9, followed by a critical review of the procedures proposed by the standard.

One particular aspect that is criticized about the standard is that only few recommendations are made on how to treat effects of practice during a pointing device test. In order to investigate this issue further, a review of literature was conducted to examine how practice is commonly accounted for in pointing device tests. A summary of findings is provided in chapter three, followed by the description of a longitudinal experiment that was conducted to assess the effects of practice for a particular device: a laserpointer interaction system developed by the Human-Computer Interaction Group of the University of Konstanz. Additionally, a novel testing application is presented. The application was designed to be similar to a game and provided performance feedback to motivate participants during the longitudinal study.

Finally in chapter five, an overview is given about how pointing and selection tasks can be carried out by observing users' gaze with an eye-tracker. The question is addressed how accurate and fast pointing with the eyes is and what the main issues of this technique are. Based on a literature review, various techniques are presented that seek to overcome these issues followed by an outline of three novel interaction techniques that combine gaze and manual input. The chapter is concluded by giving and outlook on mobile gaze-tracking in combination with large high-resolution displays followed by a report of observations of a first test of gaze-based interaction techniques.

2 Computer Input and Pointing Devices

In the following a brief overview is given on computer input devices, in particular pointing devices. The most important properties are presented that can be used to distinguish between different classes of pointing devices. The chapter is concluded by discussing aspects that determine the usability of a pointing device.

2.1 Input Devices

Input devices are "(...) those pieces of hardware by which a user enters information into a computer system” (Foley, van Dam, Feiner, & Hughes, 1997). In a more general sense, input devices are sensors or rather, a bundle of sensors. That is, certain pieces of hardware that can sense one or many properties such as position, force, speed, velocity, rotation, etc. Hinckley (2002) is a bit more general in writing that “input devices sense physical properties of people, places, or things”. In this definition not even a human user is required for the operation of the device; users can also be places or things. But of course the scope must be narrowed further. A weather station also senses a whole lot of things and a computer networking card also senses something. These may be called “input devices” in a literal sense, but the main characteristic of the input devices that are of concern for this thesis are the ones that are *directly and voluntarily operated by a human*. That is, the user manipulates aspects of the device, such as its location or the pressure applied to certain parts of it, to change the value of the properties it senses. This change is then processed by the computer, or rather, by a couple of components that belong to the computer system. Usually, the sensor data is pre-processed directly by a microcontroller on the device and then sent to the main hardware of the computer where the operating system device driver interprets and transforms the sensor data into events for the application (Douglas & Mithal, 1997).

In the following sections, two major classifications are presented that seek to systematize the variety of input devices. The first classification is based on the type of property that a device senses. The second is based on the point of view of the user and places input devices in the context of the types of tasks that could be accomplished with a device.

2.1.1 Virtual Devices

An early classification of input devices was proposed by Foley and Wallace (1974) who distinguished between six *virtual input devices* which were modeled after real devices that existed at that time. The classification is based on the type of data, or rather, the way the data delivered by the devices’ sensors.

Table 2: Virtual Devices, adapted from (Wallace, 1976)

Device	Data	Prototype
Pick	Reference	Lightpen
Button	Button code	Function Key
Keyboard	Text and cursor	Keyboard
Locator	Position/orientation	Mouse, Joystick, Tablet
Valuator	(Real) value	Potentiometer

The main reasoning behind this classification was to specify an abstraction so that application programmers did not have to care about the intricacies of programming for a specific input device. Instead, they were now able to write applications that were more portable in the sense that it could be used on different systems that had different implementations of the same virtual devices. Indeed, the notion of virtual devices, also called *logical devices*, was picked up by a major effort for the creation of a standard graphics package specification, the Core System, by the Graphics Standards Planning Committee of the ACM (GSPC, 1979). The logical devices of the Core System were the same as those of Foley and Wallace plus an additional device, the stroke, which provided a number of positions as output data.

The Core System specification provides concise information on how devices should be initialized, how data from devices should be made available to applications in the form of events or samples, or which basic “echo facilities”, that is, simple graphical feedback such as an on-screen pointer for locator devices, should be provided by the graphics package. Furthermore, it is mentioned that the graphics system should provide facilities to simulate certain input devices. For instance, if an implementation for a pick is not available, the pick could be simulated using a locator. This is the case with most of today’s graphical user interfaces that are used with a mouse. The mouse is not used to directly pick an item but to indirectly move a pointer that is then used as a picking device. On the other hand, if no locator is available, it could be simulated by using a pick and dragging a crosshair object across the screen. The need for the simulation mentioned in the last example primarily arose from the limitations of the first light pens, which were only able to identify objects but not positions on the screen. However, a similar technique is often employed in modern handheld devices if the touch sensitive screen is primarily operated in “picking-mode”. This is for instance the case in Windows Vista when used with a tablet PC, where the stylus is primarily used to select objects and not to merely specify a position. However, if one simply wants to position the on-screen pointer this is also possible by picking a translucent mouse-symbol that floats beside the cursor and dragging it to a new location, which also repositions the real pointer.

A standard similar to the GSPC’s Core System, though restricted to 2D graphics at first, was the Graphical Kernel System (GKS), which also became an ISO standard in 1985 (Hopgood, Duce, Gallop, & Sutcliffe, 1986; ISO 7942:1985). In GKS the set of logical input devices and other specifications mentioned before were practically identical to the one of the Core System. The notion of virtual or logical devices was also picked up by Foley and colleagues in their Simple Raster Graphics Package (SRGP), which is for instance, used in their popular book on computer graphics fundamentals (Foley et al., 1997). What is common to all those approaches, is that they describe input devices from the point of view of the programmer, in terms of the data that they provide.

2.1.2 Interaction Tasks and Techniques

Instead of looking at devices from a programmer's point of view, the taxonomy by Foley, Wallace, and Chan (1984) takes on the perspective of the user, classifying between devices on the basis of the basic tasks that they support. These *interaction tasks* are "(...) meaningful in the context of the application" (Foley et al., 1984). They are the basic meaningful units of interaction – the "words" of the input language that can be used to construct even more complex interaction sequences to instruct the computer system. Foley et al. suggest that there are six types of indivisible, user-oriented interaction tasks:

- Select: Choosing an item from a set of several alternatives, for instance a command in a list of commands or one or a group of objects from several objects.
- Position: Specifying a position in one, two, or three-dimensional space.
- Orient: Rotating an entity in two- or three-dimensions around one or several axes.
- Path: Generating a series of positions.
- Quantify: Specifying a number.
- Text: Creating a string of characters.

A crucial element regarding the definition of an interaction task is that it conveys meaning and that subdivisions of such a basic task do not. One could argue that selecting a menu item with the mouse actually involves a positioning task. It certainly does but not in the sense of a basic interaction task. Positioning the mouse does not convey any meaningful information in this example. It usually does not change the state of the computer system apart from the basic pointing echo of the mouse – the on-screen pointer. The task becomes meaningful not until the selection is completed by some means, for instance, clicking a mouse button while pointing at the item or dwelling over the item for some period of time. Some relevant information is now communicated to the system: the choice among several menu items. On the other hand, when the position information is "(...) put to some application purpose (...)" (Foley et al., 1997, p. 358), for instance when drawing a line or dragging a graphical item to a new position, this positioning can be considered a basic task. It conveys some relevant information: a choice among several possible coordinates, which leads to a change in the system's state such as the existence of a new line or an update in the coordinates of the object in a graphics application.

One could conceive many ways in which an input device could be used to fulfill any kind of these interaction tasks. Foley et al. call such a particular way an *interaction technique*. Put another way, an interaction technique is a binding of the sensor output of a device and the manipulations required for achieving a particular sensor state to those basic interaction tasks. A very good example for an interaction technique in the sense of Foley et al. was already described earlier in this chapter: multiple interaction techniques exist for the selection of on-screen items. With a device such as a touch panel, the item could be selected directly by touching the respective location with the stylus or finger. On the other hand, one could also use a mouse to indirectly maneuver an on-screen pointer to the location of the item and select it with a button press. One can see that different devices might require different interaction techniques. But one can also conceive of different techniques for the same device. For instance, instead of simply touching the item with the stylus, a selection could be made with a stroke gesture, marking it with a circle or crossing it out (Accot & Zhai, 2002; Apitz & Guimbretière, 2004).

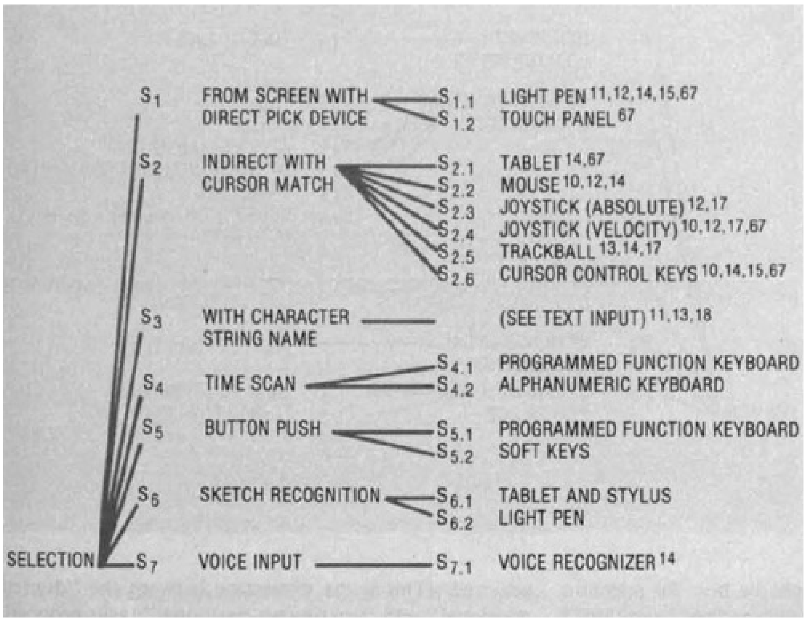


Figure 1: Several possible interaction techniques for a selection task, from (Foley et al., 1984)

An overview of some of the possible techniques for selection conceived by Foley and colleagues is shown in Figure 1. Similarly, a position could be specified directly by pointing with a mouse, typing in the coordinate values with a keyboard, or picking and dragging of a tracking symbol as described in the section before. Novel interaction techniques such as on-screen keyboards or the *Unistroke* (D. Goldberg & Richardson, 1993), a light-weight handwriting recognition technique which for instance was used in the Palm OS[2] (an operating system for mobile devices) had to be invented to enable text entry tasks for mobile devices.

Hinckley (2006) notes that there are potential problems in defining basic interaction tasks in the way it was done by Foley et al. Since all tasks are hierarchical from the perspective of the user, a task may seem “basic” although it is rather complex with regard to the use of the input device. Citing Buxton (1986), Hinckley gives the example of how scrolling in a web page and

---

[2] Interestingly though, Palm Inc. was not allowed to do so and finally paid around $22.5 million to Xerox, the patent holder of the unistroke technology, in order to settle a patent infringement lawsuit (The New York Times, 2006).

clicking on a hyperlink could either be seen as a basic navigation task or "(...) an elemental 1D positioning task followed by a 2D selection task (...)". Another objection commonly raised is that these interaction tasks could be further taken apart to even lower-level tasks. For instance, a position comprises the specification of an x- and y-value. Hinckley (2006) argues that, depending on the device used, this task may seem elemental or compound. Elemental when using for instance a mouse and compound if one uses a device with two separate handles for the x- and y-control.

2.2 Pointing Devices

The term "pointing device" loosely refers to a subclass of input devices. Examples for common pointing devices are the mouse, trackballs, graphic tablets, etc. In today's graphical direct manipulation interfaces, pointing devices play an important role (ISO 9241-16:1999). In terms of the classification of Foley et al. (1984), one of the main task that is carried out with a pointing device in these systems is the selection of objects, for instance, selection of an icon in a menu bar, a shape in a graphics application or a word in a word processor. In such a *selection-through-pointing-task* information about a position is converted into selection information, the choice of one element among several elements. This conversion requires a selection trigger. Such a trigger can for instance be a button, as is the case with the mouse. It could also occur automatically, after a certain amount of time that is measured while the pointer remains over an item that can be selected, a technique that is called "dwelling". Or triggering could be implicit, as is the case with touchscreens where the position and selection trigger are inseparable. Of course also other tasks are usually carried out with pointing devices such as dragging to move items across the screen or tracing along a path when using a scrollbar. All these tasks require the specification of one or multiple positions.

Of course, a position could also be specified with a keyboard, by entering its coordinate or by using the arrow keys as is commonly done in drawing applications to fine adjust the position of an object in a stepwise fashion. Such *discrete* pointing devices can be differentiate from *continuous* devices as those discussed above (Douglas & Mithal, 1997). Ultimately the data reported by continuous devices is also discrete, of course. For instance, the PS/2 mouse protocol sends the count of "ticks" sensed during one sampling period. These ticks represent the amount of movement detected along the x- and y-axis in a certain relation to the movement in millimeters. The pointing experience, however, is totally different because these discrete units are not detectable by the user. For the most part, this thesis deals with continuous pointing devices.

2.2.1 Properties

MacKenzie (1995) distinguishes between the properties of an input device, the "(...) qualities which distinguish among devices and determine how a device is used and what it can do" and parameters which can be "tuned" to affect the performance of a device. Two of the most important properties are presented in the following: the distinction between relative and absolute and direct and indirect. After that, in section 2.2.2 important parameters of pointing devices are discussed.

2.2.1.1 Relative or Absolute

Absolute devices such as tablets or touchscreens have a reference frame with an absolute origin. For instance, the information transmitted from a graphics tablet is the absolute location of the stylus on the tablet with respect to some of its corners and it is mapped to an absolute position on the screen. Touching the lower left-hand corner instantly moves the pointer to the lower-left hand corner of the screen. Touching the center point of the tablet area moves the pointer to the center of the screen. Some absolute devices have a fixed origin. Others require a calibration in order to specify the origin. This problem is referred to by Buxton (1983) as the "nulling problem". For instance, most handheld computing devices that are equipped with a touch sensitive screen require a four-point calibration procedure, in which the user successively points to four corner points to set-up the frame of reference and the origin that matches the output reference frame (the screen) of the device.

Relative devices on the other hand, such as the mouse, do not have an absolute origin and therefore they cannot sense absolute position. Rather than that, these devices often transmit displacement or velocity information. For instance, an isometric joystick, which is often available on laptop computers in the middle of the keyboard as for example in Figure 2, senses the force that is applied to it. This force is mapped to the velocity of the on-screen pointer. Contrary to absolute devices, the actual position of the device does not play a role with relative devices. Typically, users exploit this fact by picking up the mouse after moving it to one side and placing it down in the middle of the mouse pad to move to the same side again – an action that is also called "clutching". Since the mouse's sensor (the ball or optical sensor) is not manipulated while it is not in direct contact with the surface, no information is transmitted to the system at that moment.

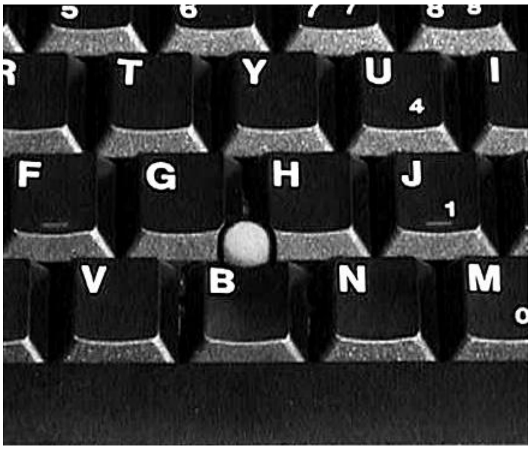


Figure 2: Isometric Joystick, from (Zhai, Smith, & Selker, 1997)

Using clutching movements with the mouse or by constantly applying force to an isometric joystick, one can specify a change in position that is theoretically infinite. The magnitude of changes in position for absolute devices, on the other hand, is constraint to the extension of the reference frame. Absolute device therefore cannot be used when infinite position changes are required. This is the case when moving a camera in a 3D environment, for instance, when playing a 3D game. Additionally, having the ability to specify large position changes gives relative devices a competitive edge in scenarios with confined movement and large pointing surfaces, as is the case with notebook computers.

On the other hand, absolute devices are better suited for tasks such as sketching and do not require clutching movements which may speed up pointing with those devices. Hinckley (2006) points out that absolute devices can also often be switched to a relative mode in which the displacement of a stylus stroke is mapped to the displacement of the pointer.

2.2.1.2 Direct or Indirect

A device is said to be direct if it has a “(...) unified input and display surface” (Hinckley, 2006, p. 6). Handheld devices often have a touch sensitive screen and pointing actions can be carried out directly by touching parts of the screen with the finger or with a stylus for more precise control.

Indirect devices such as the mouse or certain tablets that do not have a built-in display usually require that the current pointing position is represented graphically in the form of a pointer on the screen. The graphical pointer moves across the screen as the device is moved across the input surface. This enables users to judge what manipulation of the device, such as movement in a certain direction, is required to specify the desired pointing location. Essentially the pointer creates the feedback of a closed loop system that is required to carry out target directed movements. Direct devices, on the other hand, do not necessarily require such a graphical pointer, because the feedback required for control is already available from the manual pointing act itself (e.g. with a finger or stylus). Claims are made that direct devices are often easier to use for novice users because they do not require cursor control (Albinsson & Zhai, 2003).

Indirect devices usually require that a button is used to specify an object selection. The graphical pointer is maneuvered over an object and a button is pressed to indicate the selection of this object. Direct devices, on the other hand, often do not require that a button is pressed and pointing itself is directly converted into an object selection. This behavior can for instance be found in touch screens of ticket machines or other public installations and also in touch sensitive displays of handhelds. Other direct devices separate pointing from selection and allow users to move a pointer across the display without directly converting the pointing into a selection. The transition between mere pointing and selection is then carried out by pressing a button of a stylus, increasing the pressure on the display area, or shortly lifting the finger or stylus from the display (Potter, Weldon, & Shneiderman, 1988).

Occlusion – that is, covering up parts of the output area with the finger, arm or stylus – is a major problem of direct input devices, particularly if precise positioning is required with finger operated touch screens. Various techniques have been proposed by researchers to solve this problem, such as fixed pointer offsets, zooming behavior, and multi-touch techniques (Benko, Wilson, & Baudisch, 2006).

2.2.2 Parameters

2.2.2.1 Transfer Functions

A transfer function describes how the signal generated by the pointing device sensor is transformed or “mapped” to drive the movement of the pointer. Douglas (1997) distinguishes between three properties of the movement that could be affected by changes in the sensor state: position (zero-order), speed (first-order), and velocity (second-order). Hinckley (2006) points out that the type of transfer function, or mapping, heavily influences the performance that can be achieved with a device or the ease of learning of device control. That is, the pragmatics of the sensor manipulations determine which mappings are appropriate and which are not.

For instance, isometric joysticks sense force and this force is typically mapped to the velocity of the pointer, a type of mapping that is also called *rate control.* Of course, the sensor output of an isometric joystick could also be mapped to a position but it would be very hard to allot just the right amount of force in the correct direction to distinguish between several positions with such a device. Another aspect that makes isometric devices well-suited for first-order control is that these are capable of self-centering (Zhai et al., 1997). If a device does not have self-centering capabilities cursor control based on a rate mapping becomes awkward, particularly if one wants to stop the cursor motion – For instance, if the position of the mouse would be used to control the pointer velocity (Douglas & Mithal, 1997, p. 40).

2.2.2.2 Gain

The mathematical properties of the transfer function are often called the gain, that is, the ratio between the magnitude of changes in the position of the pointer on the screen and the magnitude of the sensor manipulation such as the position change or velocity of the movement (Douglas & Mithal, 1997, p. 41). This ratio is often called C/D gain or control-to-display gain, whereas sometimes the reverse is used, the D/C display-to-control gain. Gain is measured in physical units, that is, inches or millimeters and therefore the screen resolution is important to determine the actual gain value.

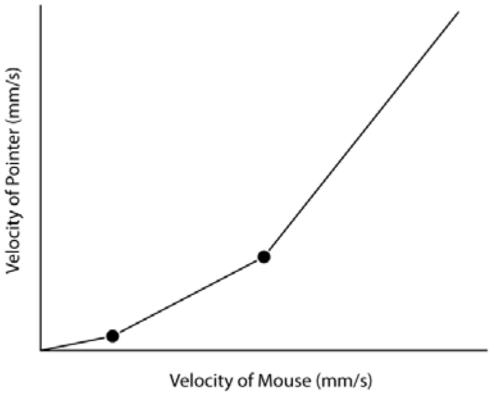


Figure 3: Variable Gain Function, adapted from (Microsoft Corp., 2002)

Often the gain factor is not constant (see Figure 3). The main advantage of a non-linear gain factor compared to a fixed ratio is that a good trade-off can be achieved between the ability to point to single pixels and the amount of manipulation required to move the pointer large distances. For instance, the Windows XP transfer function uses a higher D/C ratio for fast mouse movements than for slow ones, giving a “(...) high degree of precision and stability at low velocities” (Microsoft Corp., 2002). Non-linear gain factors have no impact on the actual pointing performance in terms of a speed-accuracy trade-off as it is presented later in chapter 3. Jellinek and Card (1990) carried out an experiment to investigate “powermice” (variable gain mice) and found that variable gain settings did not affect performance. Nevertheless, variable gain settings were preferred, presumably because they reduced the device footprint, that is, the amount of space required for using the device. Earlier computer systems had screens with lower resolution and therefore fixed gain settings could be used. In today’s high-resolution systems, the gap between the size and the total amount of screen space widened and therefore, pointing with a mouse on these systems requires variable gain settings to achieve reasonable footprint sizes while retaining the ability to precisely point to a single pixel.

2.2.2.3 Kinesthetic Correspondence

An almost trivial aspect, at least for 2D pointing, is that the manipulations required for moving the on-screen pointer should correspond to the way one would move a real-world object. That is, the visual feedback used in the control activity should resemble the feedback that would be available in the real world. In other words: if the hand that moves the device moves in one direction, the on-screen pointer should also move in this direction.This similarity of movements is called kinesthetic correspondence (Britton, Lipscomb, & Pique, 1978). Bedford (1994) points out that many tasks “(...) involve transforming the normal relation between motor activities and visual feedback in some way” and speculates that the computer mouse would not be as successful as it is if its use required a mirror transformation (i.e. moving the mouse to the left makes the pointer go to the right).

2.2.2.4 Filtering

Filtering mechanisms are often employed for indirect devices to smooth jittery motions of the input signal that are due to inaccuracies in the sensor hardware or inherent in the natural motion of the limbs required for manipulating the sensor. For example, Sears and Shneiderman (1991) found that a single touch on a touch screen often returned multiple pointing locations making accurate positioning difficult. In order to tackle this problem, they used a stabilization algorithm that smoothed movements based on information from previous locations. The algorithm allowed users to “fine-tune” small motions whereas large motions remained unaffected by the smoothing. In the end, this method made interaction with the touch screen slightly less direct for precise movements. In this respect, Hinckley (2006) points out that such a technique is only possible if the touch screen technique is combined with constant pointer feedback.

2.2.2.5 Lag

Lag is defined as the time between an input action and the response of the system (MacKenzie, 1995). The source of the lag can be either inherent in the pointing device itself, for instance due to the properties of the sensors used in the device. Another main reason for lag is the time between updates of the graphics presented on the screen. This can for instance be observed in interactive 3D applications when a complex scene requires too much rendering time. Of course, apart from software lag, lag is also induced by the refresh rate of the screen, though this is usually negligibly small (MacKenzie, 1995). In order to assess the effects of lag on pointing performance, MacKenzie and Ware (1993) carried out an experiment with a simple pointing task. The independent variable of this experiment was the amount of lag, ranging from 8.33 ms to 225 ms, that was present when using the pointing device. Their results showed that lag significantly affected pointing performance in their experiment. With the highest amount of lag (225 ms) error rates increased by 214% and pointing times by 64%. Based on these findings the authors concluded that lag is a "(...) major bottleneck for usability" (MacKenzie, 1995, p. 493).

2.3 Usability of Pointing Devices

Usability is generally defined as the "extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" (ISO 9241-11:1998). This is also the definition that is adopted by part 9 of the ISO standard 9241 which is concerned with the usability of stationary pointing devices which are typically used in office tasks (ISO 9241-9:2000). The most important measure for the effectiveness and efficiency of a pointing device is the performance of pointing movements in terms of the pointing speed and accuracy (Douglas & Mithal, 1997). The properties of a device and the configuration of its parameters have an influence on performance and designers seek to optimize these parameters.

However, the pointing precision and speed is not the sole determinant of the usability of a device. As advised by the ISO standard, it is also important to consider its context of use. For instance, experimental measurements of the pointing performance of isometric joysticks usually find that it is relatively low compared to that of the mouse (Soukoreff & MacKenzie, 2004). Nevertheless, such joysticks are widely used, for instance, in laptop computers and other mobile devices. The reason for this is that the context of use of mobile devices simply does not permit the use of a mouse because it requires a stable surface to be operated on and has a much larger footprint. On the other hand, an isometric joystick is much smaller than a mouse and can be built into a laptop computer because of its isometric properties. Another pragmatic advantage of the isometric joystick is that it works well when it is built into a keyboard, because the hands do not have to leave the writing position. However, the mouse is the device of choice when enough space is available because of its high pointing performance. But compared to other devices that allow for even higher precision and speed, the mouse has a similar pragmatic advantage. In order to switch from pointing to typing, the user simply leaves the mouse where it is and moves one of his hands to the keyboard. In contrast to that the pen of a pen-operated graphics tablet has to be dropped to be able to write with a keyboard and picked up again afterwards. On the other hand, pointing speed and accuracy is usually very high for pen-operated devices and they are therefore often used in a context where high pointing performance is required (e.g. a drawing application) and not frequent switching between entering text and pointing. Furthermore, the usability of a pointing device is determined by the ergonomic properties of a device, for instance, the strength required to actuate it or the flexion of the hand during use because these aspects could be risk factors for illnesses such as cumulative trauma disorders like carpal tunnel syndrome or tendonitis (Marras, 2007).

Pragmatic components that affect usability can usually be assessed by inspecting the device or measuring certain parts of it, such as its size or weight, the strength required for the actuation of a button or the positions of limbs during use. The ISO standard also provides very concrete requirements for some common pointing devices such as the mouse, trackballs, tablets, and others. These requirements are usually concrete enough so that conformance to the requirement could easily be verified via measurement or observation. For instance: “buttons should have a displacement force within the range of 0.5 N to 1.5 N until actuation” (ISO 9241-9:2000, , p. 14). The actual performance of the device in terms of its pointing speed and accuracy cannot be measured in this way but must be determined in a pointing test. An extensive review of the foundations and details of such tests is presented in the next chapter.

3 Testing Pointing Performance

The basis for testing pointing performance of input devices is a psychological model for human motor behavior: Fitts' Law. The experimental paradigms to derive the parameters of the model are widely used by researchers in the field of human-computer interaction to test the pointing performance of pointing devices. The procedures for this purpose are also standardized in the ISO standard 9241-9. Additionally, the model is used for predictions to evaluate pointing and selection times in graphical user interfaces.

Because of its importance for pointing device research, an in-depth introduction to Fitts' Law is given in this chapter. In the first section, the information theoretic foundations of Fitts' Law are explained. In section 3.2, the application areas within the field of human-computer interaction are presented. One area is of particular importance in this thesis: the evaluation of pointing devices. Therefore, in section 3.3, an account is given of input device evaluation methods followed by a critical review of these methods with respect to standardization issues of pointing device comparison procedures (section 3.4).

3.1 Theoretical Foundations of Fitts' Law

Interest in motor control research was primarily driven by military and industrial concerns around the 1950s. Studying the human motor system was not considered an end in itself. Rather, researchers wanted to improve the design of machines by "(...) combining the operator's characteristics with those of the mechanical parts" (Searle & Taylor, 1948) in order to eventually improve worker performance. This is reflected in the tasks studied by Fitts (1954) and other psychologists which involved tracking, tapping, and object transfer – tasks that resembled movements usually made in a production or military scenario such as assembly, control of vehicles, or aiming of weapons.

Psychologists in the 1950s began borrowing concepts and models from the engineering field, particularly from information theory. This laid the foundations for a new perspective on human behavior which is now known as *cognitive psychology* (Anderson, 2007). Humans were considered an information-processing system that, similar to a computer, are fed information which they store and processes, and finally put out. For instance, one of the most well-known models that fit into this scheme of thinking is the two-store model of memory proposed by Atkinson and Shiffrin (1967). This model distinguishes two different memory systems, short- and long-term memory, in analogy to volatile but quickly accessible main memory and nonvolatile but slow hard disks of computers. The concepts of information theory were particularly well received among psychologists that were concerned about motor behavior. Until today, one of the most popular laws in this field is characterized by a close analogy to the original information theoretic concepts: Fitts' Law, which describes the basic connection between the speed and the accuracy of simple grasping and pointing movements. Because the analogy to information theory is so close and to elucidate why motor control and pointing device researchers reason about human movements in terms of bits and bits per second, an introduction is given into the basics of information theory in the next section.

3.1.1 Information Theory

Seeking to understand communication from the standpoint of an engineer, Claude Shannon created a theory that had many implications for a variety of scientific fields. Besides popularizing the term "bit", his ideas lay the foundation for research that enabled the development of modern communications equipment that we use today. Shannon's theory is about measuring information in a communications system to reason about properties of such a system, for instance its capacity, that is, the amount of information that can be transmitted per unit of time. The components of a basic communications system typically are an information source which selects a message that is encoded in some form and transmitted across a communications channel. At the other end of this channel, the signal is picked up and the original message is reconstructed and passed on to a destination that makes use of this information (Shannon & Weaver, 1949).

3.1.1.1 Information and Information Measurement

The notion of information in the sense of Shannon is completely different to the one in everyday usage. Information in an information theoretic sense is not judged with respect to the actual meaning of a message to the sender or recipient. Information in an information theoretic sense is judged only with respect to the amount of *freedom of choice* between several messages and *reduction of uncertainty* with respect to which message was chosen. An example may clarify this conceptualization:

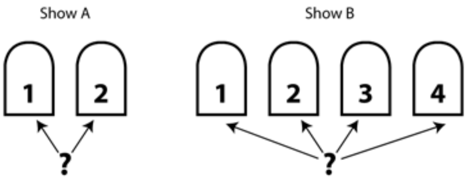


Figure 4: Game Show Example

Consider a game show where one is asked to choose among two doors and one of the doors conceals a price (see Figure 4, A). In this situation a choice can be made among two alternatives and whichever door chosen, there is uncertainty that the chosen door would lead to the price. The game show host asks the guest of the show which door he would like to open. The guest chooses door two and the uncertainty in the situation is reduced. The guest now knows which of the doors concealed the price. If a hint is given by the show's host though, for instance in the form "the price is not behind the left door", this uncertainty is reduced before the guest has to make a choice because the host already provided enough information for the guest to know which door conceals the price. But how large is this information? Is it one, since alternatives were reduced from two doors to one door? In an advanced version of the game show (see Figure 4, B), participants have to choose among four doors. This situation entails considerably more freedom of choice and thus, more uncertainty and more information. Again, the game show host gives the same hint as before. Clearly, reducing the alternatives from four to three is not the same as reducing them from two to one. The measure of information introduced by Shannon uses fractions of alternatives and is much more helpful in such decision situations than absolute numbers of choices. Its measure is the "bit". The information concealed within a decision among n equally probable outcomes has the information $\log_2 n$. Thus, in the first version of the show, the information was one bit ($n = 2$). In the advanced version the information was two bits ($n = 4$). In the easier version the host reduced the decision situation to zero bits ($n = 1$). In the advanced show, the host only reduced the information to $\sim$1.6 bits ($n = 3$).

In a more general sense Shannon proposes that the information $H$of such a decision situation can be calculated as:

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i$$

Where p_i is the probability of choice i among n choices. Consider again the situation in the show with only two alternatives. The host mentions that the probability that the price is hidden behind one of the two doors is equal, that is $p_1 = p_2 = 0.5$. This amounts to information of one bit as seen before. How much information would be involved in the situation if probabilities would not be equal? This can be easily calculated with the formula above. For instance, for the situation $p_1 = 0.75$ and $p_2 = 0.25$ the information contained within the situation would be around $H = -(0.75 \log_2 0.75 + 0.25 \log_2 0.25) \approx 0.8$ bits. If $p_1 = 1$ and $p_2 = 0$ the situation would not contain any information at all. Giving a hint with respect to the location of the price would not change anything from the point of view of the guest, since he already knows that the price is hidden behind door one. The same holds for the reverse $p_1 = 0$ and $p_2 = 1$. The amount of information is highest when probabilities are equal.

To summarize, one could say that the information is low if a situation does not involve much freedom of choice and it is high if it does involve much freedom of choice. Phrasing it differently, information is low when a situation is highly organized and it is high when much randomness is involved. If one receives information about an event in a situation that is random, this information is worth more than if the situation was highly predictable.

3.1.1.2 The Communications Channel

In a communications situation the amount of information that an information source produces depends on the number and probability of possible messages, as discussed in the previous section. If the information source can choose among four equally probable messages (whatever the content of each message is), then the amount of information it produces is two bits. Before a message is sent across the communications channel, the destination in such a communications situation is uncertain about which of the four messages it will receive. When finally a message arrives this uncertainty is reduced. Thus, the reduction of uncertainty was worth 2 bits of information.

Unfortunately, the amount of information that can be transmitted by a communications channel *per unit of time* is limited due to the physical properties of the channel. This limit is called the *channel capacity* and is usually expressed in bits per second. If the amount of information produced by the source exceeds this limit, not all of the information can be transmitted and some uncertainty remains at the destination with respect to which message actually sent from the information source.

3.1.2 Hick-Hyman Law

How these information theoretic principles were used in psychology can best be explained with a model that is quite similar to Fitts' Law: the Hick-Hyman Law of choice reaction. Hick and Hyman were both examining choice situations, particularly how reaction time in such a situation relates to the number of choices available.

3.1.2.1 Hick's and Hyman's Experiments

Hick (1952) carried out an experiment in which subjects were presented up to 10 lights, of which one was lit in a trial. After the light went on, subjects pressed one out of 10 morse keys, one key for each light and finger. The number of possible lights was varied in the experiment and the results showed that mean reaction time RT corresponded to $K \log_2(n+1)$ where n is the number of lights. If $n = 1$ then one can see that $RT = K$. Thus, K can be seen as the time required for a simple reaction or a-reaction, that is, the identification that a stimulus is present *at all* and triggering of the motor processes required to respond (Welford, 1968, p. 61). In contrast to that, b-reactions involve choice and add another component to reaction time that encompasses this choice. Thus, when one regards all possibilities including the possibility that no signal is present at all, Hick's Law can be rewritten as $RT = K \log_2 N$ where N represents the "(...) equivalent total number of equally probable alternatives from which the subject has to choose (...)" (Welford, 1968, p. 62). In Hick's experiment described above, the subject had the choice among $0, 1, 2, \dots, n$ responses and N was thus equal to $n + 1$.

Drawing the analogy to the concepts of information theory presented in the previous section, it seemed that the reaction time was dependent on the amount of information of the situation, which could be measured in bits. But the analogy goes even beyond that logarithmic similarity. Hyman (1953) conducted an experiment in which he manipulated the *probabilities* of the stimuli by increasing the probability that one particular light went on relative to another. He found that reaction time was still determined by the information contained within the situation and that it could be predicted accordingly. Thus, such a situation could be described in terms of a communications situation, with the stimulus as the message that is sent and the human processes for perception and decision as the communications channel. The subject's response could be seen as the message that is received. Further underpinning of this conception is given by results of Hick's Experiment (1952). If subjects were asked to go faster, errors were found to increase, if subjects were required to be more precise, they automatically slowed down their reaction. This points to a general limitation involved in such decisions, similar to the capacity limitations of a communications channel. If the situational demands in terms of information are beyond the information capacity of the channel, less information can be transmitted (errors are made). If they are not, all the information that is produced by the source can be communicated (subjects perform perfectly).

3.1.2.2 Hick-Hyman Law in HCI

The Hick-Hyman Law was also received by researchers in human-computer interaction (although not as well as Fitts' Law) to model and predict simple choice situations. For instance: how fast can a telephone operator react when one out of several light turns on (Card, Moran, & Newell, 1983, p. 74)? Or how quickly can a user decide among a choice of methods that accomplish the same task (Olson & Nilsen, 1987)? Similarly, Landauer and Nachbar (1985) combined the Hick-Hyman-Law with Fitts' Law to predict performance in menu selection tasks in hierarchical menus. Also, the Hick-Hyman Law is part of Card et al.'s "Model-Human Processor", a basic model of human cognitive processes and capacities as the "uncertainty principle" (Card et al., 1983, p. 27).

The general problem with the application of the Hick-Hyman Law is that in most cases, the situation is very complex and the probabilities of the stimuli cannot be determined. On the other hand, if a task is so simple that it can be expressed in terms of the information entailed, there is probably also no need to "(...)engage in the complexity of the information theoretic measures" (Seow, 2005). Presumably because its application is not hindered by such problems, Fitts' Law has gained much more popularity among researchers in human-computer interaction in contrast to the Hick-Hyman Law. The background of Fitts' Law is presented in the next section.

3.1.3 Fitts' Law

As other researchers in his field (e.g. Searle & Taylor, 1948), Fitts' believed that human motor performance is not only determined by physiological factors such as that of the workings of joints and muscles but by a central mechanism responsible for the coordination of muscles and the visual or proprioceptive feedback-loops used for "(...) monitoring the results of the ongoing motor activity" (Fitts, 1954). According to Fitts, this central capacity of the motor system manifests itself in a relationship between the speed of a movement and the distance and movement tolerance. Such a relationship was discovered by Fitts in several experiments on simple activities that involved hand movements to tap on or grasp a target object.

3.1.3.1 Fitts' Experiments

In 1954 Fitts' published several experiments that supported this view. The most prominent of his experiments (Fitts, 1954) involved a reciprocal tapping test.

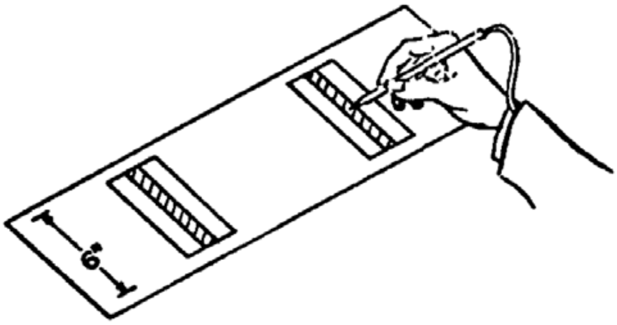


Figure 5: Reciprocal Tapping Apparatus, from (Fitts, 1954)

In this tapping test, participants held a metal stylus to tap on rectangular targets as can be seen in Figure 5. The widths of the targets W as well as the center-to-center distance between the targets, also called the amplitude A, were independent variables and were thus varied by the experimenter. Four different target sizes and four amplitudes were used by Fitts. Therefore, participants had to carry out the task with 16 different combinations in total. The task for the participants was to alternately tap on each target, scoring as many hits as possible in the given amount of time of 15 s while limiting errors (missing the target). An error was recorded when a tap hit one of the two error plates which were attached to both sides of the target plate. The number of taps was determined regardless of errors and an error rate of < 5% was tolerated. The results showed that the number of taps made by participants within 15 s, and therefore the time required for one such tap, was dependant on the amplitude and width of the stimulus. Hitting a narrower target took more time than hitting a wider target. Also, hitting targets that were farther apart took longer than hitting targets that were closer together.

3.1.3.2 Index of Difficulty

More precisely, Fitts found the following relationship among the average movement time MT required for a tap and the amplitude A and width W of targets:

$$MT = a + b \log_2(2A/W)$$

This relationship can be explained on the grounds of information-theoretic concepts. Fitts argued that his experimental setting could be compared to a communications situation: the stimulus that is presented to the participant is a message that contains some information, namely the amplitude of the required movement to hit the targets. Furthermore, in this communications situation, the human perceptual and motor systems function as a communications channel, which transmit the given message. The message that is received through this channel is the movement actually made by the participant. If the information conveyed by this message, that is, the difficulty of the movement requirements in the form of the width and amplitude of the movement, is below the capacity of the channel, the message can be transmitted without any loss of information. If, however, the information is greater than the capacity of the channel, it cannot be sent through it completely. Similarly, if the movement requirements are too harsh, participants either start making errors or slow down their movements to fulfill the movement requirement, that is, they reduce the amount of information transmitted per unit of time.

Consider a situation with one observer. The experimenter and the observer play a little game: The experimenter creates a message by presenting a stimulus, that is, a tapping situation of defined width and amplitude and "sends" it by letting the participant carry out the tapping movements as described before in Fitts' paradigm. Imagine that the observer could only see the movements made by the participant and not the actual stimulus. The task for the observer is now to guess which amplitude the experimenter actually sent. For instance, if the experimenter presented a combination of target width and amplitude as shown in Figure 6, the movements made by the participant should enable the observer to identify which of the seven alternative amplitudes was presented.

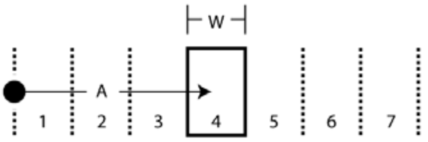


Figure 6: Information within a Tapping Situation

In other words, the uncertainty of the observer of which amplitude is meant is reduced by the movements made by the participant. Thus, similar to how the number and probability of choices determine the quantity of information in situations that follow the Hick-Hyman Law, the amount of information in such a tapping situation depends on the number of possible "movement outcomes" and can be captured by the logarithmic term of the movement time relationship shown before.

$$ID = \log_2(2A/W)$$

Fitts called this logarithmic term the index of difficulty ID since it seemed to be a major determinant of the difficulty of a movement. Consider Figure 7 which illustrates different amplitude and width combinations. In situation S1 the amplitude is 3.5 times the size of the width of the target. One can calculate the difficulty (or information) of this situation by $\mathrm{ID}_{\mathrm{S1}} = \log_2(2 \times 3.5/1) \approx 2.8$ bits. In situation S2 the amplitude is the same as in the first situation but this time the width of the target is smaller, with the amplitude being 7.5 times the size of the width of the target. Thus, more precision is required and the task is harder. This is reflected by a greater index of difficulty $\mathrm{ID}_{\mathrm{S2}} = \log_2(2 \times 7.5/1) \approx 4$ bits. Finally, in situation S3, the target is larger again, but now the amplitude is greatly increased. Due to the larger amplitude, this task is harder than that of situation S1 although it has the same accuracy requirement. Again the size of the amplitude is 7.5 times the size of the target which results in an index of difficulty of $\mathrm{ID}_{\mathrm{S3}} = \log_2(2 \times 7.5/1) \approx 4$ bits as in S2.

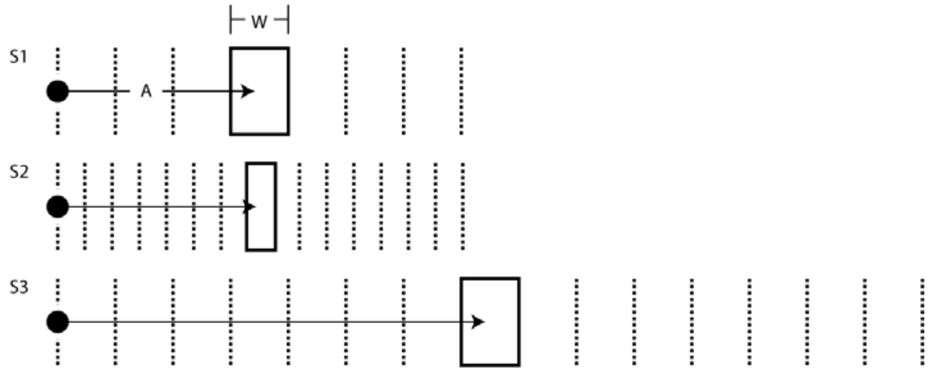


Figure 7: Visualization of Different Movement Requirements

If it takes a participant one second to perform the movement task of situation S2 (4 bits), the amount of information transmitted in that one second is exactly equal to the amount of information contained in the situation, that is, roughly four bits. If it takes two seconds to perform the task in the same situation, the information transmitted *per second* is not four bits, but only two. Assume that a participant manages to complete task S2 within one second. How fast will he be in task S3? The amplitude is almost twice as large as that of S2, so intuitively this task should take longer. However, the target width of S3 is considerably larger than that of S2 and the participant can thus speed up his movement without fear of making an error and accomplishes the task within one second. Similarly, four bit of information were transmitted per second.

What has just been illustrated is known as the *speed-accuracy trade-off* which is implied in Fitts' Law (Schmidt & Lee, 2005, p. 212). Participants can "trade-off" movement time against accuracy. They can either perform a fast movement, possibly missing the target and reducing the information actually communicated – the independent observer must infer the amplitude (the message) from the movement made by the participant – or they can carry out a slow movement, hitting the target accurately, preserving the information contained in the message. In any case, the margin of such a trade-off is limited by a central capacity which is discussed next.

3.1.3.3 Index of Performance

If one plots the data obtained from an experiment such as that of Fitts', one can see that movement time is linearly related to the index of difficulty (see Figure 8).

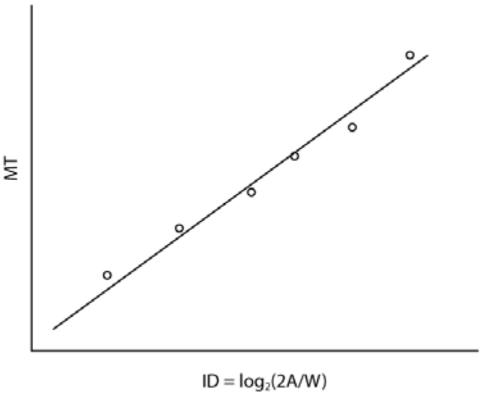


Figure 8: Common ID-MT Relationship

As was shown before, this relationship is represented by the equation $MT = a + b \log_2(2A/W)$ or $MT = a + b \times ID$ and the empirical constants a and b can be determined empirically through regression analysis of the movement data collected during the experiment. For instance, in one of Fitts' tapping experiments a was 12.8 ms and b 94.7 ms (MacKenzie, 1992). But what interpretation can be given to these empirical constants? The constant b is of course the slope of the regression line and a is the y-axis intercept. The constant b can best be understood as the movement time required for one bit. It is thus measured in seconds per bit and can also be understood as the "(...) sensitivity of the effector [i.e. the hand, foot, etc.] to changes in the ID" (Schmidt & Lee, 2005, p. 211). This means that tapping tasks with different limbs or muscle groups yield different slopes. Moreover, if one considers the reciprocal of b, one has a measure that fits neatly into the framework of information theory: the channel capacity or maximum amount of information that can be transmitted per unit of time by the motor system with the muscle or limb that was under scrutiny in the experiment. This capacity, which is measured in bits per second, is known as the index of performance IP which is also sometimes called the throughput.

$$IP = 1/b$$

For instance, in the experiment of Fitts' the slope was 94.7 ms, as described above. This means that the index of performance or throughput was ≈ 10.6 bits/s (1/0.0947 s). Unfortunately, the interpretation of the y-axis intercept a is less clear. Soukoreff and MacKenzie (2004) argue that one should not "(...) infer too much about the intercept" because it is an empirical constant obtained through regression analysis. An error commonly made with data obtained from a regression analysis is that predictions are made beyond the range of original values of the explanatory variable (on the x-axis). Such *extrapolation* often leads to invalid predictions. According to Soukoreff and MacKenzie, this is also the case with extreme ID values which have not been tested during the experiment. Logically, an intercept of zero should be obtained: If there is no information, transmitting this "non-information" should not cost any time.

3.1.3.4 Applicability of Fitts' Law

The basic relationship between the movement time and the amplitude and width of a movement modeled by Fitts' Law was found to hold for a variety of movements and situations. For instance, hand movements, tapping of the feet, wrist rotation, movements made using a microscope, movements underwater, or head movements (MacKenzie, 1991). Additionally, Fitts' Law was also found to hold for pointing movements made with continuous computer pointing devices and different models and indices of performance were obtained for different devices. For instance in a study by Card et al. (1978) models were obtained for a mouse and an isometric joystick[3]:

$$MT_{Mouse} = 1.03 + 0.096 \times ID$$

$$MT_{Joystick} = 0.99 + 0.220 \times ID$$

The throughput value of both devices differed accordingly: That of the mouse was 10.4 bits/s and that of the joystick was 4.6 bits/s. This indicates that different channel capacities can be obtained for different pointing devices. For this reason, the same pointing task with one pointing device may require more time as with another pointing device. Likewise, given that the pointing task must be carried out within a predefined amount of time, pointing may be more accurate with one device compared to another device. These empirically determined models and indices of performance serve as the basis for pointing time predictions for the design of user interfaces and for the comparison of pointing devices. An overview of these applications is given in the next section.

3.2 Uses of Fitts' Law in HCI

The use of Fitts' Law in human-computer interaction is threefold: modeling, thinking, and testing: On the one hand Fitts' models serve as engineering models that can be used to predict how certain design decision might influence the effectiveness and efficiency of a software system in terms of selection or pointing times or to assess an existing system design without actually conducting a costly usability test. The Keystroke-Level Model (Card et al., 1983) incorporates pointing as one of several low-level physical-motor operations that are used to predict and compare execution times that are theoretically attainable with a given system or two versions of that system. A short overview of the modeling aspect is given in section 3.2.1.

On the other hand, thinking along the lines of Fitts' Law and its implications has stimulated the conception of new interaction techniques. These techniques mostly seek to improve selection performance by optimizing the determinants of throughput, the amplitude of the movement and size of the target. A brief overview is given about the interaction concerns in 3.2.2.

Finally, it is a tool for the assessment of pointing devices, that is, the "raw" pointing performance that is attainable with a device. For this purpose annex B of the ISO standard 9241-9 provides information about standardized tests and procedures. A separate section (3.3) is devoted to this topic because of its importance for this thesis.

---

[3] The values cannot be compared to the results of Fitts because a slightly different formula for ID was used in the regression analysis.

3.2.1 Testing Graphical Interfaces

In their comparative evaluation of the mouse, rate- controlled isometric joystick, step keys, and text keys for selection of text, Card and colleagues showed that Fitts' Law could be used to model pointing device performance (Card et al., 1978). This discovery laid the foundation for the use of the law in models for systems design. The classical example for such a model is the Keystroke-Level Model (KLM). KLM can be used to model the execution of tasks that consist of physical-motor operations: keystroking (K), pointing (P), homing (H), drawing (D), and mental preparation (M). For each of these operations a time estimate was obtained empirically (see Table 3). In essence, the total execution time of a task comprises the sum of execution times of these operations. Suppose that when designing a new interface for a time critical application, two methods exist for triggering a function that is quite important and used very frequently during work with the interface. Method one requires the use of the mouse to select an item from a form field drop-down menu. The time required for this method can be calculated according to KLM using the values from Table 3:

$$T_{M1} = P(\text{menu}) + M + P(\text{item}) + B(\text{click}) = 1.1 + 1.35 + 1.1 + 0.2 = 3.75 \text{ s}$$

In a second method, the user jumps to the form field by hitting tab repeatedly and then typing the first letter of the item followed by the return key to select it.

$$T_{M2} = M + K(\text{tab}) + K(\text{tab}) + K(\text{tab}) + K(\text{R}) + K(\text{RETURN}) = 1.35 + 5 \times 0.12 = 1.95 \text{ s}$$

The two methods can now be compared quantitatively, for instance when estimating working time to redesign the system, possibly placing the function not in a menu, but directly on the main interface.

Table 3: KLM Operators, from (Card et al., 1983)

Operator		Time (s)
K	Press key or button	
	Best typist (135 wpm)	0.08
	Good typist (90 wpm)	0.12
	etc.	
P	Point with mouse	1.10
B	Pressing a mouse Button	
	Press (down or up)	0.1
	Click (down and up)	0.2
H	Home hands on keyboard	0.40
M	Mentally prepare	1.35

In order to create a model of "practical use" (Card et al., 1983, p. 297) the authors only used estimates for each operator to keep it as simple as possible but "tuned" the model to their needs. When devising KLM, Card and colleagues were interested in predicting text-editing performance and, undoubtedly, having an accurate typing model is conducive in this case. Therefore, the authors proposed not just a single estimate of typing speed in the original KLM version but a more sophisticated estimate that distinguished between several classes of typists (see Table 3).

Likewise, a Fitts' model could be used if it is particularly important to obtain accurate pointing predictions. Of course this would also complicate the application of KLM because obtaining movement time predictions from a Fitts' model requires that the difficulty of a pointing movement and thus the amplitude of the movement and the width of the target are known. This would require the measurement of the amplitude and width for each pointing movement and would of course be much more complicated than the simple addition of time estimates as in the KLM example presented above.

A model incorporating both laws, Fitts' Law and the Hick-Hyman Law, for the evaluation of soft keyboards for PDAs or similar mobile devices was proposed by MacKenzie and colleagues (Mackenzie, Zhang, & Soukoreff, 1999). The goal of the model was to predict and compare typing speeds of various keyboard layouts without actually carrying out an empirical evaluation of these layouts. A Fitts' Law model was used to predict the movement time required to tap on a soft key using a stylus. The Hick-Hyman Law served as a basis for estimating novice reaction time required for visual search. In the combination with a linguistic frequency estimate of common English letter pairs, a model was constructed to estimate novice and expert text entry speed. Upon comparison of several keyboard layouts with this model it was revealed that, theoretically, layouts could be optimized so that higher typing rates could be obtained, for instance, by adding two space bars instead of one or rearranging keys so that common letters are in close proximity.

3.2.2 Designing Interaction Techniques

Besides evaluating existing systems or judging design alternatives, some researchers even strive to incorporate the implications of Fitts' Law into the design at the level of interaction techniques. One simple rule of thumb to learn from pointing research, which is predicted by Fitts' Law, is that large targets are easier to hit than small targets. This design rule of course collides with the need to retain screen real estate, the amount of screen space that can be utilized to display information. In this section, some examples are given of systems that try to solve this issue.

3.2.2.1 Increasing Target Size (W)

Thinking along the lines of Fitts' Law McGuffin and Balakrishnan (2002) invented expanding targets. In their experiment, the authors found that if targets such as buttons or icons were resized just prior to their selection, the pointing performance of selection was significantly higher. However, the test was carried out only with a single target. The general question that arises is: how should the system know which target to expand if several targets are present? If this information was available before pointing, pointing itself would be wholly unnecessary. Since computers cannot yet read users' minds, researchers have to invent good heuristics to predict the next target. Of course these predictions are far from faultless and thus, the final act of turning pointing into a selection must remain a manual task. Nevertheless, the information gained through prediction can be used to facilitate this manual task. In this respect an interesting approach was presented by Zhai and colleagues (2003) who replicated the experiment from McGuffin and Balakrishnan and additionally proposed a technique that made it possible to use target expansion with multiple targets by predicting the pointing target. This prediction occurred by calculating the intersection of the likeliest path of the pointer derived from the current movement vector with the set of target elements (see Figure 9).

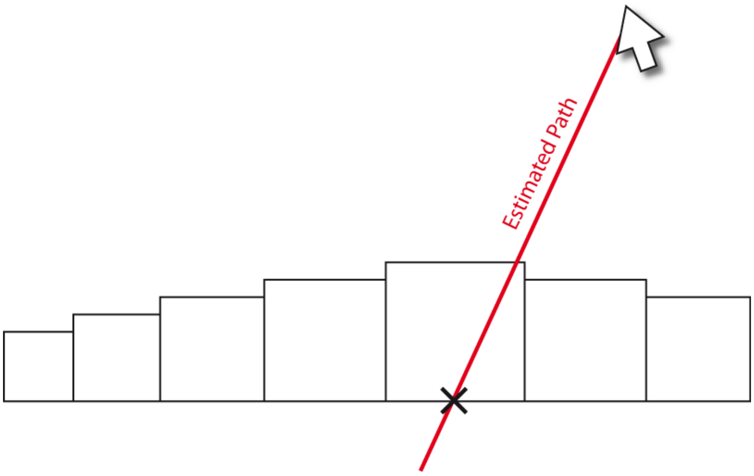


Figure 9: Illustration of Target Expansion, adapted from (Zhai et al., 2003)

Another recent application of target prediction to facilitate pointing is the on-screen keyboard of the new Apple iPhone. Similar to techniques such as T9 that facilitate typing of SMS using a dictionary of common words, the soft-keys of the iPhone keyboard are expanded depending on which letters seem reasonable based on the previously entered letters[4]. For instance, if one types the letters "TIM" the software uses the built in dictionary to learn that the only letter that could follow is "E" and expands the "target zone" of this letter. Innovative as this approach may seem, it remains unclear if the expansion used in the iPhone really improves pointing performance, at least in the sense of Fitts' law, since the keys are not expanded visually but only invisibly in "motor space".

3.2.2.2 Decreasing Amplitude (A)

An even more radical interaction technique called object pointing is proposed by Guiard et al. (2004). The authors analyzed the nature of pointing as it is being used in today's graphical user interfaces and came to the conclusion that much of the information conveyed through pointing is ignored by the system. For instance, if the user's task is to select one out of 40 items on the desktop then the amount of information that has to be communicated is $\log_2 40 \approx 5.3$ bits. Suppose that each icon has a size of 20 × 30 pixels and that the screen resolution is 1600 × 1200. The actual amount of information communicated to the system by pointing to an item is $\log_2(1600 \times 1200/600) \approx 11.6$ bits – much more information than actually required (example taken from Guiard et al., 2004). In order to do away with such information "extravagance", Guiard et al. proposed a timorous cursor (Tim) which entirely skips empty spaces between objects and instead jumps from object to object and thus completely removes the pointing amplitude. The Tim cursor determines the next object by estimating the movement path, similarly to

---

[4] A video from Apple can be obtained from www.apple.com/iphone/gettingstarted/keyboard.html that shows this function

the technique of Zhai et al. (2003) described earlier and chooses the closest object in an angular sector along that path. Instead of resizing the target that was determined by that prediction to facilitate pointing, the pointer simply jumps directly over to the object. Of course, similar to the problems of target prediction and expansion, object pointing is problematic when many targets are present because then the likeliness of a failure in prediction increases and a jump is made to the wrong target.

3.3 Testing Pointing Performance with ISO 9241-9

Apart from the uses of Fitts' Law for evaluation of user interfaces and for devising interaction techniques, the main application area of the law in human-computer interaction is testing of pointing devices. As presented earlier in section 3.1.3.4, different Fitts' Law models and indices of performance could be obtained for different devices. It seems that pointing is generally difficult and thus slower with some devices and easier and therefore faster with others and that this difference could be expressed in terms of the index of performance. With the publication of annex B of the ISO standard 9241-9 the procedure for obtaining such a performance index has been standardized on the foundation of Fitts' Law and the Fitts' tapping paradigm (ISO 9241-9:2000).

3.3.1 Overview of ISO 9241-9 Annex B

As pointed out previously in section 2.3, the usability of a pointing device is defined as the extent to which it "can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" (ISO 9241-9:2000, , p. 11). Different use contexts and goals also involve different *tasks* that impose specific demands on a device. Inversely, the properties of a device determine its appropriateness for a specific task. Similar to the classification of interaction tasks by Foley and colleagues (1997) presented in section 2.1.2, five task primitives, are proposed by the standard: pointing, selecting, dragging, free-hand input, and tracing. In annex B several tests for the assessment of efficiency and effectiveness of input devices are described which are based on these primitives.

Table 4: Overview of Tests in ISO 9241-9 Annex B

Test	Description	Task Primitive
Onedirectional tapping	Alternating selection of rectangular target objects.	Pointing, Selection
Multidirectional tapping	Selection of circular target objects.	Pointing, Selection
Path-following	Movement through a tunnel without touching its boundary.	Tracing, Dragging
Tracing	Movement through a tunnel of circular shape without touching its boundary.	Tracing, Dragging
Free-hand input	Writing of symbols (e.g. alphanumeric).	Free-hand input
Grasp-and-park	Pointing task, followed by input on the keyboard.	–

The tasks most often used in pointing device studies are the one- and multidirectional tapping task (Douglas & Mithal, 1997). The onedirectional task corresponds to Fitts' original tapping task. The multidirectional task is similar to this task but was modified so that pointing performance could be measured in two dimensions (see Figure 10). Besides the onedirectional and multidirectional tapping paradigms, the ISO standard also proposes other testing paradigms that model other tasks than mere pointing and selection (see Table 4). These are for instance tasks for steering a pointer through a tunnel. However, these tasks have received considerably less attention than the Fitts' paradigms. Presumably, because the task primitives that are tested with those kinds of tasks are much less important in today's WIMP environments (Window, Icon, Menu, Pointer, see Jacob, Deligiannidis, & Morrison, 1999).

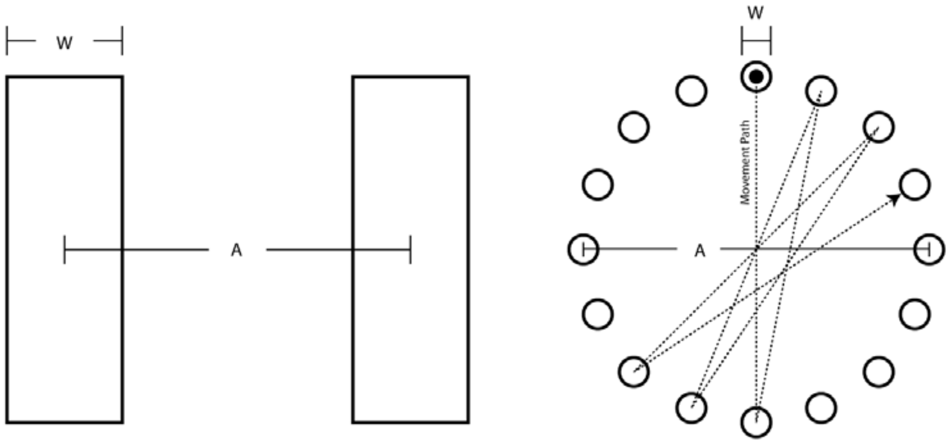


Figure 10: Onedirectional (left) and multidirectional (right) tapping task

The methods of the ISO standard for testing effectiveness and efficiency of pointing devices are based on the calculation of a performance index for a specific device. This index can be used to quantify how much information per unit of time can be transmitted in a movement situation with the tested device. It is therefore a direct measure of pointing effectiveness, the "accuracy and completeness with which users achieve specified goals" and efficiency, the "resources [e.g. time] expended in relation to the accuracy and completeness with which users achieve goals" as defined by the ISO standard (ISO 9241-9:2000, , p. 11).

Compared to Fitts' original experiments the methods and procedures in the ISO standard were slightly adapted. For instance, since computer and measurement equipment is now available that was not available at the time Fitts carried out his experiments, today, the instructions given to the participants differ. Participants are not given a fixed amount of time to score as many alternating hits as possible but a fixed number of hits is prescribed and participants are told to work as fast and accurately as possible (Douglas, Kirkpatrick, & MacKenzie, 1999). Also, per-trial movement times are not inferred from the mean of several trials but are measured directly for each trial. For instance, in the onedirectional tapping task, movement time is measured from the click on the first target until the click on the second target.

3.3.2 Test Design and Procedure

Commonly, the performance of two or more devices is compared against each other. Because of resource constraints and added benefits in statistical inference (Hussy & Jain, 2002), such experiments typically use a within-subjects design. This means that, for example, if two devices A and B are under scrutiny in the test, each participant carries out the tasks with both devices successively. On the other hand, in a between-subjects study, a participant is only presented one device and differences between performance indices are then compared between groups. Of the studies reviewed for this thesis, only one study used a between subjects design with 12 participants in each group (Douglas & Mithal, 1997).

In a within-subjects design, the sequence of device presentations has to be balanced since prolonged work on a task with one device could influence the performance on the task with the other device, due to learning or fatigue. Usually this balancing is made between subjects. For instance, if 12 participants take part in a study, two groups can be formed according to the sequence of device presentation: G1 who work with device A and then B, and G2 who first work with B and then A. For each device, participants perform a vast number of tapping trials with different combinations of A and W. Amplitude and width combinations are chosen so that easy, medium, and large difficulties are covered by the test. According to the standard, ID levels should span the range from one (easy) to six (hard) bits. Also, in order to test whether Fitts' Law is actually an adequate model for the device tested, one usually carries out a regression analysis (see section 3.1.3.3). This requires that a reasonable[5] amount of ID values are tested in the experiment to get meaningful results from the regression analysis.

At the end of the test, a questionnaire is often employed to assess the satisfaction component of the usability requirement. In annex C of the standard, two questionnaires are presented that can be used for a comparative evaluation or for independent evaluation. Items tested by both questionnaires include, for instance, the actuation force or operation speed and fatigue ratings for the finger, wrist, arm, shoulder, and neck (ISO 9241-9:2000, , p. 39).

3.3.3 Calculation of ID and IP

Several changes to the calculation of ID were proposed and used by researchers who argued that these changes would lead to a better fit between the experimental data and the model than Fitts' original model. These formulas can be seen in Table 5.

Table 5: ID Formulas

	Calculation of ID
Shannon	$\log_2\big((A/W)+1\big) = \log_2((A+W)/W)$
Welford	$\log_2(A/W + 0.5)$
Crossman	$\log_2(A/W)$
Fitts	$\log_2(2A/W)$

---

[5] A typical combination used by researchers in the field is 3A × 3W = 9 ID values (e.g. Douglas et al., 1999).

For the purpose of device comparisons the actual formula is irrelevant. It is merely important that the same formula is used within an experiment and if one wishes to make cross study comparisons. As is noted by Soukoreff and MacKenzie (2004, p. 780), differences in calculation methods between studies result in completely different performance values for similar devices and tasks. This issue can be avoided by using the formula proposed by the ISO standard, the Shannon formulation:

$$ID = \log_2\left({}^{A}\!/_{W} + 1\right)$$

In section 3.1.3.3 it was explained that the index of performance could be obtained by taking the reciprocal of the empirical constant b of the linear model $MT = a + b\log_2(2A/W)$. Typically, the parameters a and b can be obtained via a regression analysis of the movement time results. For the purpose of building an engineering model this method is accepted (Soukoreff & MacKenzie, 2004). For the purpose of input device evaluation a different method is proposed by the ISO standard[6]:

$$IP = ID/MT$$

Where MT denotes the movement time for a particular trial. Although IP could be calculated separately for each trial, this is not done so due to reasons explained in the next section. The usual procedure is to calculate IP for each amplitude and width condition for each participant and device. The mean IP values obtained for each participant with a specific device represents the total index of performance for that particular device. For instance, if three amplitudes and three widths are used in an experiment this results in nine difficulty conditions – if each amplitude is combined with each width. During the experiment each participant carries out several taps (e.g. 25 taps) for each of these nine difficulty conditions with a specific device. The index of performance for this device and this participant is then calculated as:

$$\frac{\sum_{i=1}^{n} ID_i/MT_i}{n}$$

Where n denotes the number of difficulty conditions (e.g. nine in this case), ID_i denotes the difficulty index of a particular difficulty condition, and MT_i denotes the mean movement time for the 25 taps carried out for that condition. The index of performance for the device is then calculated from the mean index of performance value of all participants.

3.3.4 Effective Width

One of the major differences of the ISO method compared to Fitts' original experiments is how the width of the targets is incorporated into the calculation of the index of performance. The width W of the targets in a onedirectional tapping paradigm is usually specified a priori, that is, it is an independent variable manipulated by the experimenter. However, participants do not necessarily distribute their movement endpoints so that they fully utilize the given target width. For instance, if two very large targets with small amplitudes are presented, most of the hits will probably fall within the right half of the left target and within the left half of the right target.

---

[6] Although this method is criticized, for instance by Zhai (2004)

This means that the participant actually used less of the available target width. Also, if targets are very small, participants often miss the targets, distributing movement endpoints over a larger width than the one given as is shown in Figure 11 (S2).

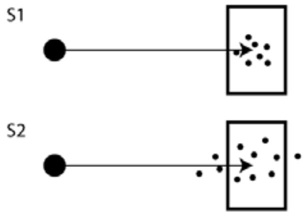


Figure 11: Different movement endpoint scatter: the user was much more precise in S1 than in S2

The consequence of these phenomena is that calculating the index of performance using a priori target widths does not necessarily reflect the true accuracy that can be achieved with a device. This was also recognized by researchers in the field and it was suggested to use the *effective width* instead, that is, the a posteriori width of a target according to the actual distribution of movement endpoints (Welford, 1968, p. 147). Calculation of effective width is also proposed by the ISO standard:

$$W_e = 4.133 \times \sigma_x$$

Where σ_x is the standard deviation of the movement endpoints measured "in the direction where movement proceeds" (ISO 9241-9:2000, p.29). The product of the z-score 4.133 and the movement endpoint standard deviation represents the size of a target which would have covered 96% of all movement endpoints under the assumption that such endpoint scatter is distributed normally. A z-score is simply a standardized observation of a distribution with $z = (x - \mu)/\sigma$ where x is the observation, μ is the mean of the distribution, and σ is the standard deviation. If data is normally distributed, around 68% of observations fall within a range of $\mu \pm \sigma$, as can be seen in Figure 12.

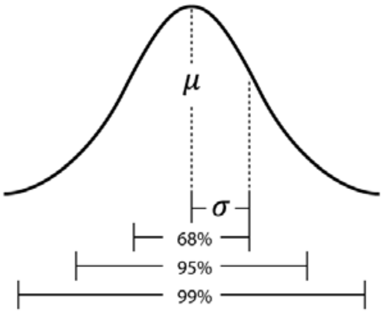


Figure 12: Normal distribution and standard deviation

Similarly, 95% of all observations fall within $\mu \pm 2\sigma$ and 99.7% fall within $\mu \pm 3\sigma$. This is true for *any* kind of normally distributed data, no matter what the actual values of the mean or standard deviation are. Thus, a z-score expresses an observation in terms of the distance to the mean measured in standard deviations. For instance, if $x = \mu$ then the associated z-score is

$z = (\mu - \mu)/\sigma = 0$. If $x = \mu + \sigma$, that is, x lies exactly one standard deviation to the right of the mean, then the associated z-score is $z = \big((\mu + \sigma) - \mu\big)/\sigma = \sigma/\sigma = 1$. Similarly, if $x = \mu - \sigma$ then $z = -1$.

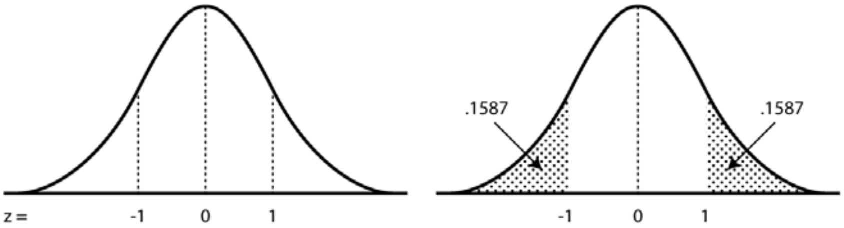


Figure 13: a z-score of -1 means that around 15% of observations fall below that score

For every z-score the number of observations that fall below it can be obtained from standard statistical tables. For instance, for a z-score of -1, such a table gives the information that 0.1587 or 15.87% of all observations fall below that score. For a score of one, 0.8413 observations fall below that score. Thus, $1 - 0.8413 = 0.1587$ fall above it (see Figure 13). Roughly 68% of observations fall into the area defined by those two scores ($1 - 2 \times 0.1587$).

For $z = 2.0665$, around $1 - 0.98 = 0.02$ or 2% of all observations fall above that score and roughly 96% fall between -2.0665 and 2.0665. Now the reasoning behind the construction of W_e becomes clear. If, for instance, a standard deviation of 10 pixels is obtained from an experimental measurement, the question is: how large must a target be that 96% of movements end within that target. The a priori target size W was, for instance, 30 pixels. From the characteristics of a normal distribution, as described before, it follows that the z-score of an observation that is located one standard deviation to the right of the mean is exactly one. This means that it is possible to scale the obtained standard deviation by 2.0665 to obtain the width of the target that would have contained 96% of all movement endpoints. In the example: $10\text{px} \times 2.0665 \cong 20\text{px}$. As can be seen from Figure 14, this calculation takes into account only one half of the target. Thus, to calculate the total size it must be doubled, or the standard deviation must be multiplied by $2 \times 2.0665 = 4.133$. One can see that the effective width is significantly larger (40 px) than the original width (30 px), reflecting the actual distribution of the movement endpoints.

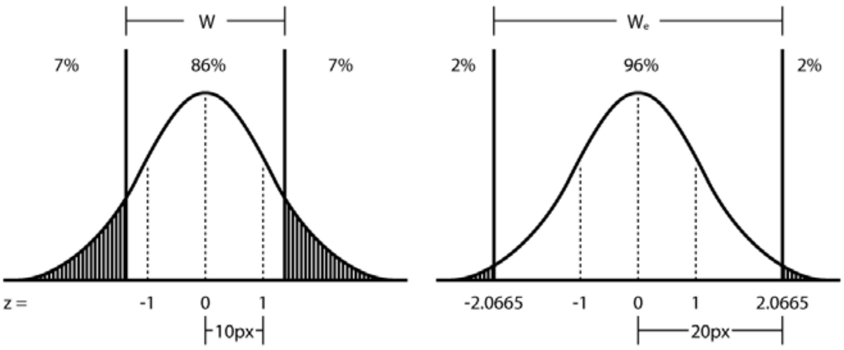


Figure 14: Actual endpoint distribution and a priori target width: around 14% of observations fall outside the target (left). Adjusted width of the target (right)

Of course, a larger effective width also affects the difficulty or information contained within the situation, which is then calculated as:

$$ID_e = \log_2 \left({}^{A}/_{W_e} + 1 \right)$$

If a participant moves quickly but inaccurately, missing the target very often, this lack of accuracy is reflected in an increased effective width which also goes into the calculation of the *effective* index of performance.

$$IP_e = ID_e / MT$$

On the other hand, if a participant is even more accurate than required by the situation, this reduces the effective width and increases the effective index of performance. In summary, calculating the effective width is important to compensate for differing bias toward accuracy when conducting an experiment[7].The effective index of performance is thus a more accurate indicator of the pointing speed and accuracy achievable with a pointing device than the index of performance without that adjustment. This adjustment for accuracy is also the reason why the (effective) index of performance cannot be calculated for a single trial. The calculation requires the measurement of the standard deviation over a series of trials. Thus, as noted earlier, the index of performance is calculated for each amplitude and width condition for each participant and device, and the standard deviation is taken from the movement endpoints of the trials that were carried out in that condition.

3.4 Pointing Test Standardization Issues

It is clear that results cannot be compared between two studies on the basis of movement times or error rates, unless experimental conditions, in particular target sizes and amplitudes are kept the same. Of course, this would be a requirement too restricting for most experimental designs. However, following Fitts' Law, it is possible to calculate the throughput of a device, a measure that is largely independent of the *actual* conditions used in an experiment. With such a measure

---

[7] Further information about the reasons and effects of the accuracy adjustment can be found in the literature (Welford, 1968; Zhai, Kong, & Ren, 2004).

at hand, it may seem possible to compare results across studies, building a common body of knowledge for driving design and research endeavors and for collecting pointing time models that can then be used for task time predictions (e.g. with KLM, see section 3.2.1).

In their publication “towards a standard for pointing device evaluation (...)”, Soukoreff and MacKenzie emphasized the importance of conformance of experimental procedures and calculations to the ISO standard, to “(...) improve the comparability and consistency of forthcoming publications” (Soukoreff & MacKenzie, 1995, p. 1). Indeed, their comparison of several studies has shown that IP results are much more consistent among the studies that followed the procedures of the ISO 9241-9. For instance, in non-compliant studies, IP values for the mouse were very inconsistent, ranging from 2.5 to 12.5 bits/s because of different calculations of ID. Also discrepancies emerged in studies that used the same methods of calculation but differed in other aspects, such as the experimental procedures. Soukoreff and MacKenzie came to the conclusion that it is impossible to get “(...) a clear and accurate answer to the question, what is the throughput of the mouse?” (Soukoreff & MacKenzie, 1995, p. 780). The IP values from ISO compliant studies, however, seem to be much more consistent as can be seen in Table 6.

Table 6: ISO IP-Values, from (Soukoreff & MacKenzie, 1995)

Device	IP (bits/s)	Studies
Isometric Joystick	1.6-2.55	4
Touchpad	0.99-2.9	4
Mouse	3.7-4.9	5

Nevertheless, several inconsistencies in testing procedures and calculation methods remain, that make comparison across multiple studies delicate. These are presented in the following.

3.4.1 Discrete vs. Continuous Tasks

For the onedirectional and multidirectional tapping task, two versions exist that have both been used widely in studies. Essentially, in the continuous variant, participants are tapping on the presented targets without pauses in between trials. This variant corresponds to Fitts’ original task (Fitts, 1954). On the other hand, researchers also used a discrete version of the task which corresponds to the later paradigm of Fitts and Peterson (1964). In this variant participants have to move the pointer back to a home position before the next trial begins. In comparing the results of discrete and continuous tasks it was found that the continuous tasks produced lower IP values than the discrete variant (Keele, 1968). The reasons for this are not clear, though. MacKenzie (1991) speculates that one reason could be that pointing movements can be prepared more thoroughly in discrete tasks.

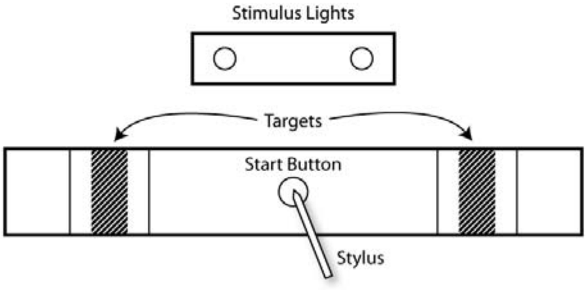


Figure 15: Discrete Task, adapted from (Fitts & Peterson, 1964)

In the experiment of Fitts and Peterson (1964) the apparatus depicted in Figure 15 was used. Subjects held the stylus in one hand, with the stylus tip touching the “start button”, a small circular plate in the middle of the device. When the stylus was in contact with this button, a small electrical current passed through the circuitry of the apparatus and body of the participant. The onset of a stimulus light was preceded by an acoustical warning. Reaction time was measured from the onset of the stimulus light to the moment the tip of the stylus left the start button. Movement time MT could then be derived as the total time for the acquisition of one target minus reaction time. In contrast to this, reaction time does not play a role in a continuous task, where movement time is taken as the time between two taps.

The discrete paradigm was also used by researchers interested in evaluating pointing devices and in fact, most of the experiments were carried out with discrete and not continuous tasks (Douglas & Mithal, 1997). Thus it is rather surprising that only the continuous variants are proposed in the ISO standard. The question is which of the two variants is better suited to capture the character of the pointing movements commonly performed: the continuous or discrete paradigm? Douglas (1997) argues that it is hard to classify real computer pointing tasks according to one or the other, but that most pointing tasks are made up of a series of trials, for instance when pulling down a menu and then selecting a menu item. However, this argument only superficially legitimates the use of the continuous task since the “nature” of such tasks could very well be discrete. In the continuous tasks, the movement of pointing goes back and forth between two targets almost hypnotically, and participants sometimes even fail to stop tapping once the testing software switches conditions, as if they were playing back an ingrained program. On the other hand, in the situation of selecting a menu item from a pull down menu, first the pull down menu has to be located visually. After pointing and selection of the menu, the menu items unfold and the participant pauses to locate the right item before a pointing movement is made toward it. A situation that is totally different from the continuous test, since pointing, in essence, is discontinuous, even though the interruptions between two pointing moves are only small. Further analysis of common pointing movements made for instance during work in an office setting would certainly be of interest with respect to this discussion. Such an analysis could include a prediction phase, where task times of real office tasks are estimated with models obtained from both tests (discrete/continuous) and an evaluation phase, in which participants actually carry out those tasks. Also, insight could be gained from analyzing the movement paths of movements made during a regular office task, measuring the ratio between continuously “chained” selections and discrete selections with stationary phases without any movement of the pointer in between.

> *Research Question:* What is the true nature of common pointing movements and which test is suited best for these kinds of movements?

The problem with discrete tasks is of course the lack of a standardized procedure and the subtle differences that may influence results. For instance, Silfverberg and colleagues tested isometric joysticks with a discrete multidirectional paradigm (Silfverberg, MacKenzie, & Kauppinen, 2001). A “home square” was shown in the middle of the screen and participants had to click on that square to start a trial. Once the trial was started, a target circle appeared and participants ended the trial by clicking on that circle[8]. The problem with this variant is that movement time and reaction time cannot be distinguished since the target appears after the user clicks on the home square possibly inflating the time measured for the calculation of IP. A similar paradigm was used by Douglas et al. who also used a clickable home square (Douglas et al., 1999). However, no information is given in their publication on when the target stimulus was presented and how exactly movement time was measured. Yet another variant was used by MacKenzie and Buxton (1992). In this experiment it was ingeniously secured that reaction time or time taken for visual search could not confound movement time. A small circle was presented in the middle of the screen and a target appeared in a controlled angle and distance from that circle. Participants were instructed to move the pointer to that circle. After the pointer was inside the circle, participants waited for the appearance of a rectangular shape above the circle which signaled that the pointer could now be moved out of the circle in order to click on the target. If a movement outside the circle was detected prior to the appearance of the rectangular signal, the trial was canceled. Movement time could thus be measured from the moment the pointer left the circle until the acquisition of the target. In an experiment to evaluate gaze pointing, Zhang and MacKenzie (2007) used a discrete multidirectional variant with a home square. Once participants moved the pointer onto the home square, it disappeared after some time. Thereafter, movement time was recorded once the pointer started moving. To exclude visual search time, the target was highlighted *before* each trial.

As can be seen from this short review, many variants of the discrete task exist that differ only slightly. However, these slight differences may render comparison of results difficult if not impossible. Unfortunately, since the discrete test is not included in the set of ISO tests, it is unlikely that researchers will use a common variant for future tests.

3.4.2 The Multidirectional Task

No matter whether a discrete or continuous variant is used for the multidirectional task, problems seem to arise when calculating results with this kind of task. As was shown in previous sections, when comparing input devices on the basis of the index of performance obtained, it is

---

[8] A version of a similar test can be seen at http://www.tele-actor.net/cgi-bin/fitts/applet2.pl

important to make the accuracy adjustment by obtaining the effective width $W_e = 4.133\sigma_x$. According to the ISO standard, the standard deviation of the movement endpoints σ_x has to be obtained by measuring "in the direction where movement proceeds" (ISO 9241-9:2000, , p. 29). This requirement is unproblematic when using a onedirectional task. It was, however, interpreted differently by researchers for the multidirectional task. For instance, Douglas et al. (1999) and also Isokoski et al. (2007) calculated σ_x from the squared Euclidean distances between the selection point and the mean point: $d = \sqrt{(x-\bar{x})^2 + (y-\bar{y})^2}$, whereas others reported that distance was measured "(...) along the axis of approach to the target" (J. Y. Oh & Stuerzlinger, 2002; Silfverberg et al., 2001, p. 120), but unfortunately, gave no exact method of calculation. Reconsidering the requirement of the ISO standard and the way σ_x is calculated in the onedirectional case, the best way to measure the movement variation is probably as follows:

Consider the situation shown in Figure 16. The axis of approach vector is denoted as a. This is a vector in the direction where movement proceeds and it can easily be obtained by subtracting the location of the movement origin from the coordinates of the target in the display coordinate system. In case of a discrete multidirectional task, this is, for instance, the home position in the middle of the screen. The movement endpoint vector e, which can be obtained in a similar fashion, has a length equal to the Euclidean distance as calculated by Douglas et al., given that the mean is the target center. Vector e represents the location of the movement endpoint relative to the target. In case of Figure 16, the participant missed the target, tapping slightly to the left. But the length of e cannot be used for the calculation of W_e because it takes into consideration more than the deviation "along the axis of approach". The correct length would be that of the vector o, the projection of e on the axis of approach. This can easily be calculated by taking the dot product of those two vectors: $\hat{a} \cdot e$ where $\hat{a}$ denotes the unit vector of a. The result of this operation is the length of the projection of e on a, which can now be used for the calculation of σ_x.

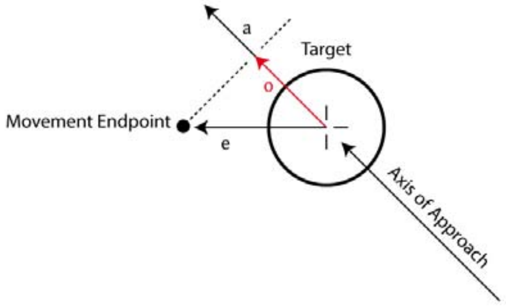


Figure 16: Measuring Distance of Movement Endpoints

What is the difference between using the Euclidean distance and using the projection? Douglas et al. (1997) used the following formula to compute the standard deviation:

$$\sigma_x = \sqrt{\frac{\sum_{i=1}^{n}[(x_i-\bar{x})^2 + (y_i-\bar{y})^2]}{n-1}}$$

Where n denotes the number of trials in that condition. One can see that in the numerator the squared Euclidean distance was used. The length of vector e in Figure 16 is larger than that of o.

This means that using the Euclidean distance produces a larger deviation and thus effective width W_e. This is demonstrated in the following example. Consider the situation in Figure 17. Two movement endpoints were recorded for this target condition, both outside the target. Point p_1 exactly hit the target, whereas p_2 deviated slightly from the approach axis. Suppose that the y-axis overshoot for p_2 is 10 pixels.

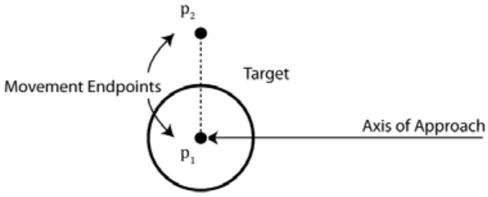


Figure 17: Movement Endpoints and SD

Since, in this situation, both movement endpoints exhibit the same x-axis value, the deviation *along the axis of approach* is actually zero. However, if one uses the Euclidean distance for calculation one obtains a standard deviation of $\sqrt{50} \cong 7$pixels.

Whereas one obtains the correct result, when using the distances d_{p1} and d_{p2} obtained from the dot product of the unit vector $\hat{a}$ of the angle of approach vector a, which in this case lies on the x-axis, with the movement endpoint vectors $p_1$and p_2 :

$$d_{p_i} = p_i \cdot \hat{a} = x_{p_i} x_{\hat{a}} + y_{p_i} y_{\hat{a}}$$

$$\sigma_x = \sqrt{\frac{\sum_{i=1}^{n} \left[\left(d_i - \bar{d} \right)^2 \right]}{n-1}}$$

It is certainly controversial which of the calculations is "correct". Following the description of the ISO standard, and with regard to the calculation of σ_x in the onedirectional case, one can surely say that using the Euclidean distance is not. However, considering the ecological[9] validity of a test, it seems reasonable that also deviations from the approach axis are considered in the calculation. This question could for instance be answered by comparing the model fit of the linear regression models calculated with ID values in one or the other way.

> *Research Question:* What is the "correct" way to compute the standard deviation of the movement endpoint scatter in a two-dimensional task?

When carrying out experiments with the continuous variant of the multidirectional task, another critical aspect must be considered to not confound results: the number of targets in the task. Figure 26 depicts two variants of the multidirectional task, one with eight and one with 16 targets. It can be seen that the amplitude of some trials (for example the one in red in Figure 18) is actually smaller than the diameter of the circle in which the targets are grouped. Comparing the variant with 16 trials against that with just eight, one can see that the effect is graver with

---

[9] The extent to how representative the test is with respect to what people do in "everyday life" (Reis & Judd, 2000)

the variant with fewer targets. Thus, some trials have smaller amplitudes than others and are therefore easier. Either these trials should be excluded from analysis or a variant with sufficient targets should be used. The ISO standard does not explicitly mention this, but the multidirectional task depicted there shows a task with 24 targets (ISO 9241-9:2000, , p. 33). Of course, having many targets per task increases the total time required by a participant to finish one test, possibly limiting the number of experimental conditions that can be presented in an experiment.

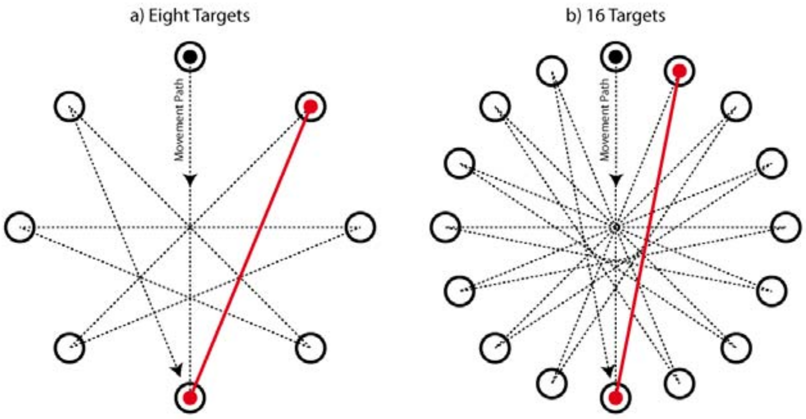


Figure 18: Multidirectional Task Variants

Another peculiarity of the multidirectional task, which can be discovered by analysis of the movement paths, is that participants sometimes seem to falsely guess the next target. In this case, movement paths often show “hooks” in the direction of the wrong target as for example depicted in Figure 19. This phenomenon also shows how movements and movement planning are intertwined, and that the planning of the next movement apparently is already undertaken before the current movement was finished. Such confusion could also be avoided by increasing the amount of targets. In a variant with many targets, the same alternating movement pattern can be used as in the unidirectional task, since targets almost lie in opposite locations. Premature planning of a movement to the wrong target, for instance, one target to the left or right to the actual target, does not have grave consequences and can easily be corrected by subsequent corrective movements.

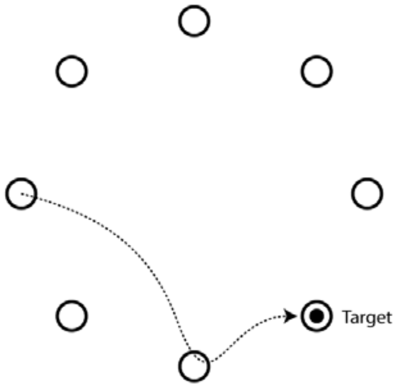


Figure 19: Multidirectional Hook Path

Similar, to the difficulties described for the onedirectional task, differences in methods are only subtle, but could possibly lead to confounded results. Those difficulties could, however, be avoided if a more thorough explanation was given by the ISO standard.

3.4.3 Cross Study Comparison

As shown in the previous sections, there are many pitfalls when one wants to reliably measure the speed and accuracy of a pointing device. Much of the standardization effort is motivated to build "(...) a coherent body of work" (Soukoreff & MacKenzie, 2004, p. 753). One purpose of such a body of knowledge could be to collect throughput values of various devices or device variations to have a basis for comparison of new devices or device variations. However, as can be seen from the table of IP values from experiments conforming to the ISO standard on page 34, this kind of comparison can only be made on a very coarse level. For instance, the throughput value for the mouse obtained from those studies spans a range from 3.7-4.9 bits per second, a difference of around 30%. In fact, even the mean results of a single study can only be used to roughly describe the performance since often the margin of error of those means is considerable. For instance, Douglas and colleagues (1997) conducted a study to compare the performance of a joystick and a touchpad and found that the index of performance of the joystick (2.15 bits/s) was 27% higher than that of the touchpad (1.7 bits/s). No doubt that there actually was a difference between the two devices. These were found to be statistically significant, but what about the absolute IP values and the magnitude of the difference? How can this knowledge be used?

In fact, 27% in mean difference is not much, considering the standard deviations of 0.4 bits/s for the joystick and 0.53 bits/s for the touchpad that were reported by Douglas et al. Thus, a fictitious other team of researchers that work on touchpad improvements could not even claim that any improvements were made on the design of the touchpad if they obtained a value of, for instance, 2.0 bits/s instead of 1.7 bits/s, that is, a difference of 0.3 bits/s or 18%. Suppose that the other team of researchers obtained these results with exactly the same procedure, similar standard deviations (0.5 bits/s), and the same number of participants (12). A simple t-test for comparison of these results calculated from this information would not even be significant: $t(11) = 0.873, p > 0.2$. The margin of error for the small difference of 0.3 bits/s for a 95% confidence interval is 0.7 bits/s. But even if the difference was significant, for example, if the other researchers obtained a value of 3.0 bits/s for the touchpad, it would not be justified to make claims with respect to the amount of improvement. Still, the margin of error of the mean difference of 1.3 bits/s is fairly high. It could be that the improvement was as small as 0.5 bits/s or as large as 2 bits/s (95% confidence).

Thus, besides difficulties in the experimental design, tasks, and procedure, one must be aware of the fact that results obtained from such studies can only be compared very roughly to the results of other studies. Particularly, if one wants to make cross study comparisons of design variants of the same device, because effect sizes are usually small in that case. Nevertheless, trying to meet the ISO requirements is sensible when designing comparative pointing and selection experiments because a comparison of movement time alone is worthless if one does not consider the accuracy of the movement –the methods in annex B of the ISO provide a good foundation for such tests.

4 Practice and Pointing Performance

Most certainly, pointing performance is dependent on the amount of practice that a user had with the pointing device. During practice with a pointing device, users learn how to manipulate the device to change the position of the pointer and they adapt their manipulations according to the way it reacts and moves across the screen. They might also develop new strategies on how to hold or actuate the device to perform better. Shouldn't these processes be considered when evaluating the performance of pointing devices?

In their assessment of the validity and practicability of the ISO 9241-9 tests, Douglas et al. (1999) pointed out that it is a "serious flaw" in the standard to not "(...) incorporate learning into the analysis". Their analysis was based on a draft version from 1999, and the final version published in 2000 does indeed give some guidance on how to handle effects of learning in annex B.3:

"In cases where training is needed, each subject should be allowed to learn the use of the input device until speed and accuracy do not show any significant improvements. Subjects should be given sufficient practice sessions to ensure that learning effects are stabilised. Thus, this should be verified by a procedure like the Duncan's Range test".

First of all, this guidance is very inaccurate because it equates learning and performance (the speed and accuracy). Learning is commonly defined as "(...) the process of acquiring the capability for producing skilled actions" (Schmidt & Lee, 2005, p. 302). This capability must be fairly stable, that is, *not* due to changes in motivation or because of physiological reasons (e.g. muscle fatigue). Stagnating performance could for instance indicate that the user reached a performance level that reflects the maximum amount of learning, or it could simply be due to the fact that the participant is exhausted and temporarily cannot increase his performance any further. Therefore the question must be asked if a procedure that solely examines *performance* (speed and accuracy) can justify claims about *learning*, as is suggested by the ISO standard. Secondly, the standard states that practice should be allowed when needed. However, no criterion is suggested to determine if this is the case with a particular device. Finally, as will be shown later in a literature review of existing pointing device studies, testing whether effects of practice have stabilized is not as straightforward as advised by the ISO standard, and the procedures for doing so vary widely among studies.

In this chapter the role of practice in pointing device tests is examined. First, a case is made for the importance of considering practice effects in pointing device evaluations based on a retrospective analysis of a comparative evaluation of a laserpointer and a mouse. Next, an overview is given about how effects of practice are treated in various existing pointing device evaluations. Third, a test application is presented that was developed in the course of this thesis to serve as the basis for an assessment of practice effects. The chapter is concluded by describing a longitudinal study that was carried out to elucidate and assess the role of practice effects in case of laserpointer interaction.

4.1 Experiment I: Laserpointer Performance

This first experiment compared the pointing performance of a laserpointer system against that of the mouse. Before considering the details of the experiment a short introduction into laserpointer interaction is given and the laserpointer system that was used in the experiment is described.

4.1.1 Laserpointer Interaction

A particular need for novel interaction devices other than the mouse and keyboard exists in the area of large high-resolution displays such as the Powerwall at the University of Konstanz. This 220 inch display has a size of 5.20 × 2.15 m and a resolution of 4640 × 1920 pixels. Because of the high resolution and the large size of the display users typically have to walk up to a certain distance to the display to be able to read small text or to step back to get an overview of all the information displayed on the screen. Since the mouse and keyboard require a table or other fixed surface for operation, users cannot take these devices with them when moving along the display, for instance, in order to click on a small button or read a small text. In view of this shortcoming, König and colleagues from the HCI group of the University of Konstanz developed a laserpointer interaction system that enabled users to carry out pointing and selection tasks while moving freely in front of the large display (König, Bieg, Schmidt, & Reiterer, 2007). Before considering the details of this system, an overview is given on existing systems, design challenges, and formal evaluations of the pointing performance of laserpointer systems.

4.1.1.1 Laserpointer Interaction Design Challenges

One of the first laserpointer interaction systems for use with regular video projectors that are commonly used for presentations was developed by Kirstein and Müller (1998). Their system used a regular off-the shelf laserpointer to control an on-screen pointer to enable selection of buttons and other controls. The laser spot was detected optically by analyzing the images of a camera that was aimed at the display. A particular issue of Kirstein and Müller's system and also that of other early laserpointer systems such as that of Olsen Jr. and Nielsen (2001) was that they were significantly affected by lag, which reduced the pointing performance that was attainable with these systems (see section 2.2.2.5). For instance, Kirstein and Müller report that their system was delivering 10 pointer locations per second, which means that the pointer position is only updated every 100 ms. Similarly, Olsen Jr. and Nielsen report that their camera delivered only 7 frames per second, which amounts to a lag of around 140 ms, although the authors reported that sometimes even longer lag was present. The reasons for this lag were that only very low refresh rates of the pointer location were possible due to the high computational demands of the image analysis procedure and because of low camera refresh rates.

Another design challenge is due to the fact that the laserpointer is an absolute and direct input device. In contrast to other direct devices like a touch screen, the laserpointer can either be used directly on the display, just like a stylus on a touch tablet, or from a distance to the display. This imposes an issue that affects pointing performance: the jitter induced by tremor of the hands when holding the laserpointer. Due to this jitter the laser spot cannot be kept still to click on a button or other item. A laserpointer system that first addressed this issue was presented by Wissen and colleagues (2001). In order to reliably position the on-screen pointer and reduce the effect of the jitter they used a special transfer function: when only small changes of the laser spot were detected (< 3 pixels from current position), the on-screen pointer was not moved at all. If, on the other hand, a larger displacement of the laser spot was detected, then the pointer location was also changed as a function of this displacement. This compensated high-frequency and low amplitude jitter in their system and enabled users to keep the pointer stationary on an object. Other systems used Kalman filters, a filtering mechanism that combines the measured laser spot location with a prediction of the next location that is derived from a mathematical model of the filter (Frolov, Matveyev, Göbel, & Klimenko, 2002; J. Y. Oh & Stuerzlinger, 2002).

Another laserpointer design challenge pertains to the selection technique. First laserpointer systems that used off-the-shelf pointers were experimenting with dwelling techniques (Olson & Nilsen, 1987; Wissen et al., 2001). The dwelling technique triggers a selection whenever the laser spot remained within the boundaries of a selectable object for a pre-defined amount of time. This has the disadvantage that some amount of selection time, depending on the dwelling duration, is always required for selection and thus sets a fixed lower limit for the speed of selection. Other systems experimented with the on-off button of the laserpointer (Olsen Jr. & Nielsen, 2001). This was found to be problematic because, if the laser is turned off, no pointer feedback can be given. When turning the pointer on again, the exact location of its appearance cannot be controlled by users very well and therefore this information cannot be used to trigger any events (Myers, 1989). Another technique was introduced by Davis and Chen, who detected entire strokes of the laserpointer that could then be used by the application for triggering gesture based events (Davis & Chen, 2002). Finally, some systems used custom laserpointers with integrated buttons that either sent the button signal electronically via a built in wireless transmitter (Wissen et al., 2001), or encoded the button press information in interruptions of the laser light (Matveyev & Göbel, 2003).

4.1.1.2 Evaluations of Pointing Performance

Some of the systems described above were also evaluated formally. For instance, Myers and colleagues (2002) developed a laserpointer system and conducted an experiment to compare it against a mouse and other devices. The results showed that the performance of the laserpointer (5.08 bits/s) was below that of the mouse (6.98 bits/s). The tests employed were Fitts' tapping tests as described in section 3.1.3 but the procedure and calculations did not adhere to the ISO 9241-9 standard and so the absolute results cannot be compared to other studies. Additionally, Myers et al. conducted an experiment where participants tried to hold the laserpointer stationary so that measurements could be made with respect to the magnitude of the jitter induced by hand tremor. The test was carried out at several distances to the display (5, 10, and 15 feet; corresponds to 1.5, 3, and 4.5 m). The results showed that the average radial distance from the target increased significantly, for instance, from 0.18" (0.45 cm) at a distance of 1.5 m to 0.46" (1.17 cm) at a distance of 4.5 m. This means that it is getting harder to point steadily to a specific location with the laserpointer the farther the distance to the screen. Another evaluation of a laserpointer system was conducted by Duncan et al. (1955). Their comparison of an infrared laser against a laser with visible light revealed that participants could perform better with the laser with visible light. Duncan and colleagues conjectured that this was due to tracking latency which prevented optimal integration of visual feedback in controlling the pointer location with the infrared pointer. In contrast to the study by Duncan et al. and Myers et al., the study of Oh and Stuerzlinger conformed to the standards of ISO 9241-9. The authors also used a custom infrared laserpointer with an integrated button and smoothed the laserpointer signal with a Kalman filter. Results of the comparison against the mouse showed that the performance of the mouse (3.98 bits/s) was better than that of the laserpointer (3.04 bits/s).

4.1.1.3 The University of Konstanz Laserpointer System

In contrast to many of the systems presented before, the laserpointer system of the University of Konstanz was not used with a regular video projector but with the Powerwall. The system detects the infrared laser spot with three cameras that are aimed at the display. In case of the Powerwall this is done from behind the display so that users can move around in front of the display without concealing the surface from the view of the cameras. The position of the laser spot on the screen is determined by locating the spot in the camera image. This position is then mapped to the coordinates of the screen. Once before this can be done, the system must be calibrated to establish this mapping between the locations in the camera image and the screen position. In view of the design challenges explained before, the key features of the system are presented in the following:

- Low lag: The system exhibits an extremely low lag of around 15-25 ms and it is thus largely within the limit of 20 ms which, according to the ISO standard 9241-9, does not degrade pointing performance because it is not perceptible by users (ISO 9241-9:2000). The low lag of the system is achieved through speed-optimized image analysis routines and industrial cameras with high frame rates (around 80 frames per second).
- Low-energy laser: In contrast to class 3 lasers that are commonly used in laserpointers for presentations, the laserpointer system is equipped with a 0.55 mW class 1 infrared diode which is completely harmless and cannot damage the human eye.
- Infrared laser: The laser spot on the wall is invisible to the user and thus, a graphical pointer can be displayed instead. The movements of this pointer can then be smoothed to compensate for the jitter induced by tremor of the hands.
- Dynamic jitter compensation: In order to compensate for jitter, a Kalman filtering system is used. A special property of this system is that it operates in two modes: static and dynamic. If the speed of the pointer slows down then the static mode is used, which is optimized to keep the pointer stationary. This is useful if the user wants to point to an item to select it. If the pointer speeds up, the filter switches into a dynamic mode to smooth continuous movements, for instance when drawing.

Figure 20: Latest revision of the laserpointer

The laserpointer itself is concealed in a metal casing that included three buttons. This has the advantages that interaction techniques could be implemented that were similar to those of the mouse, for instance, clicking a button while pointing to an item to select it, or pressing and holding a button to drag it.

4.1.2 Overview of the Experiment

In order to assess the pointing speed and precision that can be achieved with the laserpointer system, a controlled experiment was carried out. The experiment was designed and conducted by the author and his colleague Anton Stasche during the course of their masters program. The experiments are described in detail in publications by König, Bieg, Schmidt & Reiterer (2007) and König, Bieg & Reiterer (2007). A retrospective analysis of the experiment is used here to illustrate the importance of considering effects of practice and thus motivates the experiment II that is presented later.

In the experiment with 16 participants, the laserpointer system was compared against the mouse. Additionally, both devices were tested at two distances to the display: 3 and 6 m. The dependent measures were the movement time (MT), error rate, the index of performance (IP), and the effective index of performance (IP_e). A continuous onedirectional and multidirectional tapping task were used and the performance calculations adhered to the ISO standard as described in section 3.3. Based on the results of previous evaluations of other laserpointer systems that were described in the previous section it was hypothesized that the mouse's performance should be a little better than that of the laserpointer because it benefitted from the stability of a desk. In view of the findings of the experiment of Myers et al. (2002) it was assumed that the laserpointer's performance would deteriorate because the larger distance would increase the effect of hand tremor on pointing accuracy. On the contrary, the mouse should remain unaffected from the increased distance because it is an indirect pointing device.

The experiment was conducted in a within-subjects design. This means that each of the 16 participants was presented all of the 2 × 2 factor combinations (laser at 3 and 6 m, mouse at 3 and 6 m). The design was balanced between subjects creating four sequence groups as can be seen in Table 7. Each of the 16 participants was randomly assigned to one of these groups.

Table 7: Counterbalancing groups from experiment I

Group	Factor Combinations	Laser Group
1	M3 L3 M6 L6	3-6
2	L3 M6 L6 M3	
3	L6 M3 L3 M6	6-3
4	M6 L6 M3 L3	

M = Mouse, L = Laserpointer
3 = 3m, 6 = 6m

Two types of tests were used during each of the factor combinations. First, a longer multidirectional task was presented so that users accustomed themselves to the device and distance condition. After that, a onedirectional task was presented. It was assumed that the practice trials during the multidirectional task were sufficient so that practice effects were stable by the time the onedirectional task was presented. The whole experiment took around one hour, a duration which was chosen based on experience from a pre-test and experience from literature, so that participants would not get exhausted during the experiment.

4.1.3 Results

The results showed that the laserpointer's performance was slightly lower than that of the mouse. Particularly the error rate was much higher than that of the mouse. As expected, the laserpointer's performance was lower at 6 m than at 3 m. This was not the case for the mouse as can be seen in Table 8 and Figure 21.

Table 8: Results from experiment I

	Laser		Mouse	
	3m	6m	3m	6m
IP_e	3.52 (.35)	3.17 (.38)	3.96 (.42)	4.01 (.30)
IP	3.78 (.53)	3.62 (.60)	4.22 (.37)	4.20 (.33)
MT	1086 (139)	1133 (168)	954 (83)	957 (67)
Error Rate	.14 (.08)	.20 (.10)	.08 (.08)	.09 (.06)

Values are in bits/s for IP/IP_e, ms for MT
Values in braces are standard deviations

Repeated measures ANOVA (analysis of variance) showed significant main effects for the device factor for all measures ($p < .01$). This confirmed that the differences that were found between the laserpointer and the mouse were significant and not due to chance. Significant device × distance interactions were found for IP_e and the error rate ($p < .05$). Analysis of the simple effect of distance on the device factor for these two measures showed significant effects for the laser ($p < .05$) but not for the mouse. These findings confirmed that the laserpointer's performance was affected by the distance factor and the performance of the mouse was not, as hypothesized before.

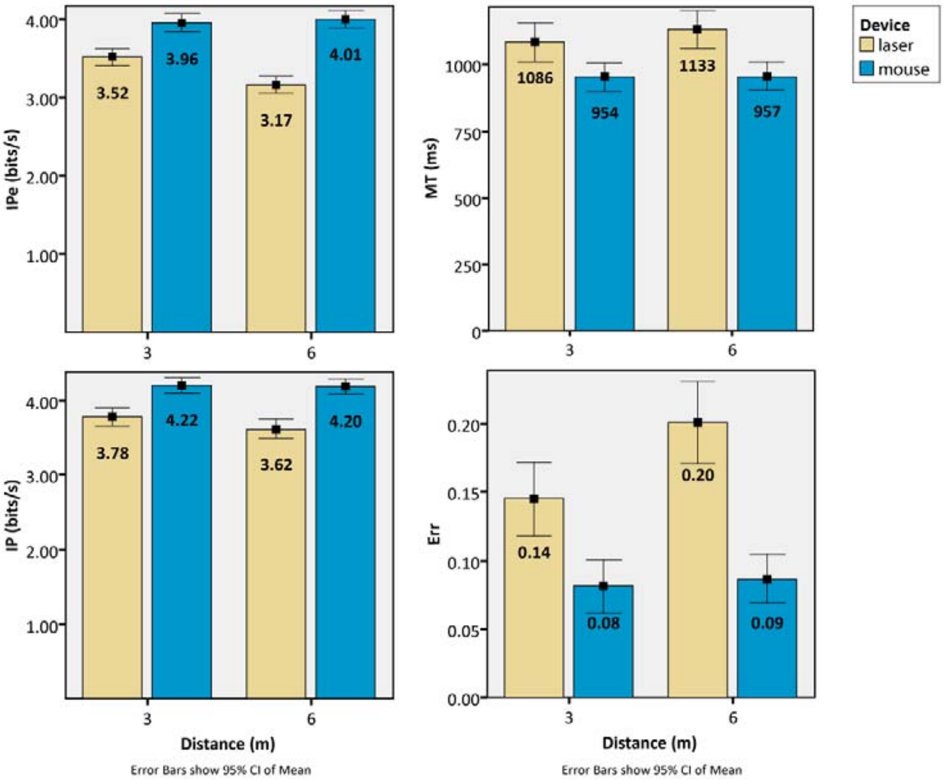


Figure 21: Results from experiment I

A closer analysis of the influence of the counterbalancing sequence group revealed that practice had a major effect on the laserpointer performance:

The left chart in Figure 22 shows the performance of the participants in the 3-6 sequence group (laser at 3 m first, and then laser at 6 m) compared to the performance of the 6-3 sequence group (laser at 6 m first, and then laser at 3 m). Participants in the 6-3 sequence group that were first using the laserpointer at a distance of 6 m performed much better at a distance of 3 m (3.69) than the participants of the 3-6 group that started using the laserpointer at 3 m (3.35). On the other hand, the performance of the participants that started using the laserpointer at a distance of 3 m (3-6 sequence group) did not decline much when moving to the 6 m condition (3.27). Their performance was much better than the performance at 6 m of the participants that started at 6 m (3.06, 6-3 sequence group).

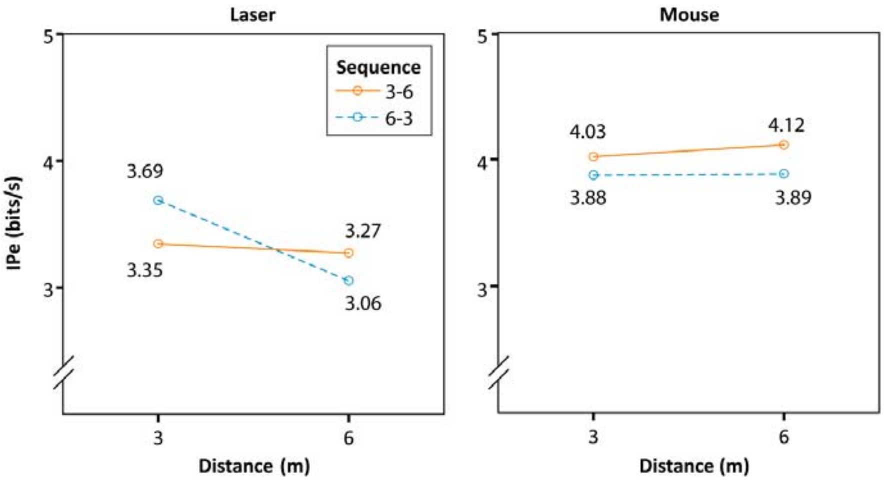


Figure 22: Results by sequence group

A possible explanation for these phenomena could be that performance was influenced by the superposition of two effects. On the one hand, pointing performance at 3 m benefitted from the smaller distance and on the other hand, performance improved irrespective of distance due to practice. Seemingly the practice phase in this experiment was not long enough to "stabilize" performance in each of the experimental conditions as advised by the ISO standard.

Additionally, the practice effect was slightly biased: The difference between the two sequence groups at 3 m was $3.69 - 3.35 = 0.34$. In contrast to that, the difference between the groups was $3.27 - 3.06 = 0.21$ at a distance of 6 m. A possible explanation for this is that using the laserpointer at a distance of 6 m required a better strategy, for instance to compensate jitter. Participants that used the laserpointer at 3 m first, were not "forced" to develop such a strategy because jittery motions of the pointer on the screen due to hand tremor were not as strong at that distance. On the other hand, the participants that started with the laserpointer at 6 m developed an anti-jitter strategy because the effect of hand tremor was larger at that distance. Perhaps this anti-jitter strategy was also helpful to improve performance afterwards, in the 3 m condition.

As can be seen in the right chart in Figure 22 no such effects were present for the mouse and only a general difference between sequence groups can be found, which can be attributed to variability between subjects. The slight improvement in one of the groups was not significant ($F_{1,14}$=0.620, p=.444). The performance of the mouse was neither affected by distance nor by practice. Because the mouse is an indirect device, the former can easily be explained. Increasing the distance to the screen does not change the requirements for actuating the device in any way. Moving the mouse to the left or right also moves the pointer the same amount to the left or right no matter if the mouse is used at 3 m or at 6 m. This is in sharp contrast to the direct laserpointer, where increasing the distance to the input area affects performance due to an increased effect of tremor. Additionally, it seems that practice had no grave effect on the mouse because participants already had a tremendous amount of experience with that device. In fact, it is probably hard to find anyone, particularly among university students which commonly volunteer for such experiments, who has no experience with the mouse at all. On the contrary, none of the participants had any experience with the laserpointer system before.

4.1.4 Conclusion

Generally, the hypotheses formed prior to the experiment could be confirmed. The performance of the mouse was slightly better than that of the laserpointer. The discrepancy increased with increased distance to the display because the performance of the laserpointer was worse at 6 m than at 3 m. Further analysis of the counterbalancing sequence group revealed that, apparently, the performance of the laserpointer was affected by effects of practice whereas the performance of the mouse was not. This raises the question if a longer practice phase could have improved the performance of the laserpointer, to match that of the mouse.

4.2 Analysis of Practice in Pointing Experiments

In light of the results of the first experiment presented in the previous section, a literature review was carried out. The goal of this review was to find out how practice effects were taken into consideration in other studies that evaluated pointing devices based on the Fitts' tapping paradigms. A table giving an overview can be found in appendix D. The review is focused on those studies that were published after the release of the final version of the ISO standard. Nevertheless, some studies are included that were published before, because these used interesting methods for the analysis of practice.

4.2.1 Considering Practice in Experimental Design/Procedure

4.2.1.1 Counterbalancing

As most of the studies use within-subjects designs, researchers usually counterbalance the sequence of presentation of devices between subjects to *compensate* for effects of practice so that differences between devices can still be determined. If between subjects counterbalancing is used, symmetrical transfer is assumed. This means that, for example, if the independent variable has two levels A and B, that is, two devices are compared in a test, it is assumed that performing a task with device A followed by device B has the inverse effect of presenting B first, followed by A.

If this is not the case, results are possibly confounded due to asymmetrical transfer (Martin, 2004, p. 163). This means that the two effects of presenting A after B and B after A do not neutralize each other. For instance, device A might have a shape that suggests a certain strategy that is very effective, like holding it in a special way. This strategy is learned while using device A. Device B, on the other hand, has a totally different shape which suggests a different strategy. Nevertheless, suppose that the strategy learned for device A could also be applied to device B and that it would also improve performance with that device. On the other hand, the strategy of device B cannot be applied to device A, for whatever reason. Thus, the performance for device B of participants in the AB group benefits from the knowledge of the strategy that was learned when using A, whereas the performance of device B of participants in the BA group does not. If such transfer of learning is expected, the experiment should not be carried out with a within-subjects design. Instead, the factor level should be tested between subjects – which of course has other disadvantages such as the increased amount of participants required (Martin, 2004). The biased effect of practice in experiment I was an example where such asymmetrical transfer occurred for the laserpointer and the distance factor, though the effect was only small in that case.

Analysis of this factor is important since asymmetrical transfer of skill could severely confound results, particularly if one wants to come to conclusions about definitive performance values that can be compared to other studies. For instance, if the effect in experiment I would have been larger and the other way round (e.g. using the laserpointer at 3 m boosted the performance at 6 m but not vice versa), then the main results could have indicated that the laserpointer's performance was actually better at 6 m than at 3 m.

4.2.1.2 Expert Criterion

An interesting approach was pursued by two of the earlier studies (Card et al., 1978; Jagacinski & Monk, 1985). Rather than analyzing effects of learning after the experiments, they defined a criterion level of performance which, once reached, was used to terminate practice.

For instance, in their now classic experiment Card et al. (1978) analyzed the performance of a mouse, an isometric joystick, step keys, and text keys by using a text selection task which was roughly comparable to a discrete multidirectional task. A comparison to today's methodology as proposed by the ISO standard can be found in (MacKenzie & Soukoreff, 2003). In their experiment, Card et al. defined that learning had no impact on positioning time if the average time of the first and last third of 600 consecutive trials did not differ significantly (this was confirmed with a t-test at a significance level of 5%). The criterion was reached after 1200 to 1800 trials which took participants around four to six hours. However, not all participants reached this criterion with each device (15 out of 20 participant-device combinations).

A similar procedure was used by Jagacinski and Monk (1985) who conducted a discrete tapping experiment with a multidirectional task. Their goal was to determine if Fitts' Law was applicable to situations where participants were uncertain about the exact location of the next target and whether the law would hold for head movements. Sessions were repeated for several days until performance reached an asymptote which was defined as a deviation of less than 3.5% from a four day performance mean. The number of days it took each participant to complete the experiment varied from six to 29 days.

From these examples the obvious drawback of this approach is discernible: The participants sometimes failed to reach the criterion at all, making it difficult to use the results from these participants. Additionally, reaching the criterion took an amount of trials impractical for most studies. Therefore, it is not surprising that attempts to analyze the effects “in situ” rather than in a post-hoc fashion are very rare among pointing device studies.

4.2.2 Post-hoc Analyses of Practice Effects

4.2.2.1 Analysis of Blocked Trials

One of the most common methods of analysis was to examine differences in blocked trial means (i.e., “bundles” of several trials), mostly with standard statistical procedures such as t-tests or ANOVA (analysis of variance). Creating such bundles of trials is required because the variance between results of single trials is usually high. Analysis of these bundles can reveal the presence of an effect of practice or an interaction between practice and some other experimental factor, for instance device, which would indicate differences in the patterns of performance changes between devices. For instance, MacKenzie and Oniszczak (1998) analyzed their results with ANOVA and found a main effect for the block factor. Three blocks of trials were used for each of the three different techniques they evaluated in their experiment. The authors used profile plots that showed the performance index of each technique plotted against blocks to reason about the effects of practice. Although the authors did not find any significant effects in this respect, such an analysis could for example reveal that practice plays a large role for one device, whereas another remains unaffected by further practice.

When only two blocks are available for comparison, merely the presence or absence of practice effects can be discovered of course. No claims can be made with respect to stabilization of learning effects as demanded by (ISO 9241-9:2000). For this purpose, more blocks must be compared. The statistical methods applied for such tests are commonly referred to as multiple comparisons or contrasts. Some researchers used simple t-tests to compare blocked trials. For instance, Mithal and Douglas (1996) carried out a study with a onedirectional tapping task to compare the mouse against an isometric joystick. Because the authors reasoned that participants were less familiar with the joystick, they scheduled three sessions per device with 600 trials each, so that participants had enough practice before the two devices were compared. Analysis of practice effects was then carried out on 15 blocks of trials with 120 trials each. These were compared with a pair-wise t-test which showed that learning had no effect after block three for the mouse and block seven for the joystick. The ISO standard, on the other hand, suggests the use of Duncan’s range test which can be used to form groups of means that cannot be differentiated significantly (Duncan, 1955). This test was used in none of the studies reviewed for this thesis and, to the knowledge of the author, also no pointing device experiment exists that uses this test. However, two studies used Helmert contrasts (Douglas et al., 1999; MacKenzie, Kauppinen, & Silfverberg, 2001)[10], which is also a method for comparing multiple means. Unlike in Duncan’s test, Helmert contrasts require that, from the start, the means that are contrasted can be ranked in some sensible fashion. Clearly, this is the case with block means, since these can be ranked according to time or amount of practice received. Helmert contrasts can be calculated easily by contrasting the first block with the mean of all later blocks, the second block with the mean of all later blocks, etc., until the block before the last, which is only

---

[10] Both studies claim to have used “Helmhert contrasts”, spelled with an “h”. However, to the knowledge of the author, there exists no such procedure. The authors probably meant “Helmert contrasts” (cf. Bock, 1975).

compared to the last one. In the two studies mentioned before, such contrasts were used to determine the point in (blocked) time, where no more significant learning occurred.

4.2.2.2 Establishing a Performance Curve

An interesting feature of a study of Card et al. (1978) was that, to further investigate the effect of practice, the data was fitted to a curve of the form $T_N = T_1 N^a$ where T_N denotes the positioning time on the Nth block, T_1 the time of the first block, and a a constant which can be interpreted as the rate of improvement. This rate was determined through linear regression analysis applied to the logarithmic transformation of this power equation: $\log T_N = \log T_1 - a \log N$ (see Figure 23). Card and colleagues interpreted this rate of improvement as the ease with which a device could be learned.

It is very common that effects of practice can be captured by such a power law, not only in the case of a motor skill, but also in a variety of other tasks (e.g. memory skill, cf. Haberlandt, 1999), simply because the rate of improvement diminishes over time. Taking up the discussion about the difference of learning and performance gains, Schmidt (2005, p. 304) notes that one must be aware that these curves do not depict learning per se and that a better term would be "performance curves". In the case of motor control tasks many studies were published that established such a relationship between practice and performance in simple tasks such as mirror drawing (Snoddy, 1926) or more complex tasks such as cigar making (Crossman, 1959).

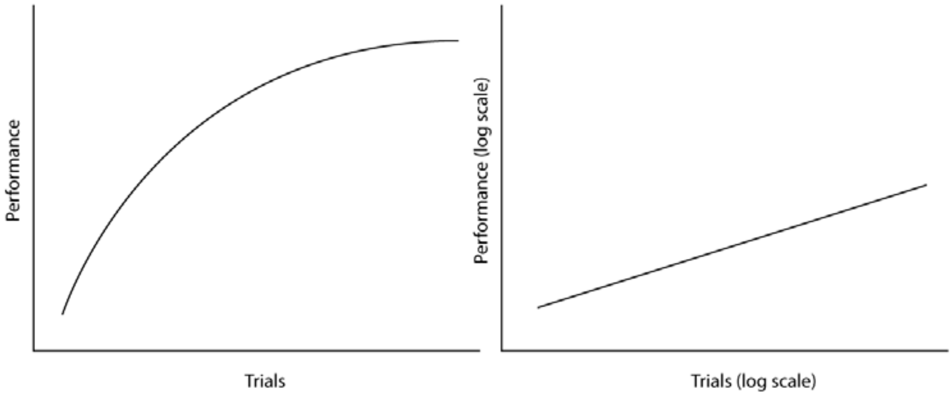


Figure 23: Idealized Performance Curve

In their analysis, Card et al. (1978) used the slope regression determinant a of the practice curves for each device to determine the magnitude of improvement, as an indication of the ease of learning exhibited with each device. Their results showed that the greatest magnitude was achieved with the mouse and text keys, whereas less improvement was made with the joystick and the step keys. Around 60% of variance was explained by the learning curve and the authors suggested that, due to gaps during sessions, even higher correlations could have been achieved.

Another example of such an analysis is the experiment conducted by Isokoski and colleagues (2007, experiment four). Four sessions were held in which a trackmouse, a combination of a trackball and a mouse with dual-stream input (i.e. two cursors), was compared to a traditional mouse in a custom tapping task that involved the alternating selection of a toolbar and other items. Although trackmouse performance was worse than that of the mouse, the magnitude of the performance increase which was derived from a power law analysis was larger and by session four, performance with the trackmouse was better than that of the mouse.

It seems that power law analysis is a good tool for the investigation of performance differences between devices. Yet, as was noted before in section 3.1.3.3, using a model derived from regression analysis for *prediction* is dangerous and results cannot be used as evidence that performance will actually reach a certain level provided that practice tasks are carried out for a certain amount of time. Nevertheless, power law results could for instance be used to stimulate further tests, if results indicate that one device seems to “catch up” on the performance of another device as in the study by Isokoski et al. before.

4.2.3 Conclusion

The review of literature shows that treatment of effects of practice is rather diverse and it seems that this is still the gravest “blind-spot” of the ISO standard. This is also probably due to the fact that analysis of effects of practice almost always requires a more complicated experimental design and longer or more testing sessions. This means that it is also much more time-consuming and expensive to carry out such a test.

Of the methods presented, post-hoc multiple comparison procedure seem to be the most practicable. The question with those methods is of course a) which one to use and b) how many blocks and trials per block to schedule for the test. The former question could possibly be answered by a more thorough analysis and comparison of statistical procedures or by a consensus of researchers in the field. The latter question could for example be answered by taking the results of other studies as a reference, provided devices are similar, or by conducting a longer multi-session pre-test to get a feeling for the magnitude and characteristics of improvement through practice. In experiment II that was conducted to analyze the effect of practice on laser-pointer performance, which will be described in section 4.4, post-hoc comparisons and power-law analysis were used.

4.3 The Bubble Test: A Novel Testing Application

Before considering the experiment that was conducted to elucidate the effect of practice on laserpointer performance, an overview is given on a novel testing application that served as the main task for this experiment: The bubble test. One goal of creating the testing application was to implement a discrete multidirectional tapping paradigm. As discussed in section 3.4.1., it is supposed that the discrete variant more closely resembles the requirements of a pointing movement in a typical application context and thus has higher ecological validity. Additionally, the multidirectional variant was chosen because the angle of a pointing movement could confound the results obtained. For instance, some device might be suited well for horizontal movements but shows lower performance when carrying out a vertical movement because this might require unpleasant movements of the arm or wrist. Testing movements with several different angles accommodates for this.

Another goal was to create a test with a game-like atmosphere that provides rich feedback to motivate participants to perform well for longer periods of time so that it could be used in a multi-session pointing experiment. It is suggested that this also improves the ecological validity of the test because pointing movements in an office or information worker scenario are typically followed by rich feedback, for instance clicking on a button brings up a different screen or navigates the browser to another page, etc. Additionally such pointing movements are made in a meaningful context, while pursuing a certain goal (writing a letter, searching a book). All this is missing in a typical pointing experiment. In order to provide at least a basic level of context, the test is given a game-like look, and a meaningful task: collect as many bubbles as quickly as you can. Additionally, immediate audio-visual performance feedback was given during a trial and additional verbal information about the performance was built in after a block of trials.

4.3.1 Overview of the Application

The bubble test application is a Java/OpenGL program that was specifically designed to run on the Powerwall in full resolution of 4640 × 1920 pixels. While it was primarily used for testing the laserpointer, any other pointing device could be used with the test provided that normalized pointing coordinates (between 0 and 1) and button presses are sent to the application via a simple ASCII UDP protocol.

The test shows a blue marine scenery with white bubbles of different sizes floating from the bottom of the screen to the top. A crosshair pointer can be moved smoothly over the scenery by pointing to the respective position with the pointing device. In phase one of the test, the *free-bubble phase*, participants can simply point and click on the bubbles that are continuously floating on the screen. This phase is used as an initial warm-up phase so that participants can accustom themselves to the device and the surroundings. When a participant hits a bubble it disappears with a little burst animation and a fitting “plop” sound to indicate success. The second phase, the *multidirectional phase,* essentially is a discrete multidirectional tapping test which presents several target bubbles of different sizes and with different amplitudes.

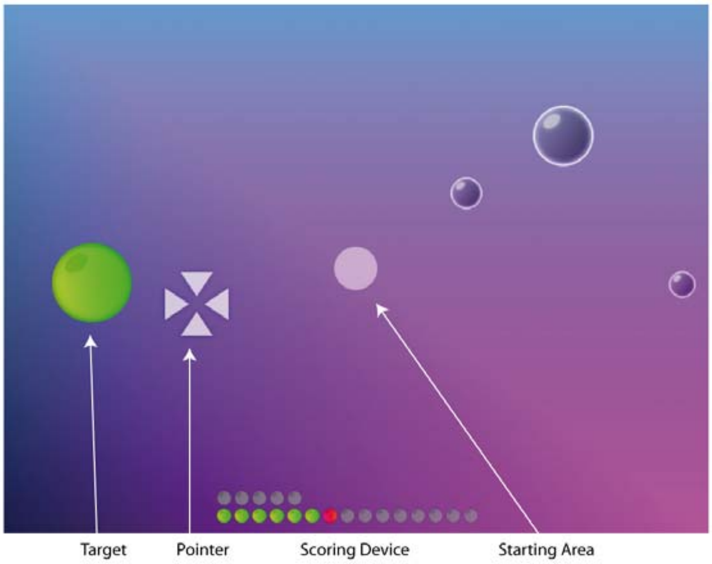


Figure 24: Bubble test screenshot showing the multi-directional phase

4.3.2 The Discrete Multidirectional Tapping Paradigm

4.3.2.1 Overview of the Multidirectional Phase

The game scenery during the multidirectional phase of the game is shown in Figure 24. A white circular object, the starting area, is presented in the middle of the screen. Once participants move the pointer to this circle and dwell there for at least 500 ms, a target bubble appears. In contrast to the other bubbles this target bubble is colored in a strong green color which makes it easily discernible in the scenery. The appearance of the bubble is accompanied by a "bubbly" noise to inform the participant of its appearance. Once the pointer is moved over the bubble and the pointing device button is pressed, the bubble bursts in the same fashion as in the free-bubble phase. If the bubble is missed, a short horn noise is sounded. In case of a miss, the target bubble remains at its position until it is finally hit. Additionally, at the bottom of a screen, an array of tiny bubbles serves as a scoring device. At the beginning of the task all scoring bubbles are painted in a dark gray color, when a hit is scored, one of the scoring bubbles turns green. If the target bubble of this trial was missed it turns red instead. Thus, participants have direct feedback on how many bubbles remain in the current block of trials and how many bubbles were hit or missed during the block.

After a hit was scored, the circular starting area begins flashing slightly to indicate that the pointer has to be returned to the middle position. If the pointer is not moved to the center position the next target bubble will not appear. This procedure is required so that it is assured that the pointer is always returned to the center position. In this way the amplitude, the distance to move in a single trial, can be a controlled variable during the experiment.

4.3.2.2 Target Amplitudes and Widths

The target amplitude and width parameters as well as the number of trials per amplitude-width condition can be specified by the experimenter before an experimental session. The amplitude defines the distance from the center position to the center of a target bubble. The width defines the diameter of a target bubble. These combinations are then presented in random order. The amplitude of the respective trial and height of the display were constraining factors with respect to the target location (see Figure 25). Since each move began in the middle of the display, at the starting area, amplitudes could generally be no larger than around 2000 pixels (horizontal resolution of the Powerwall 4640/2 plus some additional space for the target and room for overshoot). Additionally, if the amplitude chosen was greater than half of the vertical resolution, the target could only be presented in a certain range of angles so that the target, or parts of it, was not located outside the display area. An angle was then selected from the range of valid angles, the grey area in Figure 25.

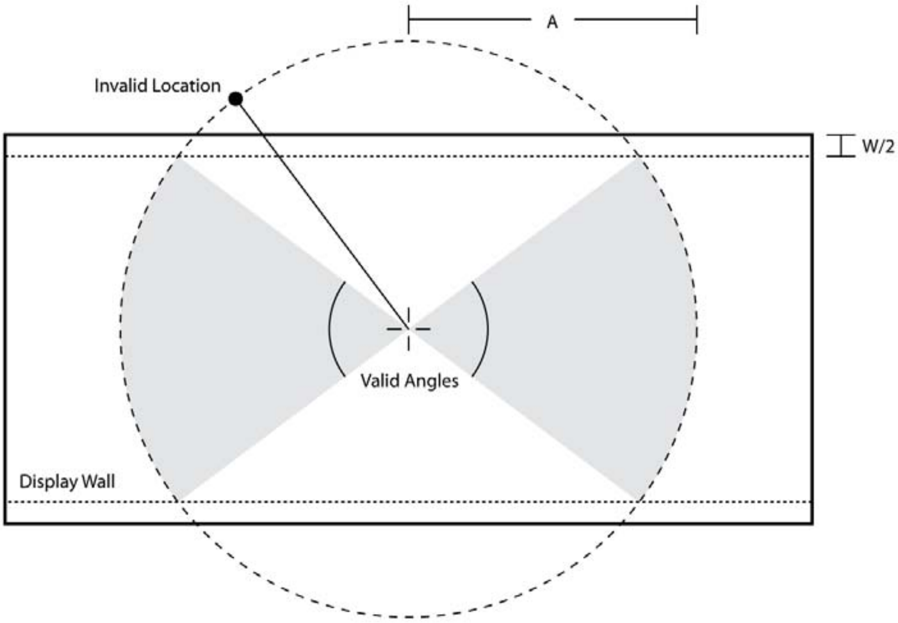


Figure 25: Valid angles for target presentations for given amplitude; valid angles are drawn in gray

4.3.2.3 Measuring Movement Time

Although discrete paradigms were used widely in pointing device evaluations, as explained in section 3.4.1, the exact procedures to present target stimuli and to measure movement time often differed. For instance Silfverberg et al. (Silfverberg et al., 2001) presented the target once the participant clicked on a home square, which is equivalent to the starting position in this application, and measured movement time in between the click on the home square and the target. This method was not used in the test because it confounds the measurement of movement time with reaction time or time for searching the target on the screen. Coupling the movement time measurement to the first movement of the pointer would be a possibility to solve this problem. However, because the test should be used for testing the laserpointer, which usually is in continuous motion due to the jitter problem described before, this procedure is also not an alternative. Instead, a method similar to that proposed by MacKenzie and Buxton (1992) is used. Movement time is measured from the moment the pointer leaves the boundaries of the starting area in the middle of the screen. If the pointer leaves this area prematurely before a target is presented, the dwelling countdown is stopped and no target appears before the pointer is moved back into the starting area and remains within that area for the pre-defined amount of time. Before conducting an experiment with this procedure it was tested that participants were able to keep the pointer inside this starting area without much effort.

Comparison of the bubble test against a continuous tapping test has shown that the measured effective performance IP_e was almost 25% higher in the bubble test. One reason for this could be that the effective width W_e measured by the bubble test was generally smaller, thus increasing the a posteriori difficulty. The other reason could be that the movement time was smaller in the bubble test. A quick empirical test with the same amplitude and width combination for both types of tests showed that the difference in W_e between both tests was only small and probably due to chance (see Table 9). The effective width was even higher for the bubble test which actually *reduced* the effective index of difficulty ID_e of the bubble trials. Nevertheless, IP_e was much larger for the bubble test. The cause of this was the difference in movement time between both tests: 486.9 ms in the onedirectional test compared to 393.0 in the bubble test.

Table 9: Discrete Multidirectional MT

Test	Overshoot SD (px)	MT (ms)	W_e (px)	ID_e (bits)	IP_e (bits/s)
Onedirectional	21.2	486.9	87.5	2.5	5.1
Bubble	24.3	393.0	100.3	2.3	6.0

The likeliest reason for this difference is that the procedure used in the bubble test "swallows" some of the movement time that is initially required to accelerate the device. In order to investigate this assumption, an analysis of pointing samples was made. Figure 26 shows the normalized positions for x and y pointer coordinates for a single trial of the discrete multidirectional task. On the abscissa the sample number is shown. Since 60 samples were recorded per second, each sample corresponds to 1/60th of a second (~16.6 ms). Recording of samples started once the target was presented and stopped once the user pressed the button to select the target. One can see that, after some time of dwelling in the middle of the screen (coordinates 0.5/0.5), the pointer accelerated quickly and moved to the top of the screen ($y \cong 0.85$) and slightly to the left ($x \cong 0.4$). Also shown is the time the pointer remained inside the starting area t_s. Although much of it supposedly was reaction time, some of it was already spent commencing the movement (t_a) which is indicated by the slight slope of the curve in that area. This time is not recorded as movement time, since recording begins once the pointer leaves the starting area.

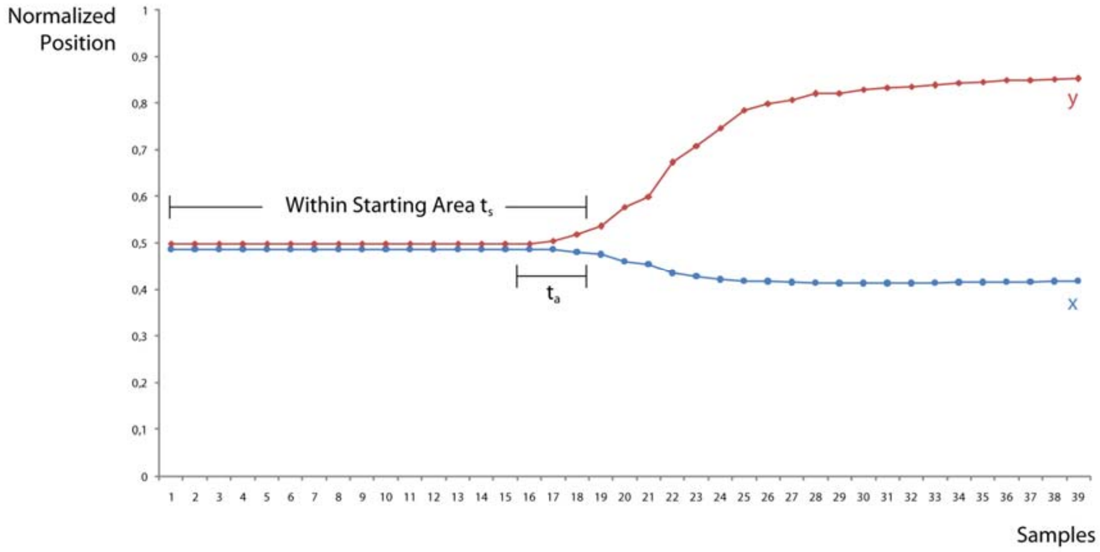


Figure 26: Movement graph of one of the discrete bubble test trials

One can see that t_a spans around two to three samples which correspond to around 33.2 to 49.8 ms. Further differences between the movement times of both tests could be attributed to peculiarities of the continuous test: Since this test records MT in between alternating selections, time taken for reversing the movement direction may possibly confound the movement time measure for all continuous trials except for the first.

Whatever causes the higher performance value for the discrete task, the phenomenon is consistent with the findings of other researchers as shown previously in section 3.4.1. Because of the difference, the absolute results cannot be compared to those obtained with a standard continuous test. Nevertheless, absolute comparisons can be made across experimental conditions if the same type of test is used.

4.3.3 Incorporation of Feedback

Many researchers have used feedback in one form or another for “(...) maintaining subjects' motivation” (Card et al., 1978). Most of the studies used visual and audible signals, such as a little beep to indicate a hit or a miss (e.g. MacKenzie & Oniszczak, 1998; Mithal & Douglas, 1996). Others gave feedback by displaying trial times or error rates. For example, Card and colleagues (1978) showed participants their average pointing time and error rate after a block of 20 trials. Jagacinski et al. (1985) not only presented the individual results of each participant but also showed those of the other participants. Isokoski et al. (2007) conducted a longitudinal study with sessions held on four successive days. In order to assure that participants were working “(...) at the limit of their capabilities” participants were instructed based on their previous results, that is, when they were focusing on accuracy they were told to go faster and vice versa.

Researchers in motor control distinguish between two types of feedback: inherent and augmented feedback (Schmidt & Lee, 2005). *Inherent* feedback is the feedback obtained from the activity itself, for instance, a golfer sees how the ball rolls over the green and then falls into the hole after he putted it. In the case of the bubble test, the sound and burst animation when a bubble was hit or the sound when it was missed provides inherent feedback. On the other hand, *augmented* feedback does not come from the activity itself. The golfer might for instance get verbal instructions from a pro after his put or count his score after a round. Similar to the other experiments mentioned above, such feedback is incorporated in the bubble test in the form of a summary screen that appears after a block of trials and gives feedback on the performance during the block.

4.3.3.1 Feedback on Acquisition Time and Error Rate

After each block of trials a summary screen appears that shows the hit rate ("Treffer"), average acquisition time ("Trefferzeit") and a comparison of the current acquisition time against the current acquisition time record in the form of a bar graph.

Treffer:
20/20 (100%)

Trefferzeit:
716 ms

Klasse!
So schnell warst Du noch nie!

Figure 27: Summary screen of the bubble test

Additionally, the result is "commented" by a graphical character as can be seen in Figure 27. The comments of the character depend on the performance of the participant: First of all, if the hit rate is lower than a defined threshold, the character suggests that the participant should try to be more precise. If the acquisition time was greater than the current record of the session, the character praises the participant by telling him that this was the fastest block yet (this is the case in Figure 27). If the record is not broken during the block, the acquisition time of the block is compared against the mean time of the previous blocks. If the comparison reveals that the acquisition time is below the mean time the application comments that the participant should continue in that manner. If it is above the mean time plus an additional safety threshold the character tells the participant that he is doing well in terms of accuracy and that he should try to speed up the movement a bit. Generally, feedback on the speed of the movement is not given on the first block, because the acquisition time record and mean times of previous blocks are not yet available.

Acquisition time and not mere movement time is given as performance feedback due to the following reason: Acquisition time is the total time that a selection takes. This means that it includes reaction time and movement time. Pre-tests of the application have shown that giving movement time as a feedback is critical because it does not reflect the intuitive understanding of performance of the participants. Participants felt that they performed well when they reacted as quickly as possible and not when they merely moved the pointer as quickly as possible.

4.3.3.2 Error Threshold

As described before, similar to Fitts' first tapping experiment (see section 3.1.3.1, p. 18) a threshold for the hit rate (e.g. 90%) can be defined in the bubble test application. If the mean hit rate of a block falls below that specified threshold, feedback is given by the program that pointing should be more accurate.

As described earlier in section 3.1.3 (p. 18) participants can usually trade-off accuracy against speed in a pointing task. By prescribing the hit rate the accuracy factor can be "fixed" leaving only the speed of the movement as a variable determinant of performance – provided that participants follow the recommendations of the comments given by the program. Of course accuracy cannot be confined completely in this way because users can still carry out the tapping task with more accuracy than is actually required (i.e. they *underutilize* the target), but the range of permissible target utilization, particularly overutilization, can be greatly reduced in this way. The reason why this is done is that, first of all, it assures that users cannot go astray and switch to a "spray-and-pray" shooting technique. This increases the homogeneity of results. Secondly, in this way participants can be given meaningful performance feedback in the form of movement or acquisition time. Without the accuracy constraint such a feedback would not give any information about the *performance* because a movement can be slow but accurate or fast but inaccurate. If it is known that the movement was reasonably accurate, information about its speed is useful. Also, movement time is now more meaningful as a measure for performance improvements over the course of the experiment and can be used for analysis of effects of practice. Furthermore, it is assumed that in this way the task is more closely related to a typical selection-through-pointing task in a graphical computer interface. The main goal of such tasks is typically to hit a button or other control element of the user interface and not to hit it as fast as possible because often a cost is associated with missing it (e.g. hitting the wrong button that is next to the target button).

4.4 Experiment II: Effects of Practice on Laserpointer Performance

In light of the results of the previous study and the literature review that was presented in section 4.2, an experiment was conducted to investigate the nature of practice effects. The experiment was carried out with four participants over a period of several days. The device that was examined in the study was the laserpointer interaction system described before. The main questions that the experiment should answer are: Is there an effect of practice on the pointing performance of the laserpointer? And if yes, what is the character of this effect?

4.4.1 Motivation and Overview

Existing pointing device experiments often compared the pointing performance of several devices against each other in a single session, usually lasting not more than one hour per participant so that participants did not become exhausted (e.g. Myers et al., 2002; A. Oh et al., 2002). Thus, the amount of practice per device was often strongly confined from the start, because too many experimental conditions must be squeezed into this single session. As was already shown in section 4.2.2.2 it is quite reasonable to assume that different devices exhibit different learning curves. Some devices might be easy to learn but only allow mediocre performance in the long run. Others may require some training up front before they can be used efficiently. Results obtained from studies that only operate on a small segment of these curves might deliver a strongly distorted view. In this experiment, participants were given considerably more time for practice. Instead of a single session, each participant worked with the device for five 30-45 min sessions which were held on five consecutive days. During each session six blocks of the bubble test (see previous section) were presented. It was therefore possible to compare the pointing performance across sessions to assess the long term performance gain and to examine performance trends across the blocks of a single session.

When measuring the effect of practice on a pointing task, several factors can contribute to the gain or loss of performance. Some are more transient like habituation to the stimulus situation, fatigue, or loss of concentration during a long session. Others are more permanent, like adopting a new strategy for holding the device to facilitate pointing. As already mentioned at the beginning of this chapter, such permanent effects are said to be due to *learning*. A technique that is commonly used by motor control researchers to separate permanent from transient effects is to use a transfer design that requires a task that is different to the one practiced during the experiment (Schmidt & Lee, 2005). In the experiment at hand, a classical onedirectional tapping test was used as this transfer task. The transfer task was carried out at the beginning and at the end of the experiment (session 1 and 5) to assess permanent performance gains of practicing pointing with the laserpointer.

Because of the effort involved in conducting such long-term experiments, the number of participants that took part in the experiment was significantly smaller than in experiment I. Four participants took part in this experiment compared to 16 in experiment I. Nevertheless, the expenditure of time and cost was even greater for the experiment at hand (16 separate sessions were held in experiment I, 20 sessions were held in experiment II). Yet, the experiment can deliver first trends and a basis to accumulate an even more extensive body of data, for instance, by carrying out more experimental sessions in the same fashion to verify these trends.

4.4.2 Hypotheses

Based on the previous experimental experience and the literature review of laserpointer systems and experiments presented earlier, the following hypotheses were formed and tested in the experiment. For these hypotheses the term “performance” refers to the speed and accuracy of a pointing movement as discussed in section 3:

H1: Pointing performance of the laserpointer improves with practice.
This hypothesis pertains to the question whether practice has an effect on performance *at all*. Furthermore, the magnitude of a performance gain is of interest to compare it with the magnitude of differences *between* devices which are commonly obtained in pointing device studies. For instance, in experiment I, the performance difference between the laserpointer and the mouse was a little less than 0.5 bits/s. However, in experiment I, practice only affected the performance of the laserpointer, presumably because participants had never used a similar device before. On the other hand practice seemed to have no grave effect on the mouse, because participants were much more familiar with that device. If the effect of practice is large, maybe even larger than the difference between the laserpointer and the mouse, then the results obtained in the previous experiment must be interpreted in a new light. The statement that the performance of the laserpointer is generally worse than that of the mouse should then be reconsidered.

H2: Improvements of pointing performance of the laserpointer follow a learning curve.
It is assumed that the effect of practice is strongest for the first few blocks and sessions and gradually diminishes with further practice. As was discussed in section 4.2.2.2 the nature of performance gains due to practice probably follows a learning or performance curve that can be modeled with a power function. Since a lot of blocks and trials are scheduled in the experiment at hand, a simulation can then be made to check if the model obtained from using just a fraction of the data is similar to the model obtained from using all the data of the experiment. This could provide valuable insight with respect to the reliability of trend predictions based on power law models in comparative evaluations that cannot use extended amounts of practice trials as in the current experiment.

H3: Performance improvements of the laserpointer are permanent.
As discussed in the previous section, in order to assess whether practicing with the laserpointer really leads to learning, the onedirectional task at the beginning and end of the experiment serve as a transfer task. It is assumed that laserpointer performance also improves in this task due to learning. Additionally, participants carry out the task with a mouse as a reference. It is assumed that the performance of the mouse does not change much between the two sessions because no practice was scheduled for the mouse in the experiment.

H4: Fitts' Law is an adequate model for the bubble test
The last hypothesis pertains to one of the tests. It is assumed that Fitts' Law can be used as a model for the pointing performance in the bubble test. Assessing the model fit can be seen as a quality criterion that informs about the suitability of the test for pointing device studies. If model fit would be bad, this would indicate that either the general procedure of the test or some part of the application, such as the recording of the movement, was flawed.

4.4.3 Participants

As can be seen in Table 10, four participants took part in the study. All participants were available on five consecutive days during the experiment and none of the participants had any previous experience with a laserpointer interaction system or anything similar. All participants received 25 € as a compensation after the last session.

Table 10: Participants

ID	Age	Gender	Student	Faculty	Computer Use	Dom. Hand
1	19	F	Yes	Humanities	1-2	Right
2	23	F	Yes	Humanities	1-2	Right
3	19	F	Yes	Physics	> 3	Right
4	22	F	Yes	Humanities	1-2	Right

Computer use given in hours per day

4.4.4 Apparatus

The experiment was carried out at the University of Konstanz in the room where the Powerwall is located. Participants were standing in front and in the middle of the 5.20 × 2.15 m display wall at a marked position with an approximate distance of 2.5 m to the screen. The frontal ceiling lights were turned off in the room to diminish distracting reflections on the display surface.

Figure 20 on page 46 shows the laserpointer that was used in the experiment. The laserpointer was contained in a metal casing with an acrylic tip and end that were slightly illuminated. Three buttons were set into the metal casing and the leftmost button was used as a selection trigger in the test applications. A cable was attached to the end of the laserpointer to reliably transmit the button signals. The cable was sufficiently long and flexible so that use of the pointer was not affected.

In the mouse condition in the first and fifth session, a Logitech MX Laser Cordless Mouse was employed. The mouse's gain adjustment in Windows Server 2003 was set to a medium condition so that participants were able to move across the complete screen without clutching. A 1.10 m lecture desk provided a stationary surface for using the mouse. It was moved to the marked location in front of the Powerwall prior to the mouse task.

Two programs were used in the experiment to display the stimuli and collect data. The ieval2 program, a successor of the software used for stimulus presentation and recording of movement times in experiment I, which was completely re-written by the author to facilitate data analysis and experiment preparation, ran on the main computer connected to the Powerwall display in full screen mode with the maximum resolution of 4640 × 1920 pixels. The standard Windows pointer icon was used in this test. Additionally, the bubble test described before was used, which ran on a laptop computer that was connected to a chromium rendering cluster consisting of eight computers, one for each projector of the Powerwall. The rendering cluster delivered a coherent image of the test scene with the maximum resolution of the Powerwall. This setup was required to achieve a smooth presentation of the test's graphics. The custom crosshair pointer of the bubble test application was used as a pointer in this case.

The laserpointer signal and button presses were collected with a management program which ran on the main computer of the Powerwall. When the ieval2 program was used during the experiment, the management software automatically positioned the Windows pointer and propagated click events to the operating system event queue. When the rendering cluster was used, the pointer position and button states were directly sent to the bubble test application via UDP by the management software. Each packet contained the location of the pointer in normalized coordinates as well as a value to indicate the button state.

Demographic data was collected in a pre-test questionnaire. Additionally, an ISO 9241-9 questionnaire was used with an independent rating scale to assess the perceived qualities of laserpointer use. The questionnaires can be found in the appendix.

4.4.5 Tasks

The primary task used during the experiment was the discrete multidirectional tapping task of the bubble test that was already described in section 4.3. The second task was the standard continuous onedirectional tapping task which is also proposed by the ISO standard (see section 3.3) and served as a transfer task to assess the permanence of practice effects.

Figure 28 shows an overview of the sessions, blocks, and tasks scheduled in each session. For instance, in the first session S1, six blocks of the bubble test were presented (B1-B6). After that, the independent ISO questionnaire was presented. Then, two blocks of the onedirectional task were carried out for the laserpointer (OL) and two blocks of the onedirectional task were carried out for the mouse (OM). In sessions 2 to 4, only the bubble test was used. In session 5, both tests were again presented to assess the transfer of learning.

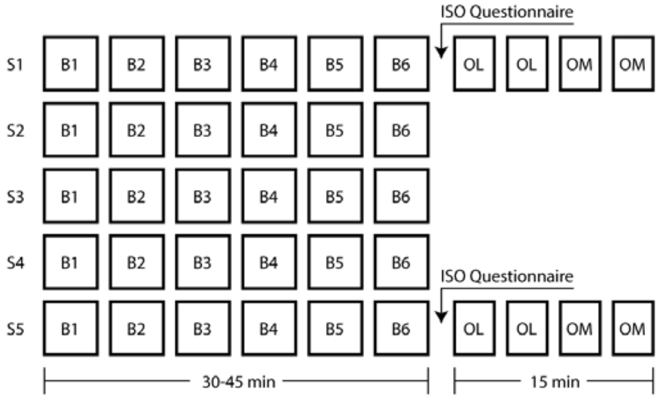


Figure 28: Blocks and Sessions

4.4.5.1 Trials and Blocks

During session one and five at the beginning and end of the experiment, participants carried out a total of 1260 trials. During the shorter sessions on day two to four, each participant carried out 756 trials. Thus, each participant carried out 4788 trials during the whole experiment, not including warm-up and free bubble trials. The bubble test consisted of six blocks with 126 trials each. 14 repetitions per amplitude-width condition were used for both, the bubble test and the onedirectional test as can be seen in Table 11.

Table 11: Trial and Block Counts

Test		Trials
Bubble Test (block)	3A ×3W × 14	126
Bubble Test (whole)	6 blocks	756
Onedirectional (block)	3A × 3W × 14	126
Onedirectional (whole)	4 blocks (laser/mouse)	504
Total		**1260**

4.4.5.2 Ranges of Difficulty

The combinations of amplitude and width were chosen so that a large range of different difficulties was obtained (from easy to hard). Since different amplitude and width combinations could lead to similar ID values, amplitudes and widths were selected so that the range of difficulty was covered evenly. For the bubble test, conditions were slightly easier than for the onedirectional task since a) not the complete width of the screen could be used as amplitude because it was a multidirectional test and b) to not make it too hard for participants to carry out the task over a prolonged amount of time the smallest target width was set to 80 px instead of 40 px. The exact values for amplitudes A and widths W can be seen in Table 12.

Table 12: Test Ranges of Difficulty

Bubble Test			Onedirectional Test		
A	W	ID	A	W	ID
400	160	1.8	400	160	1.8
	120	2.1		80	2.6
	80	2.6		40	3.5
1200	160	3.1	1200	160	3.1
	120	3.5		80	4.0
	80	4.0		40	5.0
1800	160	3.6	3200	160	4.4
	120	4.0		80	5.4
	80	4.6		40	6.3

A and W given in pixels, ID in bits

4.4.6 Experimental Design

The main factor of interest for this study was the amount of practice given to the participants over the course of the one week period. Mouse trials were used at the end and at the beginning of the experiment to test the experimental apparatus and as a means of assessing transfer of learning of the laserpointer. Therefore this factor was not counterbalanced because the difference between mouse and laserpointer was not in the focus of the experiment.

The dependent variables for the bubble test and the discrete onedirectional test were

- Movement time, MT
- Index of performance, IP

- Effective index of performance, IP_e

The index of performance and effective index of performance measures were calculated according to the requirements of the ISO standard 9241-9:

$$ID_e = \log_2 \left({}^{A}/_{W_e} + 1 \right)$$

$$W_e = 4.133 \times \sigma_x$$

$$IP_e = ID_e / MT$$

The effective index of difficulty ID_e is calculated from the movement amplitude and the effective width W_e. The effective width is derived from the movement endpoint standard deviations. The method for calculating this standard deviation is discussed in section 3.4.2. Finally, the effective index of performance IP_e is calculated from the effective index of difficulty and the mean movement time of the movements of an amplitude-width condition.

Error rate or hit rate was not a dependent measure in the bubble test because it was constrained by the hit rate threshold as explained in section 4.3. The hit rate threshold was set to 90%, a value that was tested beforehand in a pre-test session to assure that it was not too hard to obtain. As mentioned before, constraining the hit or error rate served the purpose of making movement time more meaningful for data analysis and feedback. Since no such restrictions were made in the regular onedirectional test, error rate was a dependent measure in this case.

4.4.7 Procedure

As mentioned earlier the experiment consisted of five sessions held on consecutive days. The first and fifth sessions took around one hour and session two to four took around 30 to 45 minutes. In the first session, participants first read a one page introduction about the goals and procedure of the experiment and then filled out a pre-test questionnaire. After that, the experimenter gave the laserpointer to the participant and explained its function. Participants were asked to stand on a marked positions and were given a short period of time of one to three minutes to test the pointer in the free bubble phase of the bubble test. Then the experimenter took the laserpointer and switched to a small demo mode, to explain how the bubble test worked and what the feedback at the end of each block meant. Then the laserpointer was handed over to the participant again and the software was switched to the actual task. Before starting with the task, subjects were told to work “as fast and accurately as possible”. Figure 29 shows a participant in front of the Powerwall with the the bubble test.

After each block, participants were shown the feedback screen of the bubble test which informed them about the hit rate, mean acquisition time, and the mean acquisition time of the current block compared to the their record. The feedback screen was switched to the next block automatically after 30 s. Subjects were told that they could rest as long as they wished after a block and to commence with the task they should simply walk up to the marked position and use the laserpointer to point at the starting position in the middle of the screen.

Figure 29: Participant Carrying out the Bubble Test

After all blocks of the bubble test were completed, the ISO questionnaire was handed over to the participants. While participants filled out the questionnaire, the experimenter switched the task setting to the onedirectional task.

Subjects were once more told to stand at the marked position. The experimenter started a demo task and explained it to the participant. After that, he handed over the laserpointer and a short, 16 trial warm-up block was presented. After that, recording of the trials began and participants were again told to work "as fast and accurately as possible". The task consisted of two blocks and participants were allowed to rest between blocks. After these two blocks, the participants handed over the laserpointer and the experimenter moved the desk and the mouse to the marked position in front of the display. Again, a short warm-up block was presented before the two recorded blocks of the onedirectional task.

In the second, third, and forth session, the procedure was as follows: After the experimenter greeted the participants, they were told to stand at the marked position in front of the screen and were handed the laserpointer. A short period of the free bubble task was presented to give them time to get accustomed to the environment again. Then six blocks of the bubble test were presented in the same fashion as in the first session. No questionnaires or onedirectional task were required in these sessions, therefore the time needed was shorter, and a session took around 30 to 45 minutes, depending on the speed of the participant and whether the participant used the pauses in between blocks.

The last session on day five was essentially a replication of session one, except that no pre-test questionnaire was given to the participants. After the last session participants were given the compensation for coming to the test.

4.4.8 Results and Discussion

The results reported here are based on the data of the four participants that were available for the test. The small number of participants prohibits the verification of prerequisites of some statistical tests, for instance, verifying that data was distributed normally or sphericity in case of ANOVA procedures. Nevertheless, results are reported as if these prerequisites were fulfilled so that this report can serve as a template for further analyses. Thus, the statistical results should be considered trends that require further verification by conducting additional experimental sessions. The results are reported and interpreted in light of the hypotheses formed prior to the experiment along with other findings of interest.

4.4.8.1 Treatment of Outliers

Similar to the procedures of experiment I, trial outliers were eliminated before further analyses were conducted. Such outliers resulted for example from accidental double clicks or other anomalies during a trial. For the bubble test, outliers were eliminated as follows: First, the mean movement time and the movement time standard deviation per participant, session, and amplitude-width condition was calculated. Trials were removed that were not within a range of ± 3 standard deviations around the movement time mean. These were approximately 267 trials of 15120 (≈1.7%). For the onedirectional task outliers were eliminated by calculating the mean movement time and the movement time standard deviation per participant, session, device, and amplitude-width condition. Trials were removed that were not within a range of ± 3 standard deviations around the movement time mean. These were approximately 28 trials from 4032 (<1%).

4.4.8.2 H4: Fitts Law Model Fit

Before considering the results of the test in terms of performance values, the model fit of Fitts' Law must be verified. If the model fit is good this means that the dependent measures IP and IP_e can be used as a combined measure of the speed and accuracy of the movements.

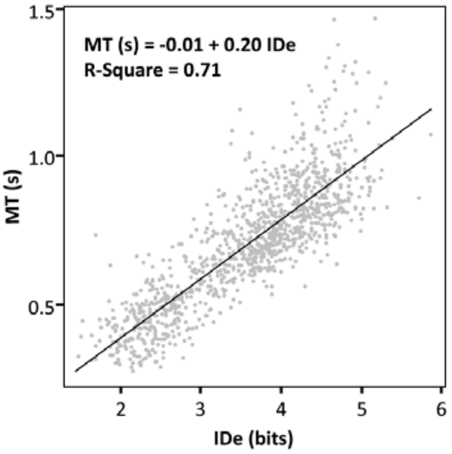


Figure 30: Bubble test model fit for combined results

To analyze the fit of the model for the bubble test, a regression analysis was carried out on the combined data from all participants and every session. The model obtained showed high correlation between the effective index of performance ID_e and MT. As can be seen in Figure 30 and Table 13 the model explains 71% of the variance. For the index of performance ID, the correlation and explained variance was slightly higher. Separate analysis per session and participant showed that correlations were around 90% for ID_e and well above 90% with explained variances between 80-90% for ID. These results are similar to those obtained in Fitts' Law studies by other researchers (e.g. Isokoski & Raisamo, 2002). Additionally, analysis with the mean MT for each ID instead of the mean MT for each ID and participant showed very high correlation as can be seen in the last row of Table 13. These results are also consistent to the findings of other Fitts' Law studies that reported mean ID regression results (e.g. Cockburn & Brock, 2006; Zhai et al., 2003).

Table 13: Regression results

		Bubble		Onedir.	
Measure	Device	R	R^2	R	R^2
ID	Laser	.87	.75	.94	.88
	Mouse	–	–	.94	.89
ID_e	Laser	.84	.71	.93	.87
	Mouse	–	–	.94	.89
ID (mean)	Laser	.99	.99	.99	.98
	Mouse	–	–	.98	.95

As expected, the model fit for the onedirectional results was also very good. A separate analysis for each device of the results of all participants for both sessions showed high correlation for ID and slightly lower correlation for ID_e as can be seen in Table 13. Regression analysis with the MT means for each ID instead of the means per ID and participant showed even higher correlation.

These results confirm the assumption made in hypothesis H4. The model fit seems to be fairly good, which indicates that Fitts' Law is an adequate model for the relationship of target amplitude and width and the movement time required in the bubble task as well as in the onedirectional task.

4.4.8.3 H1: Improvement of Pointing Performance

As can be seen in Figure 31 and Table 14, the mean index of performance and mean effective index of performance scores increased from session to session. The mean effective index of performance increased from around 4.5 bits/s to around 5.6-5.7 bits/s. Similarly, the index of performance increased from 4 bits/s to a little over 5 bits/s in session four and dropping a bit below 5 bits/s in session five.

Table 14: Bubble test session means

	S1	S2	S3	S4	S5
IP$_e$	4.5 (.13)	5.1 (.43)	5.4 (.15)	5.7 (.13)	5.6 (.22)
IP	4.0 (.24)	4.6 (.50)	4.9 (.30)	5.0 (.42)	4.9 (.46)
MT	825 (48)	710 (71)	659 (37)	642 (47)	662 (56)
Hit Rate	.96 (.03)	.96 (.04)	.96 (.04)	.96 (.04)	.97 (.03)

Values are in bits/s for IP/IP$_e$, ms for MT
Values in braces are standard deviations

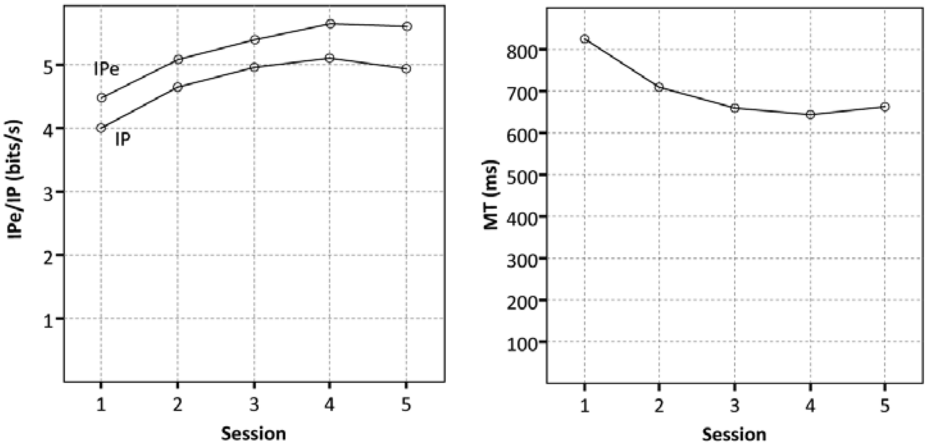


Figure 31: Session means for IP, IPe, and MT

A repeated measures ANOVA for the session factor was carried out which revealed that the session effect was significant ($p < 0.01$) for all measures (IP$_e$, IP, and MT) except for the Hit Rate ($F_{1,3} = 1.776$, $p = .198$). These results generally confirm hypothesis H1. It seems that practice had a major effect on the laserpointer performance. The performance increase from the first to the last session was around 1 bit/s for IP$_e$ and IP.

Interestingly the effective index of performance IP$_e$ was constantly 0.5 bits/s higher than index of performance IP which indicates that participants consistently underutilized the targets and were very accurate. A comparison of the effective width W_e against the actual width of the targets confirmed this assumption. As can be seen in Table 15, W_e was always lower than the given target width except for the largest amplitude and smallest target size.

Table 15: Target utilization for the bubble test

A	W	W$_e$	Utilization
400	80	64.8	.81
	120	90.7	.76
	160	114.6	.72
1200	80	67.7	.85
	120	92.8	.77
	160	118.1	.74
1800	80	81.5	1.02
	120	106.4	.89
	160	130.8	.82

Sizes given in pixels

Also analysis of hit rate showed that participants were very accurate. In fact, only one of the four participants reached the accuracy threshold of 90%. As can be seen in Figure 32, this happened only once in session three. This means that the accuracy feedback was only given once and it seems that the nature of the task itself promoted accuracy.

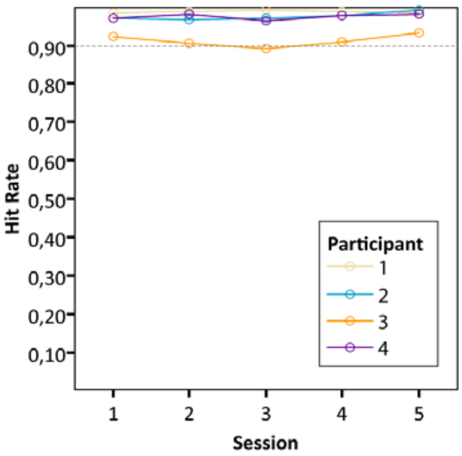


Figure 32: Hit rates per participant

This also indicates that accuracy was fairly stable throughout the experiment and suggests that the main factor influencing performance was movement time. Indeed a significant decrease of around 150 ms in movement time was found from around 800 ms in session 1 to around 650 ms in sessions 3 through 5.

The performance and movement time of each participant and the mean performance and movement time (drawn in black) for each block of each session is shown in Figure 33 and Figure 34. On the abscissa session and block are encoded as a two-digit number. The first digit of that number denotes the session, the second the block. The ending of one session and the beginning of another session is marked with vertical lines. The data showed the general pattern of improvement over sessions that was also discernible when only regarding session means. However, the figure also makes obvious how noisy and irregular the performance across each session and of each participant was. A smooth learning curve is rarely found for performance curves across blocks (see M1 in Figure 33). Instead, sharp drops in performance are visible for instance in the middle of sessions three and four (see M2, M3 in Figure 34). The reason for these drops might for instance be fatigue.

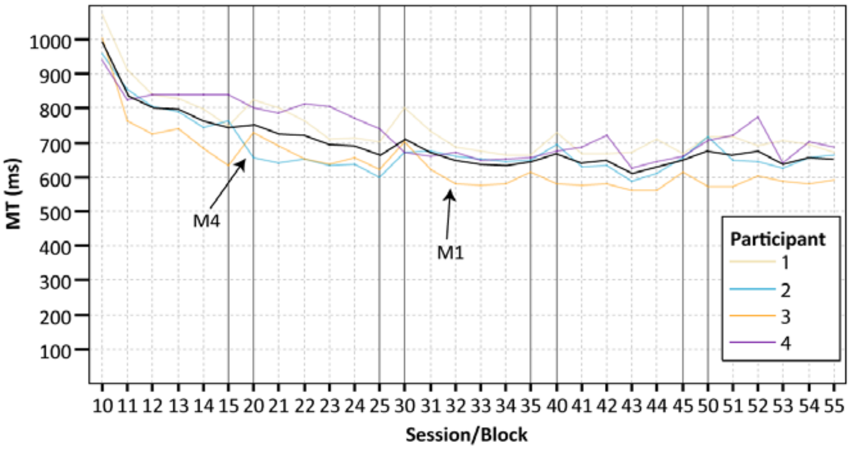


Figure 33: Movement time per block and session

It was generally assumed that the performance at the beginning of a session was worse than the performance at the end of the previous session because participants first had to warm-up to achieve a similar level of performance. This difference corresponds to the slope of the line between the marked vertical lines between sessions. With regard to these slopes it seems reasonable to assume that this was the case, except for some instances where participants showed tremendous performance gains between sessions (M4, M5).

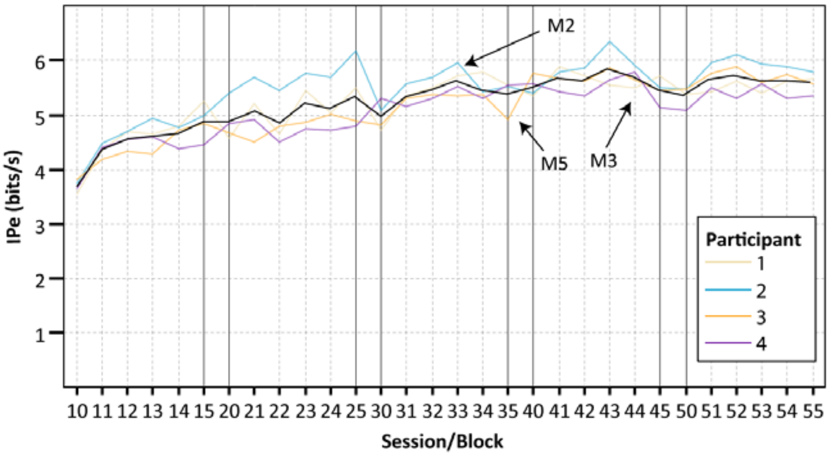


Figure 34: Effective index of performance per block and session

Judging from the shape of the performance and movement time curve across sessions and across blocks it seems that most of the performance gains occurred during session one followed by gradual increases until session four. The increase from the first block of the first session to the last block of this session was 4.88 – 3.70 = 1.18 bits/s (≈ 30%). The increase from the last block of session one until peak performance was 5.85 – 4.88 = 0.97 bits/s (≈ 20%). Almost half of this further increase occurred in session two. The difference between the last blocks of session one and two was 5.35 – 4.88 = 0.47 bits/s. The performance peak was in the middle of session four (5.85 bits/s), and no further increases in performance were made in session 5. Analysis of within-subjects contrasts between sessions using Helmert contrasts showed that the first session differed significantly from all other sessions ($p < 0.01$). As can be seen in Table 16, the contrasts of session 2 and 3 are close to significant at the .05 level and could possibly become significant if data is available from more participants.

Table 16: Helmert contrasts for session means

	1 vs. later	2 vs. later	3 vs. later	4 vs. later
F	291.7	9.4	6.716	0.754
p	.000	.055	.081	.449

4.4.8.4 H2: Performance Curve Fit

The block means of all sessions were fitted against a power curve of the form $M_N = \beta N^a$ where M_N denotes the value of the measure on the Nth block, β is a constant term, and a represents the rate of improvement. For the effective index of performance the fit of the power curve was fairly high explaining 92% of variance ($\beta = 3.918$, $\alpha = 0.115$) and thus confirming hypothesis H2. For movement time, the explained variance was slightly lower ($R^2 = 0.884$, $\beta = 927$, $\alpha = -0.115$). The fitted curves can be seen in Figure 35.

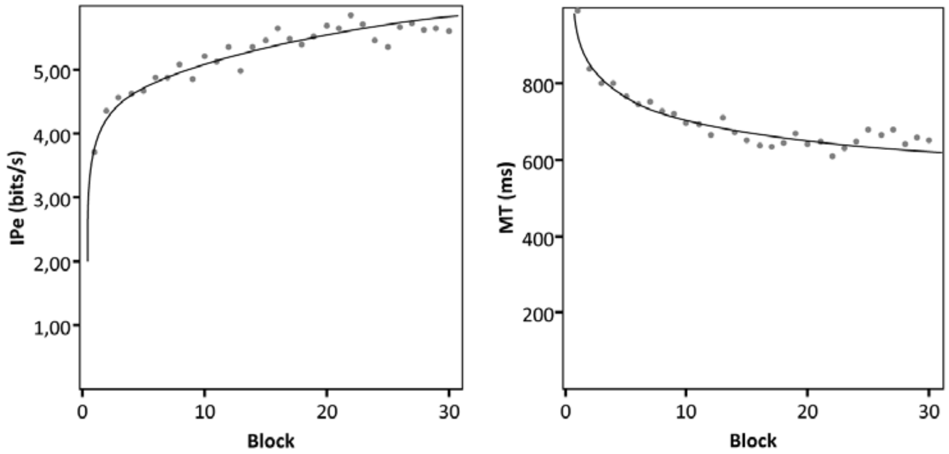


Figure 35: IP_e and MT performance curves

In a similar fashion performance curves were obtained for only a subset of blocks, these include sessions 1-5, 1-4, 1-3, 1-2, and 1, to assess the reliability of the predictions of curves obtained with less practice. The explained variance for all curves is > 90%.

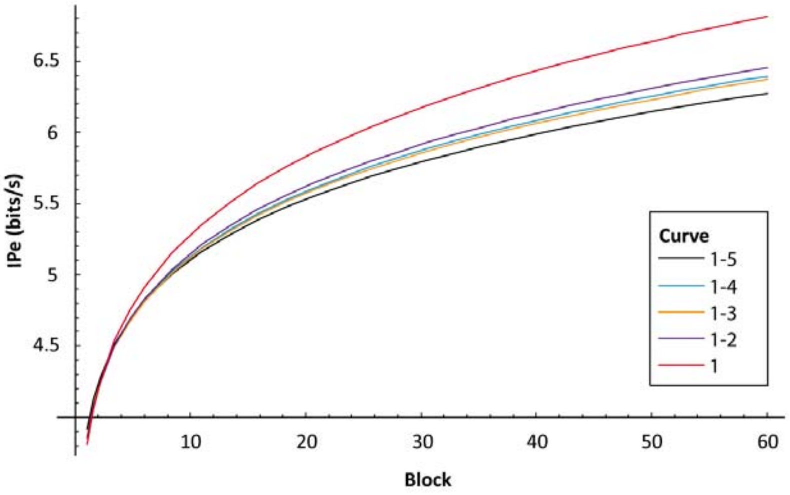


Figure 36: Performance curves for subsets of the experimental data

One can see that the models obtained for more than one session are very similar while the model sticks out that was obtained for the blocks of the first session only. This model seems to lead to particularly progressive predictions, questioning whether data from only few blocks could be used to obtain a good performance curve model. Of course, it would be interesting to see if the same pattern would be discernible if more data was available from additional participants.

4.4.8.5 H3: Permanence of Performance Gain

The results of the bubble test showed that practice lead to a significant performance improvement. The question remains whether this performance gain can be considered permanent or whether it was just a transient phenomenon which might be due to habituation to the task environment. In order to answer this question, a transfer task was carried out. As can be seen in Figure 37 and Table 17, the performance of the laserpointer increased from the first to the second session in this transfer task.

Table 17: Onedirectional session and device results

Measure	Device	S1	S2
IPe	Laser	4.1 (.13)	4.4 (.23)
	Mouse	4.7 (.23)	4.8 (.12)
IP	Laser	4.3 (.25)	4.7 (.44)
	Mouse	4.9 (.43)	4.9 (.42)
MT	Laser	934 (54)	851 (93)
	Mouse	821 (63)	817 (70)
Hit Rate	Laser	.87 (.04)	.90 (.04)
	Mouse	.92 (.06)	.93 (.04)

Values are in bits/s for IP/IP$_e$, ms for MT
Values in braces are standard deviations

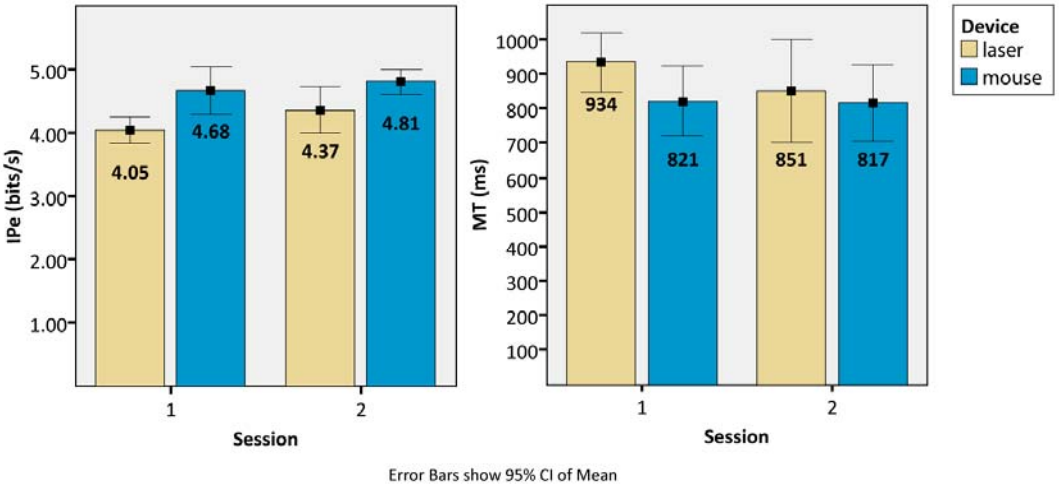


Figure 37: Onedirectional session and device results (left IP, right MT)

Repeated measures ANOVA was carried out for the factors device (laser/mouse) and session (1/2). The results are shown in Table 18. Significant main effects at the .05 level of the device factor were found for the IP_e measure and the hit rate. No significant main effects were found for the session factor. These results indicate that still a general performance difference existed between the laserpointer and the mouse and the findings are thus consistent to those of experiment I.

Furthermore, a significant device × session interaction existed for the IP_e, IP, and MT measure. Analysis of simple main effects (see Table 19) of the session factor on each device level showed significant effects for the laser for the IP and MT measure and a result close to significant at the .05 level for the IP_e measure. The effect on the mouse was insignificant for all measures as can be seen in Table 19. This indicates that there actually was a difference between the first and the last session of the transfer task for the laserpointer which means that some transfer of learning from the practice task took place. The effect on the movement time seemed to be much stronger than the effect on the effective performance measure IP_e. The movement time of the laser dropped around 80 ms in the second session whereas the improvement of the effective index of performance was only small (around 0.3 bits/s). This poses the question why performance did not increase in the same amount. Closer analysis of accuracy showed that the effective width W_e increased from session one to two. This means that participants were faster but more inaccurate with the laserpointer in session two.

Table 18: Onedirectional ANOVA results

	Measure	F	p	p. Eta^2
Device	IP$_e$	11.468	**.043**	.793
	IP	5.167	.108	.633
	MT	5.092	.109	.629
	Hit Rate	29.299	.012	.907
Session	IP$_e$	6.196	.089	.674
	IP	6.313	.087	.678
	MT	6.160	.089	.672
	Hit Rate	4.441	.126	.597
Device × Session	IP$_e$	18.837	**.023**	.863
	IP	36.686	**.009**	.924
	MT	39.054	**.008**	.929
	Hit Rate	0.888	.228	.228

Table 19: Simple effects of session on device for the onedirectional test

Device	Measure	F	p	p. Eta^2
Laser	IP$_e$	8.54	.061	.740
	IP	16.63	**.027**	.847
	MT	15.89	**.028**	.841
Mouse	IP$_e$	3.22	.171	.517
	IP	0.09	.786	.029
	MT	0.06	.825	.019

In summary, it can be said that some transfer occurred and thus a relatively permanent increase in performance. However, the result was not as clear-cut as expected and also the size of the increase in performance was only small. This could be due to the fact that either some portion of the performance gain during the practice task was due to learning of device control and the other due to learning of the task. Upon switching to the onedirectional task only the device control portion was then transferred. Another cause could be the difference between the pointing character of the continuous onedirectional test and the discrete bubble test. This issue was already touched in section 3.4.1 (p. 34). Maybe the difference between the strategies required by each of the tasks was too large. On the one hand participants were carrying out discrete, aimed movements in the bubble test, and on the other hand, a good strategy was required for pendulous motions in the onedirectional task. In any case, the results give reason to further investigate the procedures and findings of psychologists and motor control researchers such as Schmidt and Lee (2005) to determine what makes up a good transfer task in order to also make an assessment of the magnitude of the performance increase of the transfer task.

4.4.8.6 Comparison with Experiment I

As already explained in section 4.3, the absolute results of the bubble test cannot be compared with results obtained from continuous tapping experiments as the one used in experiment I. However, the results of the onedirectional task that served as a transfer task should be comparable to the results of the previous experiment. Astonishingly, the absolute results were much higher than those obtained in experiment I. For instance, the IP_e value for the mouse was around 3.96 bits/s (at 3 m) in experiment I and 4.75 bits/s in experiment II. Similarly, the performance of the laserpointer was 3.52 bits/s (at 3 m) in experiment I and 4.21 bits/s in experiment II. It is very unlikely that this difference is due to chance – although both absolute values of the mouse are within the range of 3.7 to 4.9 bits/s of values commonly reported for the performance of the mouse in other studies (Soukoreff & MacKenzie, 2004). Possibly the difference resulted from an error in one of the applications. Therefore a quick empirical comparison with a mouse on a desktop computer with the same amplitude-width combinations was made to compare the re-written version of the testing software (ieval2) against the software (ieval1) which was originally used in the first experiment. The results of this test showed that there was almost no difference between the two (ieval1: 4.8 bits/s, ieval2: 4.7 bits/s). Therefore an error in the applications can be ruled out as a possible cause for the difference. Although the relative difference between laserpointer and mouse was almost identical to the one in experiment I (around 88% in both experiments), the large discrepancy between absolute values could not be explained yet.

4.4.8.7 Comfort Assessment (Questionnaires)

No hypotheses were formed prior to the experiment with respect to the results of the ISO questionnaires used in the experiment. The questionnaire used in the experiment can be found in the appendix. As can be seen in Figure 38, almost no differences exist in the rating of the general comfort and overall operation of the input device between the beginning and the end of the experiment for all participants. It seems that general comfort and overall operation are largely unaffected by practice.

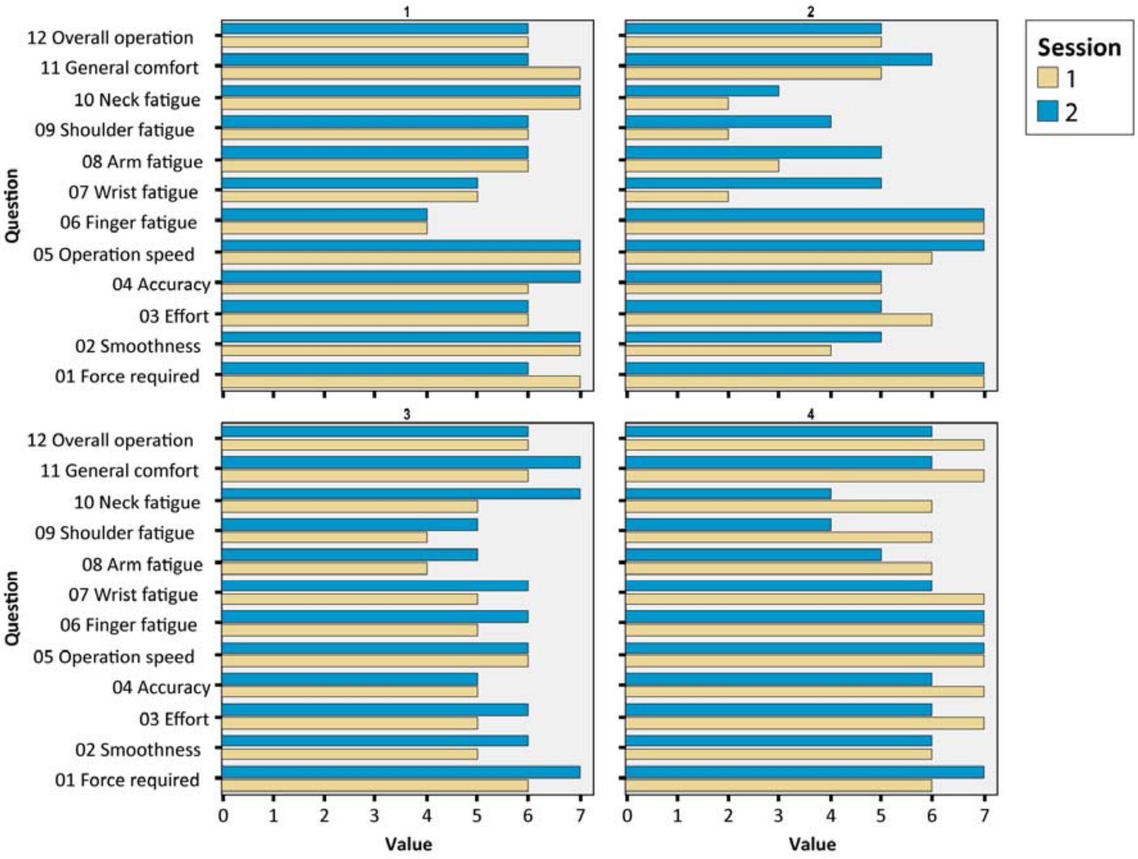


Figure 38: ISO questionnaire results for each participant and session

Larger differences exist between participants for the other ratings. Of course, data from more participants would be required so that reasonable inferences can be made. It would be interesting to see if for instance, accuracy or speed judgments differed after longer practice session to assess if the gains in pointing speed and precision (i.e. the measured IP_e/MT values) are also perceived by the user.

4.4.9 Conclusion & Outlook

4.4.9.1 Implications for Laserpointer Comparison

First of all, the results show that the methodological approach in experiment II was reasonable to analyze the effect of practice on laserpointer performance. The largest increase of performance occurred in session one of the experiment (around 1 bits/s and 30%). However, performance increased by another 1 bits/s until the middle of session four. 756 trials were carried out by each participant until the end of session one and around 2600 trials were carried out by each participant until the middle of session four (3 × 756 + 756/2). In contrast to that, only 416 trials were carried out per participant in each of the four conditions (laser/mouse, 3/6 m) of experiment I. This means that only a fraction of the performance increase was considered in experiment I. Bearing in mind that the difference between the laserpointer and mouse in experiment I was only 0.44 bits/s (3 m) the increase of 1 bits/s after session one of experiment II seems to be highly relevant. Since the performance of the mouse seems to be unaffected by practice, as the results of experiment I suggest, these findings motivate another comparison of both devices to find out if the laserpointer catches up with the mouse if more practice trials are scheduled. The question is whether a practicable compromise can be found between the number of participants and the amount of practice trials.

With respect to the extent of performance increases between the sessions in experiment II, it is suggested to schedule at least around 1500 trials per device. This corresponds to the amount of trials of session one and two where the performance increase was largest. Judging from the time required for one session in experiment II, a comparative evaluation with the laserpointer and the mouse and 1500 trials per device could be carried out in about two to two and a half hours of testing. This does not include pauses, questionnaires, delays due to technical problems, etc. Of course, this imposes the question of how to schedule these trials. For instance, splitting up the experiment in two sessions or carrying out all of the trials in a single session with several pauses. A single session is much more practical from the experimenters point of view, because of the reduced amount of time required for preparation of the experiment (15-30 min. per session were required in experiment II in order to set-up the display, device, computers, etc. before each session). The problem with such a design is that it is likely that later trials will be affected by fatigue. Certainly, testing fatigue is also important to assess the usability of a device. But this cannot be done if one wants to rate practiced performance at the same time. Annex A of the ISO 9241-9 recommends that a "(...) test should not exceed 4 h per subject per day (...)" with five minute breaks scheduled each hour. Judging from the author's own experience, from the experience of other researchers (e.g. J. Y. Oh & Stuerzlinger, 2002), and from the comments made by participants during experiment I, even only one hour of testing could be strenuous, particularly if a novel and unfamiliar device is tested. In order to assess practiced performance it is therefore suggested to schedule separate sessions on consecutive days. In each session both devices should be used. First, 750 trials for device A (e.g. the laserpointer), followed by a short break and 750 trials for device B (e.g. the mouse). The sequence of factors should be counterbalanced between subjects to control for asymmetrical transfer of learning strategies. The results for the last blocks of the second session can then be used to make comparisons of practiced performance. Moreover, a reasonable amount of block means would then be available to carry out an analysis of performance curves. This is discussed next.

4.4.9.2 Use of Performance Curves

The experiment demonstrated that pointing performance increases follow a learning or performance curve and that relatively good models for such curves can be obtained from only a fraction of the data. In section 4.2.2.2 it was shown how models obtained from learning curves could be used to compare the ease of learning of two devices. It is therefore proposed that analyses of performance curves should generally be carried out in further experiments. Comparison of the performance curves of two devices could give rise to carry out more selective long-term experiments with fewer participants to find out if results obtained in the long run show a totally different picture compared to the original experiment. For instance, if two devices or variants of the same device are compared in an experiment, the results might suggest that the index of performance is greater for device A than for device B. Additional information about this performance difference can be gained from comparing the learning coefficients of the learning curves obtained for both devices: the α term of $T_N = T_1 N^a$. If the learning coefficient of device B is higher than that of A this means that more learning occurred for device B during the experiment. If the difference between the coefficients is large and the overall performance difference between the devices is small this could mean that device B could also “catch up” on the performance of A. This prediction from performance curve analysis could then be verified in a long-term experiment with fewer participants.

4.4.9.3 Practice and Learning

A question that could not be answered satisfactory by the experiment is that of the practical relevance of the practice effect. In order to distinguish between transient and permanent effects of learning, a transfer task was used in the experiment. It was hypothesized that permanent changes in performance would also influence the performance in the transfer task and that this transfer could be used as an indicator of the amount of learning that would occurre in a non-experimental setting. Results showed that the increases in performance in the transfer task were significant. However, the increase was only small. It was already discussed that this could also be due to the characteristic of the transfer task chosen and that a more thorough investigation of procedures in motor control and motor learning research is advisable to appropriately revise this task.

4.4.9.4 The Bubble Test

For the experiment, a new testing application, the bubble test, was developed. The application implemented a discrete multidirectional tapping test and a procedure to separate reaction from movement time during the measurements made for each trial. Additionally, several feedback mechanisms were employed to motivate participants to perform well when carrying out the tapping task. Regression tests showed that the application is useable to investigate pointing performance. Moreover, participants liked the game-like atmosphere of the test and mentioned that it was a “fun experience”. Improvements of the bubble test could be made with respect to the measurement of movement time. The measurement method described in section 4.3.2.3 “swallows” some of the movement time. An idea that could remove this issue would be to require another click on the starting position after a target appeared, before the selection of the target was carried out. In this way, movement time could be measured between the click on the starting position and the click on the target.

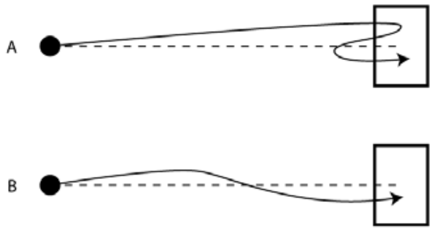


Figure 39: Refined accuracy measures: target re-entry (A), task axis crossing (B), adapted from (MacKenzie et al., 2001)

Further improvements of the application could be made by implementing the recording of refined measures as they are proposed by MacKenzie et al. (MacKenzie et al., 2001). For instance, the amount of target re-entry, that is, the number of times the pointer moves into the target, or task axis crossing, that is, the number of crosses of the ideal path along the way to the target. These measures could help to elucidate the causes of performance differences between devices and possibly aid in the development of novel techniques to remove these causes.

4.4.9.5 Analysis of Micromovements

Further information about the causes of performance differences could be gained from the analysis of the "movement microstructure" (Mithal & Douglas, 1996), for instance, the path of the pointer during a trial, its velocity, and acceleration. In an analysis of an isometric joystick and a mouse, Mithal and Douglas revealed that the pointing performance of the isometric joystick was much lower than that of the mouse. Closer analysis showed that the movement path of the joystick was much more affected by tremor, causing involuntary changes in the velocity of the pointer. These velocity changes made it difficult to stop the pointer at the desired target. Although Mithal and Douglas did not pursue their findings any further, such explanations of results could be valuable to devise technical improvements such as new filtering mechanisms. For instance, in the case of the laserpointer system tested in experiment I and II, analysis of the movement paths in the vicinity of the target could be used to improve the switching between the modes of the Kalman filter. Or information about the velocity profile of a movement could be used to improve the mathematical models of the filter that are used for prediction of the next pointer location.

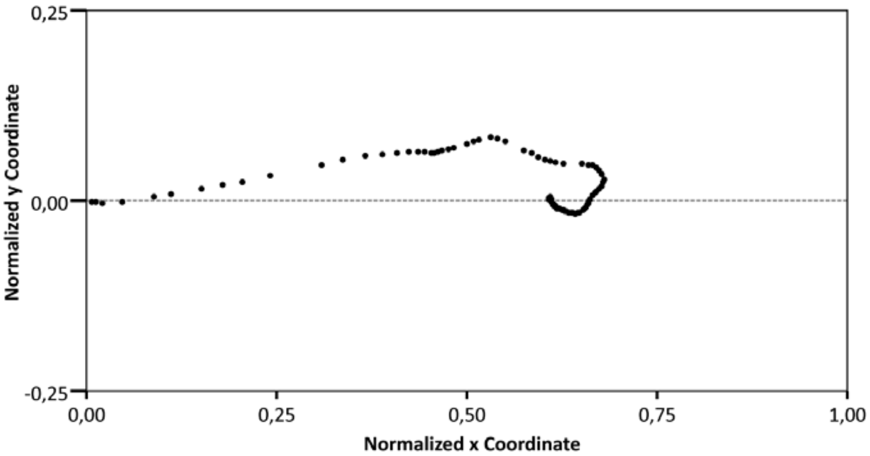


Figure 40: Rotated movement path during a single trial of the bubble test

For this purpose, the pointer position samples were recorded for each trial of the bubble test (see Figure 40). All movement paths were rotated so that the axis of approach corresponds to the x-axis to facilitate visual and quantitative analysis. Further analyses and interpretation is now required to effectively use this data to improve the pointing performance of the laserpointer.

5 Gaze Pointing

As pointed out by Salvucci (1999), eye movements can reveal a lot about the mind of a person. By observing the eyes of a person one can estimate the direction of their *gaze* to determine the *point of regard*, that is, the location where that person is looking at. This point of regard can be used for making inferences about what is going on in the mind of that person. Particularly, it can be used to make inferences about the *attention* of a person, that is, the distribution of cognitive capacities (Anderson, 2007). In other words, if a person looks at an object, it is almost certain that this object is of *interest* with respect to a task that the person deals with because some information can be gained for that task by looking at the object.

This reasoning motivated researchers in the field of human-computer interaction to build systems that use gaze information to interact with computers (e.g. Ware & Mikaelian, 1987). This chapter examines these gaze interaction techniques. First, the foundations of gaze interaction are considered. Next, existing systems for gaze controlled pointing and selection are reviewed and the main issues of these systems are outlined. Based on this review, novel concepts for interacting with gaze are proposed. The section is concluded with an assessment of the capabilities of mobile eye-tracking in the context of large high-resolution displays.

5.1 Foundations of Gaze Interaction

This section presents the foundations of gaze based interaction. First, the properties of vision and the functioning of the eye in particular the purpose of eye movements are elucidated. Then the technical aspects of tracking eye movements to infer gaze information are outlined. The first part of this chapter is finally concluded by a brief presentation of the use of eye movement information in human-computer interaction.

5.1.1 Foundations of Vision

Perception of the environment is essential to guide our actions. It creates an "(...) experience of the environment and enables us to act within it" (Goldstein, 2002, p. 3). Light, a certain spectrum of electromagnetic radiation, has certain properties that are particularly well suited to make inferences about our surroundings. For instance, light is practically everywhere and it is reflected from and absorbed by the surfaces of objects which enables us to make inferences about these surfaces even if they are located miles away from the observer. These and other characteristics of light "convinced" evolution to create an organ that was capable of detecting it: the eye. The architecture of the eye is so complex that even Charles Darwin himself wrote that it was hard to believe that such an intricate organ could have been formed by natural selection alone. For the purpose of this thesis only the utmost important aspects of the physiology and visual perception are outlined. Further details are for instance presented by Goldstein (2002) and Cornsweet (1974).

5.1.1.1 Physiology and Psychophysics

In order to make any use of the information provided by light, it must first be collected. This task is performed by the cornea and the lens. The cornea is a rigid transparent coating at the front of the eye and delivers most of the refractive power. The rest comes from the lens, whose refractive power is controlled by muscles attached to it which allow us to deform the lens to obtain a sharp image of objects at different distances. Before reaching the lens, the light entering the eye travels through the pupil, the opening in the iris, the colorful structure that can change its size to modulate the amount of light that enters the eye. After the light travels through the cornea, pupil, and lens, it is focused on the retina. The retina consists of small transducers, the photoreceptors, which convert light energy into electrical signals that are then processed further by the retinal ganglion cells before leaving the eye along the optic nerve to higher processing areas in the brain.

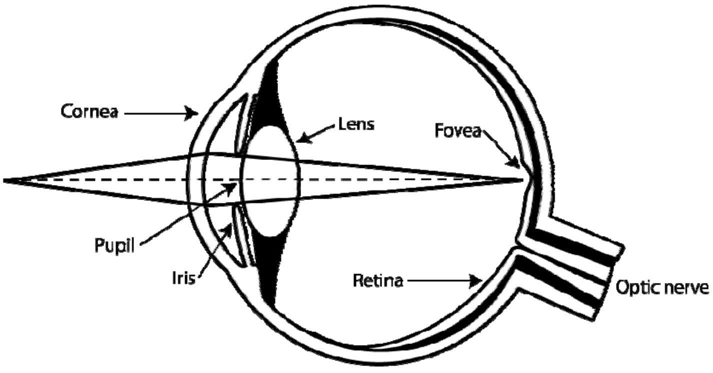


Figure 41: The eye

Two types of photoreceptors can be differentiated: rods and cones. Rods are highly sensitive to small amounts of light and therefore enable us to see at dawn or dusk (scotopic vision). Rods cannot be used to differentiate color. The color of a light is essentially determined by its spectral composition and to determine this composition with sufficient reliability, photoreceptors with at least three different levels of spectral sensitivity are required. Cones exist in three different types that have this characteristic and therefore enable the perception of color. Cones are also much less sensitive than rods and thus require more light (photopic vision). Rods and cones also differ in the way they are connected to other cells in the eye that forward the activation of the receptors to higher processing areas. Several rods usually converge on a single neuronal cell. The activation of a single rod activates that cell and thus enables the high sensitivity to even to the smallest amounts of light. However, this comes at the cost of spatial resolution. If the neuronal cell is activated, the information about the source of the activation – one out of several rods that connect to that cell – is lost. Cones on the other hand have less convergence. Therefore, they are less sensitive to light but also have higher spatial resolution.

Rods are predominantly found in peripheral areas of the retina. Cones, on the other hand, are concentrated in a tiny area on the retina. This is also the area of highest acuity, the *fovea*. The center of the fovea is the origin of the visual axis and the location of objects is often described relative to that axis in the form of the *angle of eccentricity* α as can be seen in Figure 42.

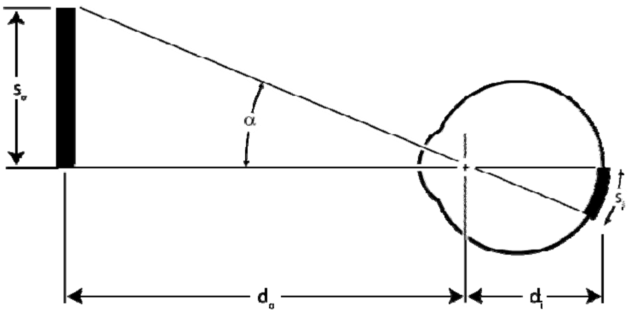


Figure 42: Angle of eccentricity

The angle of eccentricity α can easily be calculated from the given size of the object s_o and distance toward the object d_o.

$$\alpha = \arctan(s_o/d_o)$$

The foveal region extends out to an eccentricity of 1°, the region from 1° to 5° is called parafoveal region, and the remainder of the visual field the peripheral region (Findlay & Gilchrist, 2003). The very center of the fovea is called the foveola or fovea centralis where cone density is highest and visual acuity is best. Findlay and Gilchrist (2003) note that the definition of these regions is somewhat arbitrary since transitions of the determinants of acuity such as cone density are fluid. The acuity required, and therefore the amount of foveation needed, is highly dependent on the stimulus presented and therefore on the task.

5.1.1.2 Eye Movements

Since only the foveal area of the retina is really suited well for distinguishing fine details, the eye is continuously rotated to foveate details of interest. Interestingly, the purpose of some class of movements is not so much to actually orient the eye as to keep the retinal image of the environment stable and a certain detail foveated. The vestibular-ocular reflex (VOR) is a mechanism that uses information from the vestibular organ to compensate body and particularly head movements, so that the eye's position relative to the environment remains unchanged and fixation on a particular item is possible. Similarly, the optokinetic reflex (OKR) uses information from optic flow, that is, changes in the field of vision due to motion, in order to enable a realignment of the eye on a stimulus. The OKR generates a characteristic pattern of slow movements in one direction followed by rapid movements in the opposite direction that can for instance be observed when someone is looking out of the window of a driving train. The last type of movement in this class are smooth pursuit movements that stabilize the retinal image of a slowly moving target, for instance, when watching a flying plane or bird (Joos, Rötting, & Velichkovsky, 2003).

Findlay and Gilchrist (2003) point out that VOR and OKR are essentially involuntary and automatic whereas the other movements seem to have some volitional component relevant for target selection, particular saccades and vergence movements. Vergence movements adjust the rotation of the eyes so that lines of sight of both eyes converge on the target object. This is important when targets are located at a different viewing distance. Saccades, on the other hand, are rapid movements that rotate the eye so that the retinal image of an object falls on the foveal area. The period between two saccades is called a fixation and the eye remains relatively stable during this period compared to the saccade movements.

Vergence movements are usually coupled to saccades and are rarely important in the context of human-computer interaction because the distance to the computer screen usually remains constant. As mentioned before VOR and OKR movements do not allow much inference about attention or volitional acts of control. Pursuit movements cannot be controlled voluntarily and require a moving stimulus (M. E. Goldberg, 2000). Thus, saccades and fixations seem to be the most important aspects of eye movements in the context of HCI and therefore, the remainder of this section deals with the details of these.

Saccades

As mentioned previously, saccades move the eyes to foveate certain areas of the field of view, driven by voluntary, but not necessarily conscious processes, and involuntary processes, for instance, when a new unexpected stimulus appears in the field of view (Bruce & Friedman, 2002; Findlay & Gilchrist, 2003).

Saccades are very fast with a velocity of several hundred degrees per second. The movement characteristic of saccades is highly *stereotyped*. This means that the trajectory and acceleration and deceleration pattern is similar for saccades of the same amplitude (angular rotation). The movement is also *ballistic*, which means that the direction of a saccade is usually preprogrammed and remains unchanged even 50-100 ms before the initiation of a saccade and during a saccade (Bruce & Friedman, 2002, p. 284). Instead, corrective secondary saccades sometimes follow the main saccade if required by the task (Findlay & Gilchrist, 2003, p. 68). The time taken for a saccade ranges from 10 to 100 ms and can generally be predicted with a simple linear equation determined by the amplitude (Duchowski, 2003, p. 42; Findlay & Gilchrist, 2003, p. 26). During a saccade, processing of visual information is drastically reduced, a phenomenon which is known as *saccadic suppression* (Duchowski, 2003). However, one usually does not consciously experience this phenomenon similarly to how one is not aware of the fact that only foveation delivers a “sharp” image of the surroundings. Finally, the direction of gaze is determined by head as well as eye movements. Therefore, visual orienting usually happens in unison of eye and head movements. Because of the head’s inertia, eye movements usually precede rotation of the head. After the eyes have reached the target, the head movement follows and the vestibular-ocular reflex realigns gaze on the target (M. E. Goldberg, 2000, p. 796). Within 20° eccentricity only the eyes are commonly used for orienting. This region is called the *eye field*, as opposed to the head field, which extends to 90° eccentricity. Within the *head field*, orienting involves both, head and eye movements (Findlay & Gilchrist, 2003, p. 55).

Fixations

Fixations are, of course, not a separate class of movements but the pauses in between saccades where the foveal area is in alignment with the area of interest so that visual information processing can occur. The duration of fixations is not fixed but ranges from 100 to 2000 ms depending on the difficulty of the task and the visual information processing requirement (Joos et al., 2003; Just & Carpenter, 1976). During a fixation, the eye is not entirely stationary. Several types of movements affect the eye during a fixation: Slow *drift* and occasional *microsaccades* are said to play an important role in keeping photoreceptors desaturated by continuously moving the image projected on the retina. Additionally, very small and fast *tremor* movements jiggle around the projected image, presumably because of a similar reason. Experiments showed that vision faded rapidly once these kinds of movements were prevented by artificially stabilizing the retinal image. Other experiments showed, however, that also higher-level processes could affect these types of movements, demanding for a more thorough explanation (Findlay & Gilchrist, 2003). Whatever the reasons, the instability during a saccade imposes a challenge for building gaze interaction systems. First, users are usually not aware that their eyes are not still during a fixation and using the unfiltered stream of gaze data in a gaze contingent system could lead to some confusion. Second, usually these movements are of no interest anyway, because research has not yet provided enough information about what role these movements play in high-level processes. A particular problem is therefore to build systems that appropriately filter these movements and extract the fixation information (Salvucci & Goldberg, 2000).

5.1.2 Eye- and Gaze-Tracking

A vast amount of literature exists that describes the details of tracking procedures and techniques and gives information about advantages, disadvantages and the functionality of these techniques (e.g. Duchowski, 2003; Joos et al., 2003; Kolakowski & Pelz, 2006; Young & Sheena, 1975). For the purpose of this thesis only a brief overview is given about the basic procedures.

Three principal tracking techniques can be distinguished:

a. Electro-oculography (EOG) records electric potential differences between retina and cornea using electrodes that are placed on the skin near the eye to measure eye rotation.

b. Contact lenses that are inserted into the eye of the participant and that are equipped with some mechanical or optical feature that can be easily tracked such as small coils whose motion can be sensed in an electro-magnetic field (search coil).

c. Video-based tracking methods that employ a video camera to record a video stream of the eye and use computerized image processing methods to identify the location of the pupil in the video images to measure the rotation of the eyeball.

Besides the tracking accuracy and speed the suitability of a technique for use as a pointing device or usability studies is determined by several other criterions. A tracking technique

- Should work in a regular office setting
- Should be unobtrusive
- If required, should be easy to calibrate
- Should function for extended periods of time

In this respect, video-based tracking is by far the most important technique and is also most often used in studies and systems in human-computer interaction. Compared to the other techniques it is more practical and comfortable because it does not require that an object is inserted into the eye of the user and is also more reliable than electrode measurements (Duchowski, 2007). Additionally, current systems have reached a level of accuracy that is sufficient for most applications in human-computer interaction.

When using an eye-tracker with a computer screen it is usually not only required that the system can report the rotation of the eye but that the *gaze* and subsequently the point of interest on that computer screen is available. In order to determine this *point of regard* a calibration procedure is required since the video image of the eye-tracker only gives information about the rotation of the eye. In such a calibration procedure users typically look at several predefined points on the screen one after the other so that eye rotations can be mapped to points on the screen and other points can then be determined via interpolation.

In some systems, the video cameras for filming the eye are attached to the head, for instance with a headband that is worn by the user. Other systems employ cameras that are fixed to the desk in front of or even built into the computer screen. Since the point of gaze is not only affected by eye movements alone but also by movements of the head, these must also be considered when calculating the point of regard on the screen. The crudest, and certainly most reliable, procedure for doing this is to fixate the head with a dental bite bar. This assures that head does not move and the point of regard can thus be determined solely by considering eye movements. More comfortable methods exist though, depending on the location of the tracker cameras. Systems that have cameras mounted to a headband usually employ a separate tracking system that determines the relative location of the head to the screen, for instance, by using infrared markers that track the position of the head. Trackers that have fixed cameras attached to the screen or desk typically employ a corneal reflection method (see Figure 43). A small infrared light source is usually located near the camera whose light is reflected from the cornea or lens of the eye (this reflection is also called a Purkinje image) back to the camera. Because of the spherical shape of the eye and cornea, the point of regard can then be determined by observing the relative motion of the pupil and corneal reflection (Kolakowski & Pelz, 2006). This technique is also often employed in head-based trackers to improve tracking accuracy and compensate for headband slippage.

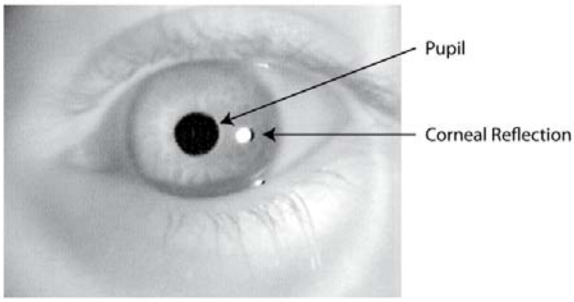


Figure 43: Corneal Reflection, adapted from (Babcock & Pelz, 2004)

5.1.3 Use of Gaze Information in HCI

First attempts in eye-tracking were already made at the end of the 19th century and the benefits of eye-tracking for human-computer interaction were also discovered early. In 1980, Bolt predicted that "(...) eyes-as-output has a promising future (...)" (Bolt, 1981, p. 118). Nevertheless, eye-tracking still seems to be confined to scientific ventures and is not encountered in any mass-market consumer applications yet. In their review on eye-tracking applications Jacob and Karn (2003) note how progress in the field was slowed down due to technical restrictions, the high cost of eye-tracking equipment and demanding data analysis. Yet, they also conclude that it is still a "growing field" given the increasing number of publications on the topic and that many of the problems are now solved, paving the way for new eye-tracking research and applications.

Two main applications of eye-tracking can be distinguished in general: eye-tracking for diagnostic purposes and interactive use of the eye-tacking data (Duchowski, 2007, p. 206). In the field of human-computer interaction eye-tracking is used a diagnostic tool for analysis of interfaces to improve the usability of applications and interactively to adapt or control the interaction with a computer system.

5.1.3.1 Usability Research

It is generally assumed that fixations of gaze correspond to the visual information that is currently processed and that this information could provide insight on how certain problem-solving activities proceed or how difficult an activity is. For example, analysis of the fixations made by a participant while solving a mathematical equation could reveal the mental steps that are involved in this task (Salvucci, 1999). Similarly, gaze information could aid in the analysis of a computer interface, for instance, to reveal why users do not employ a certain function in an application: either because they do not discover how to activate it because the button or menu is placed in an area where no attention is paid to, or because they do not understand the label or image used for the button which could be inferred from extraordinarily long fixation times. Jacob and Karn (1991) note that such a "bottom-up" approach is usually not guided by hypotheses and is rather a way to come to qualitative conclusions that elucidate the cause of findings that were obtained through classical usability testing. Generally, eye-tracking information can complement classical usability testing methods where no direct interaction data is available (e.g. the user looks at a window but does not do anything) and where think-aloud techniques do not provide adequate information because users simply are not aware of this "microlevel" behavior (J. H. Goldberg & Wichansky, 2003, p. 497). Additionally, attempts were made to relate eye tracking information to global measures employed in usability studies such as completion time or error rates. Goldberg and Wichansky (2003) give an overview of those attempts and conclude that some gaze measures could serve as predictors for specific aspects that influence the usability of a system such as its visual clarity.

The first study that employed eye-tracking in the (extended) area of usability research involved a researcher whose work was already presented in the previous chapters: Paul Fitts. In their study on eye movements of aircraft pilots, Fitts, Jones, and Milton (1950) observed how long and when pilots were looking at specific instruments to make inferences about the importance of an instrument or the difficulty involved in extracting the required information from an instrument. The study by Fitts et al. also illustrated one of the major issues that prevented a widespread use of eye gaze information in usability studies. Fitts et al. used a camera to record the pilot's eyes and to calculate mean fixation time and other measures for each instrument, the footage was analyzed manually in a frame-by-frame fashion. In a setting other than an aircraft cockpit and with today's tracking systems this analysis is certainly much easier now. However, novel problems emerge when interactive systems are the subject of analysis. It may not be an issue anymore to obtain the point of regard on the computer screen but to make sense of this information it must be matched with the content that is displayed on the screen (J. H. Goldberg & Wichansky, 2003). For instance, gaze information alone is not enough to determine whether the user looked at the contents of a pop-up window or if he looked at the objects that were concealed below after he closed it. These and other technical challenges complicate the use of eye-tracking in usability studies.

5.1.3.2 Interaction

Much research has also been devoted to the second aspect of use of eye-tracking data: interaction. Although the classification is not very clear-cut one can distinguish between systems that employ eye gaze as a primary means for interaction and systems that use gaze for subtle adaptation of certain aspects of the interaction.

Such an adaptive use of eye gaze was for instance presented by Starker and Bolt (1990). They developed a "self-disclosing" system that revealed its contents based on the interest of the user. Users were looking at a scene that was presented on a computer screen while listening to a narration about it. While users were looking around, the program analyzed the pattern of gaze relocations to infer the interest about certain objects that were presented in the scene. This interest information was then used to adapt the narration, including either more or less information about certain features of the scene. Another subtle use of gaze was demonstrated by Hyrskykari et al. (2000) in their iDict system. iDict did not adapt the content itself but provided additional information based on gaze information. The system was developed as an aid to reading foreign language texts. Analysis of the eye movements during reading was used by the system to determine which words were unknown to the reader. When such a word was detected, a translation was automatically displayed above the word.

However, the majority of other systems used gaze not for adaptive purposes but as a means for selection-through-pointing, for instance, to allow users to type with the eyes by selecting letters of an on-screen keyboard, or generally enabling selection of objects, menu items, or buttons in graphical user interfaces. These systems are considered in the remainder of this chapter.

5.2 Advantages and Issues of Gaze Interaction

In this section the advantages of gaze controlled selection-through-pointing put forth by writers in the field are presented. After that, the problems inherent in gaze based selection and solutions to these problems of various systems are considered.

5.2.1 Advantages

Gaze pointing is considered to be faster than pointing with the mouse since people tend to look at objects before interacting with them (Jacob & Karn, 2003). However, experimental results are contradicting in this respect. For instance (Miniotas, 2000) showed that gaze pointing is much slower than pointing with a mouse, whereas Sibert et al. found it to be significantly faster (Sibert & Jacob, 2000). There are probably diverse reasons for these conflicts most probably differing experimental methods, interaction techniques, and trackers with differing tracking capability. Some authors emphasize that although actual movement time may be higher, non-existing acquisition and homing times reduce discrete gaze pointing time when using it in conjunction with an ordinary keyboard where the task requires switching between typing and pointing (Fono & Vertegaal, 2005, p. 158; Kumar, Paepcke, & Winograd, 2007).

Jacob (2003) reports that some participants in his eye-gaze experiments mention a “feeling of a highly responsive system” as if the computer was reading the user’s mind, executing commands with clairvoyant capabilities. Clearly this perceived responsiveness could be a major strength of gaze interaction. It however depends on the implementation of the system.

Some authors propose that gaze-pointing requires no or less training due to its direct mapping which requires no learning of complex input-output transfer (Stampe & Reingold, 1995). However, no studies exist that thoroughly verifies this assumption. Presumably, as with perceived responsiveness, whether or not an eye-gaze system is highly intuitive also depends on the specific interaction mechanisms employed.

Finally, besides the obvious argument that gaze pointing could aid persons with disabilities, some authors mention that selecting without the need to move limbs, particularly hands, may reduce physical strain which often leads to illness such as repetitive strain injury (Zhai, Morimoto, & Ihde, 1999).

5.2.2 The Intention Issue

First gaze interaction systems were already available in the 1980s but they have not become an everyday technique for computer input and remain constraint to special, though not less useful application domains such as enabling computer input for people with motor disabilities[11]. Some writers attribute this to the high cost and lack of reliability of existing tracking systems (Li, Babcock, & Parkhurst, 2006). However, upon analyzing the existing systems it seems that the main issue is not the hardware but an inherent limitation with respect to the user's *intention and attention*. In this section the problem is considered of how to interpret gaze to make reasonable inferences about the user's *intention*: why is someone fixating something?

For gaze contingent selection-through-pointing this problem boils down to the question of how to actually trigger a selection. It is relatively clear that when a user moves the mouse and clicks on an object that it is his intention to manipulate this object in some way. In contrast, it is not clear at all what it means when a user looks at a specific object on the screen. Although, fixations may be a good indicator of overt shifts of attention, no information is available of the cause of these shifts. Indeed, eye movements are not fully under deliberate control and are for instance elicited automatically by peripheral stimuli (Joos et al., 2003).

The issue of how to trigger a selection is commonly referred to as the "Midas Touch Problem" (Jacob, 1991, p. 156). This metaphor, however, may be slightly confusing since it does not describe the fundamental underlying dilemma but rather the specific problem of a single "solution": the immediate selection of anything that is the target of one's (eye-)pointing – a technique that would certainly also lead to much confusion when using a common pointing device such as the mouse. In case of gaze pointing, the problem is more apparent since the primary function of the eyes is perception and not control (Zhai, 2003). An interaction technique that hinders the perceptual function is doomed to fail from the start because users cannot even get an overview of the information displayed on a screen before interacting with it. Therefore, gaze interaction systems must be able to distinguish between those two functions. In a naive "Midas Touch" system such a distinction is not made. Nevertheless, numerous approaches have been presented by researchers to cope with this issue, most of which let users' specify acts of control explicitly to turn pointing into a selection.

In appendix C a list of systems reviewed for this thesis can be found. The list indicates the exact nature of the selection-through-pointing technique employed by the various systems by giving information about how pointing is done, how a selection is triggered, what feedback is given for pointing, whether or not a change of representation like magnification of the selection area is required, or whether the system needs additional context information for the selection.

5.2.2.1 Dwelling

A technique found in many systems is dwelling in which selection is made contingent to a continuous fixation of some pre-define time interval (Ware & Mikaelian, 1987). The amount of dwelling used differs widely in existing systems as can be seen from Table 20.

---

[11] Consider for instance the assistive technology from Tobii: http://www.tobii.com

Table 20: Dwelling Times

Authors	DT (ms)
Ware & Mikaelian, 1987	400
Jacob, 1991	150-250
Stampe & Reingold, 1995	750
Lankford, 2000	500
Miniotas, 2000	250
Sibert & Jacob, 2000	150
Miniotas et al., 2004	750-1250
Miniotas & Špakov, 2004	1250
Ashmore & Duchowski, 2005	500
Miniotas et al., 2006	1000-2000

DT: dwelling time

Sibert and colleagues used very small dwelling times of 150-250ms, arguing that this lead to improved responsiveness of the system and that longer dwelling times were “awkward and unnatural” (Sibert & Jacob, 2000). Furthermore, Jacob (1991) suggested that small dwelling times could be used for actions that are quickly undoable such as dynamic information displays, whereas longer dwelling times, which require more explicitly controlled staring, may be used to trigger actions that potentially have “adverse effects”. The reason for this is of course that with a short dwelling time chances increase that selection happens inadvertently.

An advanced type of dwelling was used by Laqua et al. (Laqua, Bandara, & Sasse, 2007). Their “interest accumulation algorithm” uses a “gaze-interest” counter for each selectable item. Once the point of gaze falls within the area of an item, the counter is continuously increased until it reaches a threshold, which selects the item. When a user looks away and focuses on a different item, the counter of the old item stops and is just reset once the other item is activated. This has the advantage that interest information is not lost completely, as is the case for regular dwelling techniques, once the users quickly glances to another item and then returns to the item he was dwelling on before. In another variant of the algorithm, a decay function was used to decrease the counters when the associated items were not at the center of visual interest.

An inherent difficulty with dwelling is that, unlike for instance a pointer controlled with a mouse, the gaze point cannot be kept stationary. Small, involuntary 30 ms movements or slow drifts disturb the measurement of a position (Ashmore, Duchowski, & Shoemaker, 2005). Two approaches were made to tackle this problem. First, filtering mechanisms were used to smooth the pointing information (Surakka, Illi, & Isokoski, 2003). Second, the area of tolerance used to decide whether a dwelling action was stopped or not (i.e. by moving the gaze position out of that area to fixate something else) was increased. Techniques that pertain to the latter are presented in section 5.2.3.

Miniotas and colleagues proposed a new algorithm called “grab-and-hold” (GHA), which should compensate for deviations from a target due to involuntary movements. What is special about the GHA algorithm is how dwelling time is interpreted. Dwelling continues even if, during a saccade, gaze slightly shifts to the area outside of the target. Only detection of a new saccade leads to termination of the dwelling process. In their user studies (Miniotas & Špakov, 2004; Miniotas, Špakov, & MacKenzie, 2004) Miniotas and colleagues showed that, in a discrete task, the GHA algorithm effectively reduced errors induced by micromovements. Although the results seemed promising, Miniotas et al. recognized that it was still uncertain if their approach would work in a more natural setting with multiple targets.

Moreover Zhai and colleagues (Zhai et al., 1999) note that in modern graphical interfaces that are controlled with a mouse, objects can usually be selected in different ways through single- or double clicks. Such a distinction is not easy to support with dwelling although attempts have been made by (Lankford, 2000) to translate varying dwelling times to different button events such as button press/release or a full click to distinguish between dragging and selection by using changing dwelling feedback.

As can be seen from the overview of existing systems in the appendix C, dwelling is the most popular selection triggering solution for gaze pointing systems to date. The main advantage of this approach is that it does not require an additional device, rendering it highly attractive for systems used by persons with motor disabilities. On the other hand, the usability of the system highly depends on an appropriate setting of the duration of the dwelling time which may differ from person to person.

5.2.2.2 Hardware Button

Another way to give users more explicit control over the selection process is to couple the selection trigger to a button as it is done with many more common devices such as the mouse. Of course, this means that such a technique could not be used for eyes-only control. Nevertheless, there are certain advantages for users that are able to actuate a button, particularly in combination with a keyboard for entering text. For instance, Kumar et al. (2007) used keyboard buttons in combination with pointing and found this technique to work extremely well when users were given tasks that required pointing and text entry. In a comparative evaluation of their system against the mouse, Kumar et al. found that selection speed was similar to that of the mouse and even faster in a task where users had to type and point in alternation because no homing time was required. However, error rates were significantly higher for the gaze technique. The authors reasoned that the interplay between pressing a button and looking could be an explanation for this. Sometimes participants pressed the selection button before the eyes oriented toward it, possibly because participants were relying on information from the peripheral visual field. To overcome this issue, Kumar and colleagues proposed that the relevant fixation should be calculated from a time window that extends beyond the actuation of the button. Ware and Mikaelian (1987) reasoned that hardware buttons could be superior to dwelling because the latter always involved a fixed amount of time that is "lost" every time a selection is made. However, the results from their experiments are contradicting: In their first experiment the authors found that selection speed with dwelling was equal to that of the hardware button – with dwelling producing slightly more errors. In their second experiment with a continuous target selection paradigm instead of a discrete, the hardware button was faster but was also more error prone. Similar to Kumar et al., Ware and Mikaelian supposed that the higher error rate of the hardware button was due to missing synchronization of button presses with the location of the gaze position. In a recent experiment for the comparison of dwelling and hardware button techniques, Zhang and MacKenzie (2007) found that the hardware button was significantly faster than the dwelling techniques. However, similar to the results of Ware and Mikaelian, the gaze button technique exhibited a high error rate and the authors conjectured that the button was often pressed prematurely or too late.

In light of these results it seems that a principal question needs clarification with respect to manual actions and looking. In order to motivate gaze interaction Jacob and Karn (2003) state that a user "(...) usually looks at the destination to which he or she wishes to move". In another publication Sibert and Jacob mention that "(...) people look at what they are working on." (Sibert & Jacob, 2000, p. 282). To give underpinning to this claim Sibert and Jacob cite Just and Carpenter (1976) who give an overview of "Eye Fixations and Cognitive Processes". In their overview it becomes clear that fixations are spatially and temporally related to cognitive tasks such as mental rotation, sentence verification, and quantitative comparison. But it is questionable if one can use the results of these experiments to come to simplified general conclusions as those above.

No studies exist that verify the claims made by Jacob, Karn and Sibert in the context of human-computer interaction tasks. The issues with techniques that combine gaze and manual actuation of a button described above show that, supposedly, it is not always the case that users "look at what they are working on". At least the temporal contiguity may not be as strong as expected. Consider the simple action of selecting a word in a word processor and selecting an action to modify the word (e.g. to italicize the word). The action might be selected from a drop-down menu and users might only glance quickly at such a menu item or button while moving the mouse toward it but move their gaze back to what they were looking at prior to that movement, before actually triggering the selection of the menu item because they want to "witness" the effects of the action. A more thorough examination of the nature of eye movements during manual pointing tasks might provide interesting insight in this respect and could be helpful for the conception and improvement of novel gaze interaction techniques.

> *Research Question:* What is the nature of eye movements in conventional HCI tasks? Do users really "look at what they are working on"?

5.2.2.3 Other Selection Mechanisms

Ware and Mikaelian (1987) also tested a condition in which an on-screen button was used for selection. Users had to first fixate the target and then move their gaze over to the button to select it. Results showed that using the on-screen button was slower than dwelling and using the hardware button. Additionally, the on-screen button could probably only function in the special task setting of the experiment where no additional items are located between the target items and the on-screen button.

An interesting approach involving speech was presented by Miniotas and colleagues (Miniotas, Špakov, Tugoy, & MacKenzie, 2006). Speech was not used to select an action that was applied to the selected object but to aid the selection process itself. In their speech-augmented eye gaze interaction system, Miniotas et al. color-coded items that fell into a pre-defined region of interest of 100 × 100 pixels around the current point of regard. Each selectable item within this area was highlighted with a different color and users were able to select items by announcing the respective color. In an experiment, the speech-augmented technique was compared to a dwelling technique and results showed that selection was less error prone with the speech technique. However, the drawback of the method was its low speed, since much time was required to recall the name of the respective color and to pronounce it.

Electromyography (EMG) was used to measure facial muscle contractions in a system by Partala et al. (2001) as a selection trigger. The system required that electrodes were placed in the region of a specific facial muscle, the corrugator supercilii, which is located above the eyes at the level of the eyebrows and is typically used when frowning. A comparison against a mouse in an experiment with a discrete multidirectional pointing test revealed that technique was significantly faster than selection with a mouse. However, error rates of the technique (34%) were much higher than that of the mouse (0.9%). The authors report that, similar to the problems with hardware buttons mentioned before, many errors were due to failure of synchronization of muscle activity and gaze. Also, questionnaire results showed that users were less satisfied with the new technique compared to the mouse.

5.2.3 The Attention Issue

The other main difficulty for gaze pointing has its cause in the dissociation of gaze and attention, or more precisely: the dissociation between the current point of regard and the visual information being processed. All eye gaze systems assume that "attention is linked to foveal gaze direction (...)" (Duchowski, 2003, p. 14). In fact, this is not always the case. For instance (Posner, 1980) distinguishes overt and covert orienting. In the former, gaze and attention are conjunct whereas in the latter attention moves to the parafoveal visual field. Experiments reported by Posner strongly suggest that attention and gaze are controlled by separate systems which are interrelated in some way, comparable to the relationship "(...) found to hold between eye and hand movements" (Posner, 1980). Hand movements are usually preceded by movements of the eye. Similarly, eye movements are usually preceded by shifts of attention but of course hand and eye movements can also be carried out independently. This also holds for movements of the eye and attentional shifts, however, Posner notes that this kind of separation is "(...) not a normal property of visual perception" (Posner, 1980) and occurs only under specially controlled conditions. Posner suggests that demands of the task, rather than an intrinsic mechanism, account for the conjunction of gaze and attention. For instance, tasks requiring precise visual information and therefore high foveal acuity also require a readjustment of the fovea, whereas other tasks, such as luminance detection tasks used by Posner and colleagues, do not.

Covert attentional states are certainly not the primary subject of concern for gaze interaction. Even if attention is overt, that is, attention corresponds to the direction of the pupil and the fovea, it is still not known where inside the approximately 1° foveal field, or whatever foveal size considered, the attentional "spotlight" is. Yamato et al. (2000) conjecture that one of the main problems of gaze contingent pointing is that the measurement accuracy of 0.5 to 1° of today's tracking hardware is not high enough to provide for accurate gaze pointing. However, improvements in this respect are of no use for gaze interaction research without any progress in elucidating the complex relationship of attention, or rather, the visual information being processed, and the location of gaze.

Below, accuracy calculations for the Powerwall (22.7 ppi) display of the University of Konstanz and a regular monitor are provided to illustrate the problem. For instance, when viewing a common computer screen with a resolution of 96 ppi at a distance of around 50 cm, a visual angle of 1° subtends an area of 33 pixels in diameter. These calculations make obvious that a fair amount of uncertainty with respect to the visual information being processed by a user remains. And the amount of uncertainty could even be higher, since 1° is rather an arbitrary value as was explained before.

Table 21: Uncertainty about true Gaze Location

	Pixels	
Distance (m)	22,7 ppi	96 ppi
0,5	7,8	33,0
1	15,6	66,0
3	47,8	197,9
6	93,6	395,8

Distance in m
Pixels calculated for 1°

Calculating visual angles from common widgets from today's graphical interfaces give a good impression of what accuracy is actually required. Selection of a regular textual menu item of today's graphical systems requires an accuracy which is far above this limit.

Table 22: Widget Sizes

Widget (height)	Pixels	Angle (°)
File item (Explorer details view)	17	0,52
Text menu item (Acrobat)	20	0,60
Icon menu item (Explorer)	33	1,00
File item (Explorer thumbnail view)	96	2,90

Calculations for a distance of 50 cm and resolution of 96 ppi.

One can see that the amount of inherent uncertainty is tremendous. And in addition to this issue, technical inaccuracies due to the tracking system increase this uncertainty even further. Getting more information out of the gaze position with respect to the visual information processing demands is a challenge that requires researchers to further elucidate the functioning and characteristics of eye movements. However, several approaches to bypass this issue and these approaches are presented next.

5.2.3.1 Representation Changes

The first approach is similar to the techniques presented earlier in section 3.2.2.1 that improved pointing performance in a manual pointing task by increase the target width of the target.

In Lankford's ERICA system, a pop-up window appeared in the middle of the screen after a certain dwelling time, showing a magnified view of the area around the current gaze position (Lankford, 2000). Just as in the unmagnified view, users were able to point to certain areas and select items via dwelling. In order to close the zoom window users could select a large on-screen button at the bottom of the window. Ashmore and colleagues (2005) argue that although such full-screen zooming facilitated target acquisition it was not desirable due to the loss of contextual information when zoomed in. They therefore propose the use of fisheye lenses, a technique in which only a part of the screen area is magnified and the remaining area is compressed (see Figure 44), in order to preserve this contextual information. In an experiment Ashmore et al. evaluated three fisheye pointing techniques: in the first technique the fisheye lens was always present (omnipresent) and its position was slaved to the gaze point. This technique was highly affected by lens oscillation, that is, the lens was continuously jumping around due to the inconstant nature of gaze. In their second approach (MAGIC[12]), the lens only appeared at the location of a fixation after it was detected. Similar to the omnipresent variant, the position was then slaved to the current gaze point. In the grab-and-hold variant (see also 5.2.2.2), the lens appeared also just when a fixation was detected but remained fixed in place afterwards. Results showed that target selection times for both, the grab-and-hold and MAGIC approach, were significantly faster than the omnipresent technique or a control variant without any magnification. Ashmore and colleagues assumed that the the grab-and-hold and MAGIC techniques worked well because, unlike in the other variants, the magnification did not interfere with phases where eye movements were not used for control but to get an overview of the objects presented on the screen.

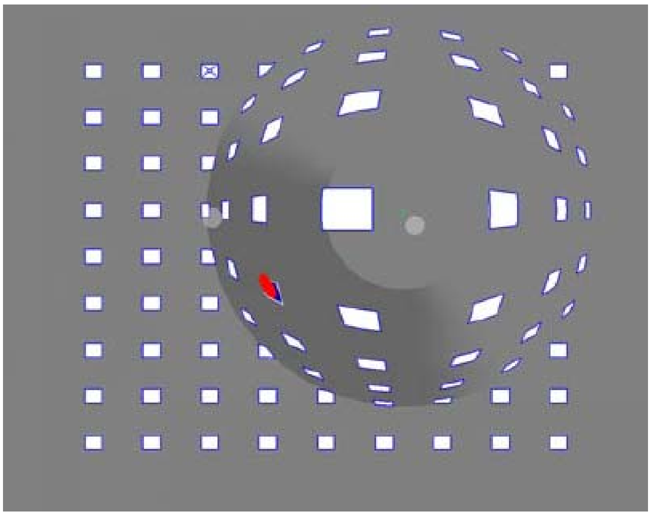


Figure 44: Fisheye Pointing, from (Ashmore et al., 2005)

A similar zooming technique was invented and tested by Kumar and colleagues (2007). In their EyePoint system, users were able to select items using their gaze and hardware buttons from a conventional keyboard in a look-press-look-release cycle to overcome the accuracy limitations of gaze pointing. The buttons were used to trigger a selection and various actions that are nor-

---

[12] Not to be confused with the original MAGIC pointing from Zhai et al. presented later.

mally bound to single or double-clicks of the mouse. Once such a button was pressed, the region around the current point of gaze was magnified. Finally, when the button was released, the item under the current gaze position in the magnified area was selected.

5.2.3.2 Context Information

In order to make selections via gaze pointing more reliable Salvucci and Anderson (2000) employed an algorithm that determined the likeliest selection through analysis of the gaze point in the context of the current task. In order to test the system, an entire WIMP-style system was built, with folders, files, and menus. The probabilities used for determining the selection were built into this system in the form of a rule set and probability score that was calculated for each selectable item. For instance, the likeliness that a user selected the "empty trash" function was reduced if no trash was available to be emptied. Therefore, the score of this special function was reduced. User testing and error measurements made during simple tasks in their system showed that this intelligent approach was effective. The drawback of the system proposed by Salvucci and Anderson is of course that the system is highly dependent on knowledge about the application context. This means that it cannot improve performance with legacy applications that do not have this knowledge built-in.

Context information is typically also used in eye-typing systems. Such systems present an on-screen keyboard that allows the selection of letters through pointing with gaze. Generally those systems also face the attention and intention issues as all other gaze selection systems (Majaranta & Räihä, 2002). However, these systems can also exploit grammatical or lexical information for instance by looking up words in dictionaries to map the gaze positions to the most likely word (Salvucci, 1999).

5.2.3.3 Gaze and Manual Pointing

Recognizing the inherent accuracy limitation of gaze interpretation, some researchers propose the combination of gaze and manual pointing. In their manual and gaze input cascaded (MAGIC) system Zhai et al. (1999) used eye gaze information to augment conventional pointing and selection with a mouse and isometric joystick. Essentially, their technique boils down to reducing the amplitude of a selection similar to the ideas presented in section 3.2.2.2 on page 26. The MAGIC system was tested with two approaches.

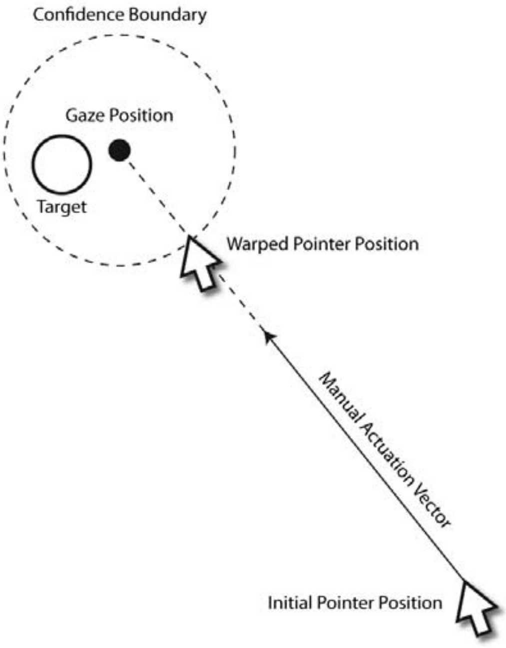


Figure 45: Conservative MAGIC pointing approach, adapted from (Zhai et al., 1999)

In the liberal approach the on-screen pointer was simply moved to the center of a 95% confidence area of the current point of gaze. Users could then move the pointer manually if it was not already positioned over the desired target. In the conservative approach the pointer was not moved at all unless the manual pointing device was actuated. The pointer then jumped not to the center of the 95% confidence area of the gaze point but to the intersection of the (ideal) movement path to the target and that confidence area. The authors speculated that this had the advantage that the initial motion to actuate the device could be used to naturally move the pointer from the morphed point onto the target to complete the selection. In a small user test the liberal MAGIC approach was found to be slightly faster than manual pointing alone. The conservative approach was slightly slower but some users found it less distracting than the liberal technique since the pointer did not always jump around with each gaze.

5.3 Novel Gaze Interaction Techniques

From the review of literature it becomes clear that gaze information alone may not suffice to enable rapid and accurate pointing and selection. However, information about users' gaze could also be used to improve manual pointing as is demonstrated by systems such as MAGIC (Zhai, 2003). With further progress in the development of unobtrusive and inexpensive eye-tracking systems, the barrier is lowered to use gaze information for improvements of manual pointing – even if the improvement is only small. In this section three new interaction techniques are presented based on ideas from the literature review. Void expansion and time scan are two selection techniques that use gaze as a primary means for pointing and for instance a button for triggering the selection. Sticky gaze illustrates how gaze information could be used to improve ideas to facilitate manual pointing.

5.3.1 Void Expansion

Void Expansion is a selection technique that seeks to weaken the effects of jittery motions during a fixation and general accuracy issues. Selection-through-pointing usually requires additional information to convert a pointing action into a selection. Numerous techniques that do this in the case of gaze pointing were already presented in section 5.2.2. Typically, the selection is determined from the spatial-temporal contiguity of pointer location and selection trigger (e.g. a hardware button). In other words: to find out which target is selected, the application compares the coordinates of the pointer against those of the objects on the screen once the triggering occurred. For instance, an item is selected in the Windows XP file browser if the pointer location is within the boundary of the item once the left mouse button is pressed. If a user accidentally overshoots the target and clicks slightly to the left or right of the item, it will not be selected. This behavior is unfortunate, since jitters during a fixation often let the gaze pointer wander outside the boundary of the desired target and a selection will fail if it is triggered just in that moment. The question is: is this harsh spatial-temporal contiguity necessary?

In the Void Expansion technique the pointing signal is continuously monitored to check if it points to one of the on-screen targets. Once a matching target is found, it is highlighted similar to the highlighting effect found in many applications when the pointer is moved over a button. The difference is that this highlight is “sticky” and it will not disappear once the pointer moves out of the target area and into the void, that is, screen space without any selectable items. If the gaze pointer is moved into the activation area of another item, this item becomes highlighted and the previous item de-highlights. Similar to the “grab-and-hold” technique of Miniotas et al. (Miniotas & Špakov, 2004) a selection is triggered irrespective of the current location of the gaze pointer – the target gets expanded to the void area surrounding it. This means that targets cannot be de-highlighted and de-selected by pointing to an empty area, as is commonly the case in file browsers. But one could certainly think of better methods for doing this, for instance, allowing a de-selection by selecting an item twice or providing for a special de-selection area on the screen. Also, items could generally get de-selected once an action was applied to them. Allotting the whitespace of a graphical interface to de-selection simply wastes too much space.

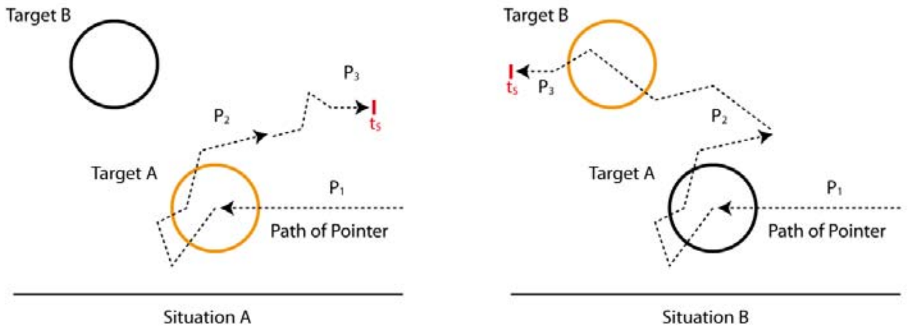


Figure 46: Example of the void expansion selection technique

Figure 46 illustrates how the Void Expansion technique works. In situation A, the pointer first moves to target A and the target becomes highlighted. After that, the pointer moves into the void. The selection is triggered at time t_S while the pointer is still in the void. Nevertheless, since target A is highlighted, it now becomes selected. In Situation B, the pointer first enters target A, the target is highlighted and then the pointer moves out of the target again. The next movement, however, is made toward target B, and upon entering this target, target A becomes de-highlighted and target B is highlighted instead. When the selection is triggered in this situation, target B becomes the selected target.

Because the highlight does not have a meaning itself, setting it falsely does not have "adverse effects" in the sense of Jacob (1991) – it can be undone instantly. The problem with the technique is of course that it does not work very well if no empty space is present around the target items. This is commonly the case with icons bars, where items are packed closely together to save space. But even then a little space might be available to increase the jitter tolerance when one tries to select such an item.

5.3.2 Time Scan

The time scan idea is similar to the technique employed by Miniotas and colleagues (2006) which was already presented in section 5.2.2.3 on page 96. In order to select a target, users first fixate it and press down a hardware button, for instance a key on the keyboard. As outlined in the previous sections, it cannot be assumed that this initial gaze is very accurate with respect to the true locus of interest. Therefore, similar to the technique of Miniotas et al., all selectable items must be considered that fall within a certain area of interest or confidence. Instead of using colors to mark these items, a temporal marking is used in the form of a moving highlight.

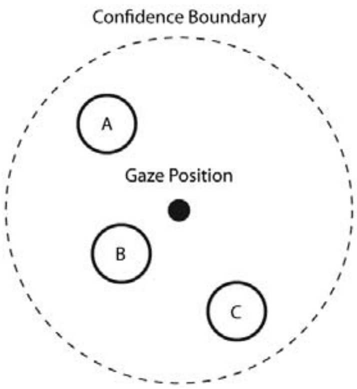


Figure 47: Several targets within a gaze confidence boundary of the time scan technique

For instance, in Figure 47 three targets can be found within the confidence boundary. These targets are then highlighted one after the other for a certain amount of time. Once the user releases the depressed key, the target that was highlighted at that moment gets selected. If the highlight cycle reaches the last target and no button release was detected it starts over with the first target. If only a single target is present within the confidence boundary, the target remains highlighted continuously and the user can select it by instantly releasing the key.

A reasonable highlighting period per item must be determined experimentally and probably ranges somewhere between 500 and 1000 ms. Also, the amount of time allotted for each item could be varied on the basis of additional information, for instance its importance if task information is available (see 5.2.3.2, p. 100) or its nearness to the gaze position.

The advantage of the time scan method is that no pop-up windows or similar representation change are required that temporarily obstructs the scene. Additionally, the technique could be employed with a keyboard for text entry and does not need any other modality, for instance speech. The fundamental drawback of the technique is of course that selection requires the user to react rather than act. As a result it may induce a feeling of time pressure that users could find unpleasant. Also the confidence boundary must probably be fairly small so that not more than a handful of targets must be scanned.

5.3.3 Sticky Gaze

As presented in section 3.2.2, thinking along the lines of Fitts' Law has simulated the development of novel interaction techniques that are aimed at selection-through-pointing. One such technique was to increase the size of the targets to facilitate pointing movements (McGuffin & Balakrishnan, 2002; Zhai et al., 2003). The drawback of these techniques is that it must be inferred which targets to expand prior to the movement. Additionally, the problem of overlap or distortion exists. If targets are packed closely, then the size of the other items must either be reduced or the expanded target overlaps parts of the other items. A technique that does not have this drawback but a similar effect was presented by Worden et al. (Worden, Walker, Bharat, & Hudson, 1997). Their "sticky icons" simply increase the C/D gain (the ratio between movement of the device and movement of the on-screen pointer, see 2.2.2.2)[13] of the pointing device while the pointer was within the boundary of such an icon. For instance, if the C/D gain is initially set to 1, moving the mouse for one unit also moves the pointer one unit. Setting it to 2/1 = 2 means that now two units of mouse movement are required to move the pointer for one unit. This had the effect that when users were moving the pointer over such an icon it seemed as if the pointer was "grabbed" by the object and users required larger movements of the mouse to move it out of the icon again. In effect in this way the size of the target was expanded without actually changing its representation and occluding or distorting anything else. The problem with this technique is that, if all potential targets were made sticky this would potentially hinder the long movements to a target across many other sticky targets.

It is suggested that the problems inherent in these techniques could be mitigated by additionally using eye-tracking information. In the case of expanding targets the point of regard could be combined with analysis of the movement trajectory of the pointer to predict the possible target and confine expansion to that target only. In case of C/D gain adapted interfaces the eye tracking data could be used in a "Sticky Gaze System" to enable C/D gain adaptation in the vicinity of possible targets so that acquisition of these targets is facilitated while disabling it for those targets that are not in the vicinity of the locus of visual attention.

---

[13] Note that Worden et al. use D/C gain which is simply the reverse of C/D gain.

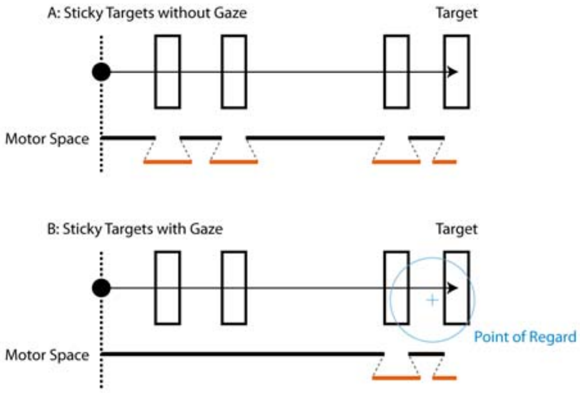


Figure 48: Sticky Targets

An example is shown in Figure 48 situation A: the movement path of the pointing movement passes three sticky objects before reaching the intended target object. The other three sticky objects that are in the movement path are also selectable but currently not the target of the pointing. However, because the C/D gain is also increased for these targets, the amplitude of the movement is significantly longer than that of the original movement. On the other hand, if gaze information is used, which is shown in situation B of Figure 48, the stickiness of two targets can be turned off, because these are far away from the point of regard and thus, the movement amplitude in motor space gets reduced.

As illustrated in the example, the precise point of regard is actually not located on any of the targets. Either because the tracking system is affected by technical inaccuracies or simply because the exact point of visual attention cannot be determined and the tracking system is reporting a "false" location (section 5.2.3). In order to compensate for this, the point of regard is extended to a circular area similar to the confidence interval of the MAGIC pointing technique of Zhai et al. (1999). It is thus assumed that the true locus of attention is somewhere within this circular area. The size of a reasonable "area of confidence" could be determined empirically or it could be made adjustable by the user. In the example the confidence area overlaps two objects, and therefore stickiness is turned on for both of these objects. Enabling stickiness of the object that was not the target does not have an adverse effect, because it increases the movement time required only slightly.

A possible problem for the application of such a "sticky gaze" technique could be the nature of eye movements during a manual pointing task. Do the eyes jump between the pointer and the target during such a task? Or do they remain fixated on the target? Or is there a characteristic movement pattern: first fixating the target, then searching the pointer and fixating the target once movement toward the target commenced? Or, even more intricate, is the movement behavior dependant on the movement distance?

Research Question: What is the pattern of eye movements during a manual pointing task? Is there a difference in patterns for direct and indirect devices?

A first informal test with a mobile eye tracker and the laserpointer indicated that shifting the point of regard occurred prior to the movement of the pointer and that it remained at the target location for some time. However, these findings should be verified in a more controlled experiment in which the point of regard is recorded along with the location of the manually controlled pointer during a target acquisition task such as those presented in chapter 3. Such an experiment should also include an indirect device such as the mouse. With an indirect device the pointer must first be located before a movement can be initiated. With a direct device such as the laserpointer or a touch screen device, this is not the case because those can be directly aimed at a target.

5.4 World of Windows: Reissue

One of the first systems at all to use gaze information for interacting with a computer was presented by Bolt (1981). His "World of Windows" was an interactive application that was shown on a wall-sized display in the "Media Room" of the Architecture Machine Group of the MIT[14]. The application presented a number of windows showing images or video snippets and users were able to "select" one of these windows simply by looking at them (see Figure 49 below). Once users were fixating a particular window, first of all the volume of the audio track of the window's video was increased so that is was louder than the rest of the audio from the other videos. Furthermore, if gaze remained on a video for a certain amount of time, the window was zoomed to fill the whole area of the projection screen. Several scenarios were described how zooming in and out of windows could be achieved by combining gaze and manual input, for instance, by using a joystick to quickly trigger the zooming without any need for prolonged dwelling or manually overriding the zooming function if it was not desired. Bolt's system was part of a larger effort of the MIT Architecture Machine Group to create novel human-computer interfaces that were particularly aimed at using spatial relationships as a basis for the organization of information (Bolt, 1984).

---

[14] Videos of these systems can be obtained from http://www.media.mit.edu/speech/sig_videos.html

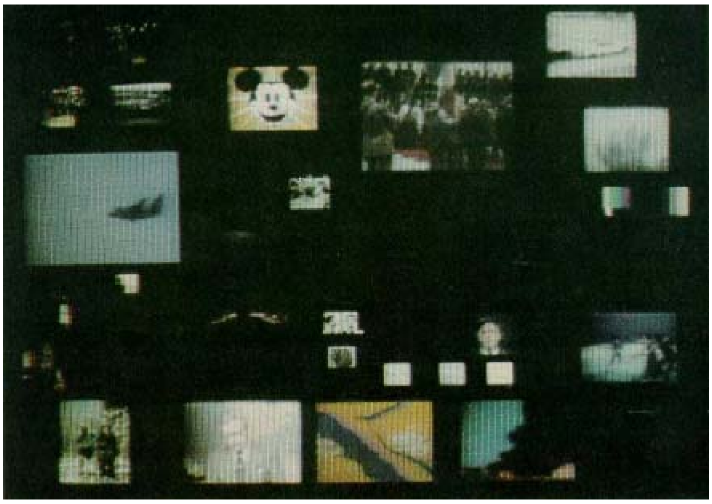


Figure 49: The World of Windows, from (Bolt, 1981)

The “World of Windows” demonstrated how eye gaze could be used to determine a user’s interest into spatially distributed information to “enable and protect” (Bolt, 1981). On the one hand, it enabled and facilitated information intake and on the other hand it protected the user from the vast amount of other information that was currently not of interest to the user.

A system similar to that of Bolt was developed during this thesis. However, in contrast to Bolt’s system the aim was not to build a complete system for information workers but to evaluate the applicability of mobile gaze-tracking and gaze based interaction in front of large displays such as the Powerwall at the University of Konstanz. In the remainder of this section, this system is described and various observations are reported that were made by the author while using the system.

5.4.1 Hardware and Tracking

The system used for gaze tracking for movable observers was developed by researchers at the Max Planck Institute, Biological Cybernetics, Department of Cognitive and Computational Psychophysics in Tübingen who made their system available to the author (Herholz, Tanner, Canto-Pereira, Fleming, & Bülthoff, 2007). The main feature of their “libGaze” system is the calculation of the point of regard on the display by using information about the rotation of a user’s eyes and head position relative to the display. The eye-tracker used with the system for this applicability study was an SR Research[15] Eyelink II eye-tracker, a head-mounted device with two cameras that were attached to a headband that was worn by the user. As pointed out earlier, to determine the gaze of the user to locate his point of regard on the display the head position must be located relative to the display. This was done by attaching a structure of small reflective markers to the headband that could then be seen by an optical tracking system. The optical tracking system used for the study was a commercial system from ART[16] that was already installed in the room where the Powerwall was located. It consisted of several infrared cameras and diodes that sent out infrared light flashes that were reflected by the markers so that they could then be de-

---

[15] http://www.eyelinkinfo.com

[16] http://ar-tracking.eu

tected by the cameras. Because the field of view of the cameras overlapped, the position of a pre-defined structure of markers could be detected with six degrees of freedom (the xyz-position and rotation around these axes). Therefore it was possible to determine the position and rotation of the head relative to the display wall.

However, before the system could track the user's gaze, it had to be calibrated. In a first step, the approximate location of the eyes relative to the headband markers was determined by holding a custom staff with tracking markers at the bridge of the nose of the participant. This procedure was required to calculate the direction of gaze from the combined information of the location of the head and rotation of the eyes. After that, the calibration of the eye-tracker had to be carried out. This was done in a similar fashion as with regular displays by looking at distinct points. During calibration, the user stood in the center of the display to fixate several points that were displayed one after the other on the screen. The head had to remain still during this procedure. After the calibration was carried out, the eye-tracker system was ready for use and the user could walk freely in front of the Powerwall.

5.4.2 Application

Similar to Bolt's system, the application consisted of several windows that were located randomly on the screen. The windows contained images of various sizes and content and could be moved across the screen. By default the images in the windows were displayed in a lower resolution than was actually available. As with the original world of windows, individual windows could then be zoomed so that they stood out of the other windows and were displayed in full resolution.

Figure 50: Zooming a Window via Dwelling

Several interaction techniques were tested with the system. First eye-gaze could be used to point to a window and zoom it as is shown in Figure 50. A transparent blue dot was displayed to give pointing feedback and the windows were slightly highlighted when the gaze pointer moved within a window. Once the pointer remained within a window for a pre-defined amount of time (1s) the window was zoomed to full size and resolution. Additionally, it was moved in front of other windows that were overlapping it. Once the pointer left the window it was scaled back to the smaller size. Furthermore, the eye-tracker was combined with a button to allow for dragging operations. For this purpose, the user was simply given a mouse that he held in one hand and one of the mouse buttons was used as the selection trigger for the drag operation. In order to drag a window, the user first pointed to an image by looking at it and then pressed down the button. The window position was then slaved to the gaze of the user and could be moved around by looking at a certain position. In order to drop the window, the user simply released the mouse button.

Figure 51: Dragging a window with gaze

A dragging operation is shown in Figure 51. First the user looks at the green image. After the one second dwelling time, the window is zoomed to its full size. The user then decides to drag the window slightly to the right. Upon pressing the mouse button it is moved to the front and now overlaps the smaller window to the left. The user then moves his gaze to the right and releases the mouse button to drop it. The window remains zoomed while the gaze is still detected within its boundaries and is scaled back when it leaves the window.

Furthermore, gaze-tracking was tested in combination with the laserpointer system that was already described in section 4. Two differently colored on-screen pointers were displayed on the screen: one slaved to the point of regard and the other to the laser spot position. The laserpointer was used for dragging around windows on the screen. Users pointed to a window and pressed the button of the laserpointer to enable the tracking. The gaze information was merely used to augment the system by zooming the windows in the same fashion as described before.

5.4.3 Observations

5.4.3.1 Drift

First of all, a major issue existed with respect to the tracking accuracy because of *drift*. Although drift is also used to describe the drifting motions of the eye during a fixation, here the term refers to how the difference between the calculated and the true direction of gaze gradually increased, primarily due to slippage of the head-band or unintended movement of the cameras. This issue, of course, also exists for eye-trackers that are used with regular computer displays (Kumar et al., 2007). However, participants typically do not move much with these systems and remain stationary in front of the display. This was not the case in the setting described above and the user could move around in front of the display freely which significantly increased drift. While the system was reasonably accurate shortly after calibration, using it for an extended period of time was not possible because it became inaccurate after a couple of minutes, depending on the amount of head or body movement. Before the system could be used again it had to be recalibrated completely which was very annoying for the user.

In typical eye-tracking experiments, a drifting correction is usually built into the experiment at pre-defined intervals. In such a drifting correction procedure a small dot is displayed in the center of the screen and the participant fixates this dot. The location of the dot is then compared to the point of regard on the screen to calculate the difference between the two which can then be used to correct the point of regard without the need for a full recalibration. For gaze interaction systems dynamic drift correction procedures have already been proposed (Stampe & Reingold, 1995; Velichkovsky & Hansen, 1996). In these systems the drift correction is built into the system by assuming that gaze falls at the center of an object once the user selects it. Therefore, drifting corrections can be made without the user even noticing this procedure by comparing the point of regard to the object location in the event of a selection.

Before commencing any further undertakings with gaze based interaction in the context of large displays with this system, the drifting issue must be tackled. A first step in doing so would be to add means for drift correction to the libGaze library so that application programs can carry out drifting corrections once in a while, without closing the program and switching to a separate calibration application. If such functionality is available, also other means as the dynamic drifting correction described above could be built into systems and tested to improve accuracy.

5.4.3.2 Gaze Pointer

Secondly, it was observed that displaying a gaze pointer was rather disturbing. Due to inaccuracies in the tracking mechanism the location of this pointer was sometimes displaced from the true point of regard: when the user tried to look directly at the displaced pointer it was therefore moved slightly away from the point of regard and the user started chasing it around in an open-loop fashion just like a dog is chasing his own tail. Although, one could simply try to ignore the pointer to stare “correctly” to effectively control the pointer, this experience was not very natural. Similar experiences were reported by others. Kumar and colleagues (2007) came to a similar conclusions. Although they designed their system without any direct pointing feedback at first, some users mentioned that they would prefer a pointing feedback. When the feedback was finally implemented in the system the same users concluded that it was rather distracting and it was therefore turned off again.

It seems that pointing feedback is not desirable for gaze based interaction. However, it is a good way to discover a malfunctioning of the system due to calibration failure. It is thus suggested that forthcoming systems should be designed without requiring pointing feedback but to make it optional if desired.

5.4.3.3 Dragging

It was also found that slaving the position of the object in a drag and drop operation to the point of regard was not desirable. One reason for this was that, when large objects were dragged, a large part of the foveal area was covered by this image so that locating the dragging destination had to be done by looking “out of the corners” of the eyes, that is, without directly foveating it. Another issue was that, in the dragging technique the object was dragged from the location where the user grabbed it. For instance, if the user looked at the lower left corner of an image while clicking the button, moving the gaze also moved the window so that the lower left corner coincided with the point of regard. One could easily fall into an open-loop dragging behavior similar to the one that described above for the displaced pointer, when relocate relative to the window itself is desired, for instance, when one wants to drag it from the center instead of from the lower left corner where it was grabbed.

Similar observations were also made by Lankford (2000) who employed dragging in his ERICA system. Although users were able to effectively drag objects across the screen, the dragging operation required that they ignored the dragged object to be successful rendering this technique very unnatural. It is therefore suggested that relocations should be done without slaving the gaze position to the object and rather than dragging and dropping the object, users should be able to pick up and release an object to make it *warp* to the new position so that their view is not obstructed while they are trying to locate this new position.

5.4.3.4 Zooming

Another issue was discovered with the zooming technique. While zooming into a window was found to be very natural, zooming out was not. In the first implementation the zooming ended once the point of regard left the boundary of the zoomed window. This was unfortunate because it often happened when looking at a detail of the image in the window that was close to its edge. Therefore in a succeeding implementation a border was added to the windows which reduced this issue. Further improvement could be achieved by also implementing a dwelling period into the zoom-out action.

6 Summary and Conclusion

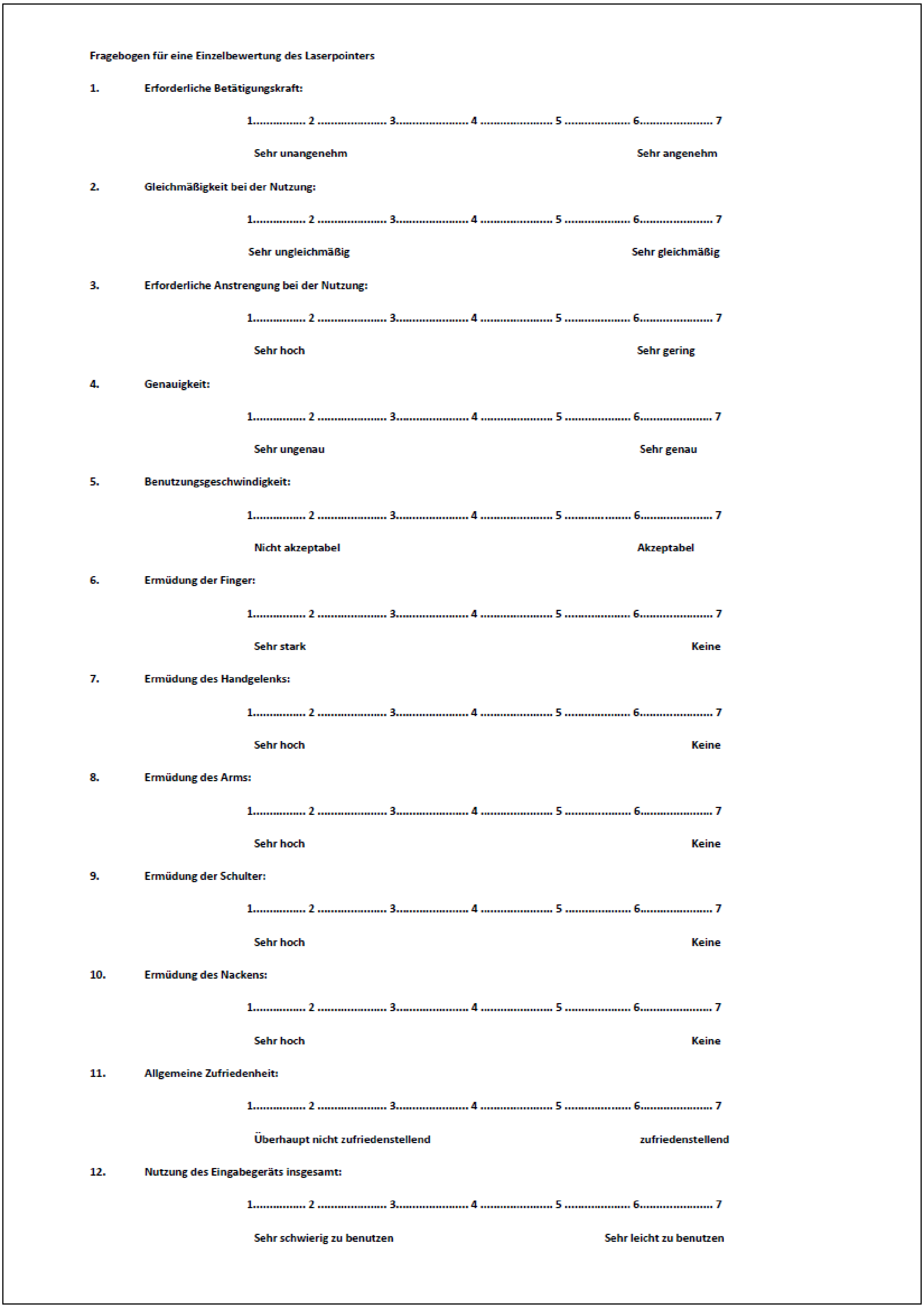
In this thesis a comprehensive explanation was given of the foundations and methodology for assessing pointing performance of computer pointing devices based on the international standard ISO 9241 part 9. The explanation was followed by a critical review of this methodology. It was suggested that the pointing movements made in discrete tapping tests are more similar to natural movements in the context of a real application. Further empirical investigation of this claim could be useful to support arguments for an inclusion of discrete tests in the standard. Furthermore, it was suggested that the standard should be more specific in describing the multidirectional test variant to clarify and standardize calculation procedures. Efforts are currently undertaken by the technical committees of the ISO to revise the ISO 9241 standard and this could be an opportunity to contribute these suggestions (the current status of the relevant part "411: Laboratory test and evaluation methods for the design of physical input devices" is: "planned").

It was also criticized that the standard does not give thorough advice on how to treat effects of practice when carrying out an experimental evaluation of pointing devices. A review of literature gave an overview about the possible approaches. In order to examine the effects of practice, a long-term experiment was conducted. The pointing device under scrutiny was a laserpointer system that was developed by the HCI group of the University of Konstanz. It was used at the Powerwall at the University of Konstanz, a large 5.20 × 2.15 m high-resolution display. The goal of the experiment was to assess whether practice had an effect on the performance of the laserpointer. The experiment was conducted with four participants and five sessions were held for each participant on consecutive days. During the course of the experiment each participant carried out almost 4800 pointing and selection trials. The results showed that practice had a large effect on the performance of the laserpointer. Performance increased around 1 bits/s in session one (≈ 30%) and another 1 bits/s from session two to the middle of session four (≈ 20%). These results implicate that future pointing device evaluations should allow for more practice trials so that effects of practice can be considered. Based on the results of the experiment, an improved comparative design was discussed and it was suggested that analyses of performance curves should be used in comparative evaluations. These analyses could help with the interpretation of results and could be used to motivate further long-term experiments.

In the last section of the thesis an overview was given about gaze interaction and two main issues were identified in a literature review: Interaction techniques that are solely based on gaze information are critical because the user's intent cannot be inferred from gaze information alone. This is problematic because it could lead to inadvertent triggering of selections. Furthermore, a general problem exists with respect to the pointing accuracy of gaze interaction system. This issue is not technical but due to the "(...) dissociation between the gaze point and the user's visual attention" (Salvucci & Anderson, 2000). Additionally, observations were reported that were made during first tests with a mobile gaze-tracker that was used on a large high-resolution display. This system allowed the user to walk around freely while determining the users' point of regard on the display. This first test showed that the most important issues with mobile eye-tracking are still technical. Prolonged accurate pointing is not yet possible with the system because it is severely affected by slippage of the head-mounted eye-tracker. Despite these shortcomings the author is convinced that further research into gaze-based interaction techniques is fruitful – particularly in combination with other input modalities. Two new techniques were outlined to enable gaze-based selections in combination with a hardware button (Void Expansion, Time Scan) and one to facilitate manual pointing operations (Sticky Gaze). These techniques could now be implemented so that first observations and experiences can be reported to devise other novel techniques.

Appendix B

ISO 9241-9 Independent rating scale (German Version)



Fragebogen für eine Einzelbewertung des Laserpointers

1. Erforderliche Betätigungskraft:

1................ 2 .... 3..................... 4 .... 5 .... 6...................... 7

Sehr unangenehm
Sehr angenehm

2. Gleichmäßigkeit bei der Nutzung:

1................ 2 .... 3..................... 4 .... 5 .... 6...................... 7

Sehr ungleichmäßig
Sehr gleichmäßig

3. Erforderliche Anstrengung bei der Nutzung:

1................ 2 .... 3..................... 4 .... 5 .... 6...................... 7

Sehr hoch
Sehr gering

4. Genauigkeit:

1................ 2 .... 3..................... 4 .... 5 .... 6...................... 7

Sehr ungenau
Sehr genau

5. Benutzungsgeschwindigkeit:

1................ 2 .... 3..................... 4 .... 5 .... 6...................... 7

Nicht akzeptabel
Akzeptabel

6. Ermüdung der Finger:

1................ 2 .... 3..................... 4 .... 5 .... 6...................... 7

Sehr stark
Keine

7. Ermüdung des Handgelenks:

1................ 2 .... 3..................... 4 .... 5 .... 6...................... 7

Sehr hoch
Keine

8. Ermüdung des Arms:

1................ 2 .... 3..................... 4 .... 5 .... 6...................... 7

Sehr hoch
Keine

9. Ermüdung der Schulter:

1................ 2 .... 3..................... 4 .... 5 .... 6...................... 7

Sehr hoch
Keine

10. Ermüdung des Nackens:

1................ 2 .... 3..................... 4 .... 5 .... 6...................... 7

Sehr hoch
Keine

11. Allgemeine Zufriedenheit:

1................ 2 .... 3..................... 4 .... 5 .... 6...................... 7

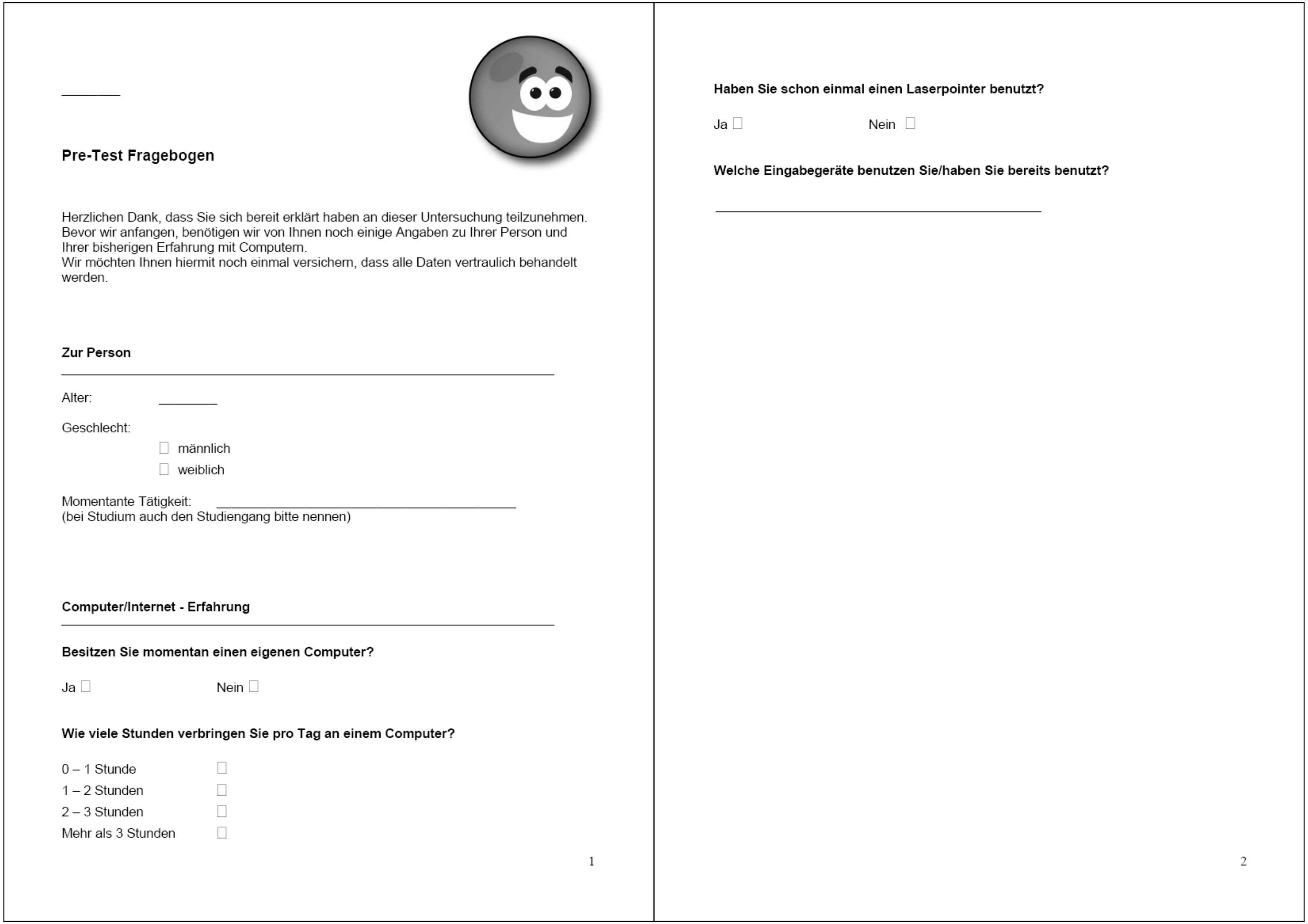
Überhaupt nicht zufriedenstellend
zufriedenstellend

12. Nutzung des Eingabegeräts insgesamt:

1................ 2 .... 3..................... 4 .... 5 .... 6...................... 7

Sehr schwierig zu benutzen
Sehr leicht zu benutzen

Pre-Test Questionnaire



Pre-Test Fragebogen

Herzlichen Dank, dass Sie sich bereit erklärt haben an dieser Untersuchung teilzunehmen. Bevor wir anfangen, benötigen wir von Ihnen noch einige Angaben zu Ihrer Person und Ihrer bisherigen Erfahrung mit Computern.
Wir möchten Ihnen hiermit noch einmal versichern, dass alle Daten vertraulich behandelt werden.

Zur Person

Alter: ________

Geschlecht:
☐ männlich
☐ weiblich

Momentante Tätigkeit: ____
(bei Studium auch den Studiengang bitte nennen)

Computer/Internet - Erfahrung

Besitzen Sie momentan einen eigenen Computer?

Ja ☐ Nein ☐

Wie viele Stunden verbringen Sie pro Tag an einem Computer?

0 – 1 Stunde ☐
1 – 2 Stunden ☐
2 – 3 Stunden ☐
Mehr als 3 Stunden ☐

1

Haben Sie schon einmal einen Laserpointer benutzt?

Ja ☐ Nein ☐

Welche Eingabegeräte benutzen Sie/haben Sie bereits benutzt?

____

2

Appendix C: Summary of Gaze Interaction Systems

Repr. = requires change of representation, Context = requires contextual information (e.g. pertaining to the task) for selection, Fitts = Fitts' Law model fit

Authors	Date	Repr.	Context	Pointing	Selection Trigger	Feedback	Notes	Fitts
Bolt	1981	-	-	Gaze	Dwell and Hardware Button	Zoom/Audio	World-of-Windows application for information workers.	-
Ware & Mikaelian	1987	-	-	Gaze	Dwell	Pointer/Highlight	Comparison of selection methods with discrete and continuous tasks.	Yes
		-	-	Gaze	Screen Button	Pointer/Highlight		
		-	-	Gaze	Hardware Button	Pointer/Highlight		
Jacob	1991	-	-	Gaze	Hardware Button	Unknown/None	Comparison of drag-and-drop techniques and observations from pilot study.	-
		-	-	Gaze, Mouse	Dwell	None		
		-	-	Gaze	Dwell	None		
Stampe & Reingold	1995			Gaze	Dwell	Highlight	Comparison of dynamic drift correction technique.	-
Zhai et al.	1999	-	-	Gaze, Mouse, Isom. Joystick	Hardware Button	Pointer	Combination of gaze and manual pointing and comparison against mouse.	Poor
Lankford	2000	Zoom	-	Gaze	Dwell	Dwelling Feedback	Dwelling with several stages. Magnification of focus area through zooming.	-
Miniotas	2000	-	-	Gaze	Dwell	Pointer/Audio	Comparison of gaze pointing against pointing with the mouse.	Yes
Salvucci & Anderson	2000	-	Task, Objects	Gaze	Hardware Button	Pointer	Improving accuracy with a probabilistic model based on task semantics.	-
Sibert et al.	2000	-	-	Gaze	Dwell	Unknown/None	Evaluation against mouse	No
Yamato et al.	2000	-		Gaze	Hardware Button	Pointer	Several techniques to enable gaze interaction under Windows.	-
		-	Objects	Gaze	Hardware Button	Pointer		
		-		Gaze, Mouse	Hardware Button	Pointer		
Partala et al.	2001	-	-	Gaze	EMG	Unknown	Using muscle activity measured with EMG to trigger selection.	-
Miniotas et al.	2004	Expansion	-	Gaze	Dwell	Highlight	Target expansion and grab-and-hold algorithm.	No
Ashmore & Duchowski	2005	Fisheye	-	Gaze	Dwell	Pointer	Magnification of focus area through a fisheye.	-
Špakov & Miniotas	2005	Expansion	-	Gaze	Dwell	Highlight	Test of expansion of menu items.	-
Miniotas et al.	2006	-	-	Gaze	Dwell, Speech	Highlight	Color coded targets, selectable by pronouncing respective color.	-
Kumar et al.	2007	Zoom	-	Gaze	Hardware Button	None	Look-press-look-release selection process. Magnification of focus area through zooming.	-
Zhang & MacKenzie	2007	-	-	Gaze	Dwell Hardware Button	None	ISO 9241-9 compliant test.	-
Laqua et al	2007		Objects	Gaze	Dwell	Highlight	Test of dwelling algorithms.	

Appendix D: Summary of Fitts' Law based Evaluations

S = Number of Subjects, Task layout (M = Multidirectional, O = Onedirectional), Task Type (D = Discrete, C = Continuous)

Authors	Year	Devices	Task Layout	Task Type	S	Practice Phase	Post-hoc Tests for Practice	IP-Calculation	Notes
Card et al.	1978	Mouse Isometric Joystick Step Keys Text Keys	M	D	4	MT within 5% comparing trials of first and last third of a 600 trial block.	Regression analysis (Power Law)	Welford	
Jagacinski & Monk	1985	Joystick Helmet	M	D	8	MT within 3.5% of 4 day mean (6-29 days)	No	Fitts	
Mithal & Douglas	1996	Isometric Joystick Mouse	O	D	6	No	Differences between blocks of trials (pair-wise t-tests)	Unknown	Analysis of jitter (spectogram)
MacKenzie & Oniszczak	1998	Touchapd (3 selection tech.)	O	D	12	One block of practice (180 trials)	Presence of effect of blocks of trials (ANOVA)	Shannon	
Douglas et al.	1999	Isometric Joystick Touchpad	O/M	C/D	24	Multidirectional task prior to onedirectional task served as a learning task (720 trials)	Differences between blocked multidirectional trials (Helmert Contrasts)	Shannon	Between-subjects design
Silfverberg et al.	2001	Isometric joysticks (3 devices and 2 selection tech.)	M	D	12	One minute practice phase	Presence of effect of blocks of trials (ANOVA)	Shannon	
MacKenzie & Jusoh	2001	GyroPoint Desktop, GyroPoint Remote, RemotePoint, Mouse	O	C	12	One block of practice (180 trials)	No	Shannon	
MacKenzie et al.	2001	Mouse Trackball Joystick Touchpad	M	C	12	One sequence of practice (15 trials)	Differences between blocks of trials (Helmert Contrasts)	Shannon	
Isokoski & Raisamo	2002	Optical Mice (3) Mechano-Optical Mice (3)	O	C	12	Two or three 20-click tasks with each device	No	Shannon	Calculation of effective amplitude
Oh & Stuerzlinger	2002	Laserpointer Mouse	M	C	12	Each task/device three times or more	Differences between blocks of trials (ANOVA)	Shannon	Use of laserpointer on small and large screen (additional condition)
Zhai et al.	2003	Mouse (three styles of target expansion)	O	D	12	No	Presence of effect of blocks of trials (ANOVA)	Shannon	
Poupyrev et al.	2004	Tablet (with and without tactile feedback)	O	C	12	Training phase	No	Shannon	
Isokoski et al.	2007	Trackball-Mouse Mouse and Trackball Mouse	M	C	12	No	Presence of effect of repeated sessions (ANOVA)	Shannon	(Exp. 1) Two sessions held on consecutive days

Works Cited

Accot, J., & Zhai, S. (2002). More than dotting the i's - foundations for crossing-based interfaces. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 73-80.

Albinsson, P. A., & Zhai, S. (2003). High precision touch screen interaction. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 105-112.

Anderson, J. R. (2007). *Kognitive Psychologie* (6th ed.). Berlin: Springer-Verlag.

Apitz, G., & Guimbretière, F. (2004). CrossY: a crossing-based drawing application. *Proceedings of the 17th annual ACM symposium on User interface software and technology*, 3-12.

Ashmore, M., Duchowski, A. T., & Shoemaker, G. (2005). Efficient Eye Pointing with a FishEye Lens. *Proceedings of the 2005 conference on Graphics interface*, 203-210.

Atkinson, R. C., & Shiffrin, R. M. (1967). Human Memory: A Proposed System And Its Control Processes. In J. T. Spence, K. W. Spence & G. H. Bower (Eds.), *The Psychology of Learning and Motivation*: Academic Press.

Babcock, J. S., & Pelz, J. B. (2004). Building a lightweight eyetracking headgear. *Proceedings of the Symposium on Eye Tracking Research & Applications*, 109-114.

Bedford, F. L. (1994). Of computer mice and men. *Cahiers de Psychologie Cognitive/Current Psychology of Cognition, 13*(4), 405-426.

Benko, H., Wilson, A. D., & Baudisch, P. (2006). Precise selection techniques for multi-touch screens. *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 1263-1272.

Bock, R. D. (1975). *Multivariate Statistical Methods in Behavioral Research*. New York, USA: McGraw-Hill.

Bolt, R. A. (1981). Gaze-orchestrated dynamic windows. *Proceedings of the 8th annual conference on Computer graphics and interactive techniques*, 109-119.

Bolt, R. A. (1984). *The Human Interface*. Belmont, CA, USA: Lifetime Learning Publications.

Britton, E. G., Lipscomb, J. S., & Pique, M. E. (1978). Making nested rotations convenient for the user. *Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, 222-227.

Bruce, C. J., & Friedman, H. R. (2002). Eye Movements. In V. S. Ramachandran (Ed.), *Encyclopedia of the Human Brain*. Boston: Academic Press.

Buxton, W. (1983). Lexical and pragmatic considerations of input structures. *ACM SIGGRAPH Computer Graphics, 17*(1), 31-37.

Buxton, W., & Myers, B. (1986). A study in two-handed input. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 321-326.

Card, S. K., English, W. K., & Burr, B. J. (1978). Evaluation of Mouse, Rate-Controlled Isometric Joystick, Step Keys, and Text Keys for Text Selection on a CRT. *Ergonomics, 21*(8), 601-613.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, New Jersey, USA: Lawrence Erlbaum Associates.

Cockburn, A., & Brock, P. (2006). Human on-line response to visual and motor target expansion. *Proceedings of the 2006 conference on Graphics interface*, 81-87.

Cornsweet, T. N. (1974). *Visual perception* (3. print. ed.). New York u.a.: Academic Press.

Crossman, E. R. F. W. (1959). A Theory of Acquisition of Speed-Skill. *Ergonomics, 2*, 153-166.

Davis, J., & Chen, X. (2002). Lumipoint: multi-user laser-based interaction on large tiled displays. *Displays, 23*(5), 205-211.

Douglas, S. A., Kirkpatrick, A. E., & MacKenzie, I. S. (1999). Testing pointing device performance and user assessment with the ISO 9241, Part 9 standard. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 215-222.

Douglas, S. A., & Mithal, A. K. (1997). *The Ergonomics of Computer Pointing Devices*. London, UK: Springer.

Duchowski, A. (2003). *Eye Tracking Methodology: Theory and Practice*. London, UK: Springer.

Duchowski, A. (2007). *Eye Tracking Methodology: Theory and Practice* (2nd ed.). London, UK: Springer.

Duncan, D. B. (1955). Multiple Range and Multiple F Tests. *Biometrics, 11*(1), 1-42.

Findlay, J. M., & Gilchrist, I. D. (2003). *Active Vision: The Psychology of Looking and Seeing*. Oxford, UK: Oxford University Press.

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

Fitts, P. M., Jones, R. E., & Milton, J. L. (1950). Eye Movements of Aircraft Pilots During Instrument-Landing Approaches. *Aeronautical Engineering Review, 9*(2), 24-29.

Fitts, P. M., & Peterson, J. R. (1964). Information capacity of discrete motor responses. *Journal of Experimental Psychology, 67*(2), 103-112.

Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. (1997). *Computer Graphics: Principles and Practice* (2nd ed.). Reading, Massachusetts, USA: Addison-Wesley.

Foley, J. D., & Wallace, V. L. (1974). The art of natural graphic man—Machine conversation. *Proceedings of the IEEE, 62*(4), 462-471.

Foley, J. D., Wallace, V. L., & Chan, P. (1984). The human factors of computer graphics interaction techniques. *IEEE Computer Graphics and Applications, 4*(11), 13-48.

Fono, D., & Vertegaal, R. (2005). *EyeWindows: evaluation of eye-controlled zooming windows for focus selection*. Paper presented at the SIGCHI Conference on Human Factors in Computing Systems.

Frolov, P., Matveyev, S., Göbel, M., & Klimenko, S. (2002). Using Kalman Filter for Natural Hand Tremor Smoothing during the Interaction with the Projection Screen. *Workshop Proceedings VEonPC*, 94-101.

Goldberg, D., & Richardson, C. (1993). Touch-Typing With a Stylus. *Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, 80-87.

Goldberg, J. H., & Wichansky, A. M. (2003). Eye Tracking in Usability Evaluation: A Practitioner's Guide. In R. R. Hyönä & H. Deubel (Eds.), *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research* (pp. 495-516). Amsterdam: Elsevier.

Goldberg, M. E. (2000). The control of gaze. In E. R. Kandel, J. H. Schwartz & T. M. Jessell (Eds.), *Principles of Neural Science. McGraw-Hill* (4th ed., pp. 782-800). New York: McGraw-Hill.

Goldstein, E. B. (2002). *Sensation and Perception* (6th ed.). Pacific Grove, California, USA: Wadsworth Publishing Company.

GSPC. (1979). Status report of the Graphics Standards Planning Committee. *Computer Graphics, 13*(3).

Guiard, Y., Blanch, R., & Beaudouin-Lafon, M. (2004). Object pointing: a complement to bitmap pointing in GUIs. *Proceedings of the 2004 conference on Graphics interface*, 9-16.

Haberlandt, K. (1999). *Human Memory: Exploration and Application*. Needham Heights, MA, USA: Allyn and Bacon.

Herholz, S., Tanner, T. G., Canto-Pereira, L. H., Fleming, R. W., & Bülthoff, H. H. (2007). *Real-time gaze-tracking for freely-moving observers.* Paper presented at the European Conference on Eye Movements ECEM, Potsdam.

Hick, W. E. (1952). On the Rate of Gain of Information. *Quarterly Journal of Experimental Psychology, 4*, 11-26.

Hinckley, K. (2002). Input technologies and techniques. In J. Jacko & A. Sears (Eds.), *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications* (pp. 151-168). Mahwah, New Jersey, USA: Lawrence Erlbaum Associates.

Hinckley, K. (2006). Input technologies and techniques (Draft of revision of chapter in Handbook of Human-Computer Interaction, ed. by A. Sears & J. Jacko.) [Electronic Version]. Retrieved May 30, 2007.

Hopgood, F. R. A., Duce, D. A., Gallop, J. R., & Sutcliffe, D. C. (1986). *Introduction to the Graphical Kernel System (GKS)(revised for international standard)*. London, UK: Academic Press.

Hussy, W., & Jain, A. (2002). *Experimentelle Hypothesenprüfung in der Psychologie*. Göttingen: Hogrefe.

Hutchins, E. L., Hollan, J. D., & Norman, D. A. (1985). Direct Manipulation Interfaces. *Human-Computer Interaction, 1*.

Hyman, R. (1953). Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology, 45*(3), 188-196.

Hyrskykari, A., Majaranta, P., Aaltonen, A., & Räihä, K. J. (2000). Design issues of iDICT: a gaze-assisted translation aid. *Proceedings of the Symposium on Eye Tracking Research & Applications*, 9-14.

ISO 7942:1985. Information processing systems -- Computer graphics -- Graphical Kernel System (GKS) functional description.

ISO 9241-9:2000. Ergonomic requirements for office work with visual display terminals (VDTs) — Part 9: Requirements for non-keyboard input devices.

ISO 9241-11:1998. Ergonomic requirements for office work with visual display terminals (VDTs) — Part 9: Guidance on usability.

ISO 9241-16:1999. Ergonomic requirements for office work with visual display terminals (VDTs) — Part 16: Direct manipulation dialogues.

Isokoski, P., & Raisamo, R. (2002). Speed-accuracy measures in a population of six mice. *Proceedings of APCHI*, 765-777.

Isokoski, P., Raisamo, R., Martin, B., & Evreinov, G. (2007). User performance with trackball mice. *Interacting with Computers, 19*(3), 407-427.

Jacob, R. J. K. (1991). The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Transactions on Information Systems, 9*(2), 152-169.

Jacob, R. J. K., Deligiannidis, L., & Morrison, S. (1999). A software model and specification language for non-WIMP user interfaces. *ACM Transactions on Computer-Human Interaction (TOCHI), 6*(1), 1-46.

Jacob, R. J. K., & Karn, K. S. (2003). Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises. In R. R. Hyönä & H. Deubel (Eds.), *The Mind's*

Eye: Cognitive and Applied Aspects of Eye Movement Research (pp. 573-605). Amsterdam: Elsevier.

Jagacinski, R. J., & Monk, D. L. (1985). Fitts'Law in two dimensions with hand and head movements. *Journal of Motor Behavior, 17*(1), 77-95.

Jellinek, H. D., & Card, S. K. (1990). *Powermice and user performance.* Paper presented at the SIGCHI conference on Human factors in computing systems.

Joos, M., Rötting, M., & Velichkovsky, B. M. (2003). Die Bewegungen des menschlichen Auges: Fakten, Methoden, innovative Anwendungen. In G. H. Rickheit, T.; Deutsch, W. (Ed.), *Psycholinguistik. Ein internationales Handbuch.* Berlin: De Gruyter.

Just, M. A., & Carpenter, P. A. (1976). Eye Fixations and Cognitive Processes. *Cognitive Psychology, 8*, 441-480.

Keele, S. W. (1968). Movement control in skilled motor performance. *Psychological Bulletin, 70*(6), 387-403.

Kirstein, C., & Müller, H. (1998). Interaction with a projection screen using a camera-tracked laserpointer. *Multimedia Modeling Proceedings*, 191-192.

Kolakowski, S. M., & Pelz, J. B. (2006). Compensating for eye tracker camera movement. *Proceedings of the symposium on Eye tracking research & applications*, 79-85.

König, W. A., Bieg, H. J., & Reiterer, H. (2007). Laserpointer-Interaktion für große, hochauflösende Displays. *Proceedings of the Mensch & Computer*, 69-78.

König, W. A., Bieg, H. J., Schmidt, T., & Reiterer, H. (2007). Position-independent interaction for large high-resolution displays. *Proceedings of IADIS International Conference on Interfaces and Human Computer Interaction 2007*, 117-125.

Kumar, M., Paepcke, A., & Winograd, T. (2007). EyePoint: Practical Pointing and Selection Using Gaze and Keyboard. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 421-430.

Landauer, T. K., & Nachbar, D. W. (1985). Selection from alphabetic and numeric menu trees using a touch screen: breadth, depth, and width. *ACM SIGCHI Bulletin, 16*(4), 73-78.

Lankford, C. (2000). Effective eye-gaze input into Windows. *Proceedings of the Symposium on Eye Tracking Research & Applications*, 23-27.

Laqua, S., Bandara, S. U., & Sasse, M. A. (2007). GazeSpace: Eye Gaze Controlled Content Spaces. *Proceedings of the 21st BCS HCI Group Conference*.

Li, D., Babcock, J., & Parkhurst, D. J. (2006). openEyes: a low-cost head-mounted eye-tracking solution. *Proceedings of the 2006 symposium on Eye tracking research & applications*, 95-100.

MacKenzie, I. S. (1991). *Fitts' law as a performance model in human-computer interaction.* University of Toronto, Toronto, Ontario, Canada.

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. In W. Barfield & T. A. Furness (Eds.), *Virtual Environments and Advanced Interface Design*. Oxford, UK: Oxford University Press.

MacKenzie, I. S., & Buxton, W. (1992). Extending Fitts' law to two-dimensional tasks. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 219-226.

MacKenzie, I. S., & Jusoh, S. (2001). An evaluation of two input devices for remote pointing. *Proceedings of the 8th IFIP International Conference on Engineering for Human-Computer Interaction*, 235-250.

MacKenzie, I. S., Kauppinen, T., & Silfverberg, M. (2001). Accuracy measures for evaluating computer pointing devices. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 9-16.

MacKenzie, I. S., & Oniszczak, A. (1998). A comparison of three selection techniques for touchpads. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 336-343.

MacKenzie, I. S., & Soukoreff, R. W. (2003). *Card, English, and Burr (1978): 25 years later.* Paper presented at the SIGCHI Conference on Human Factors in Computing Systems.

MacKenzie, I. S., & Ware, C. (1993). Lag as a determinant of human performance in interactive systems. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 488-493.

Mackenzie, I. S., Zhang, S. X., & Soukoreff, R. W. (1999). Text entry using soft keyboards. *Behaviour & Information Technology, 18*(4), 235-244.

Majaranta, P., & Räihä, K. J. (2002). Twenty years of eye typing: systems and design issues. *Proceedings of the symposium on Eye tracking research & applications*, 15-22.

Marras, W. S. (2007). Basic Biomechanics and Workstation Design. In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics* (pp. 340-370). Hoboken, NJ, USA: John Wiley & Sons.

Martin, D. W. (2004). *Doing Psychology Experiments* (6th ed.). Belmont, CA, USA: Wadsworth.

Matveyev, S. V., & Göbel, M. (2003). The optical tweezers: multiple-point interaction technique. *ACM symposium on Virtual reality software and technology*, 184-187.

McGuffin, M., & Balakrishnan, R. (2002). Acquisition of expanding targets. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 57-64.

Microsoft Corp. (2002). Pointer Ballistics for Windows XP [Electronic Version]. Retrieved Dec. 16, 2007 from http://www.microsoft.com/whdc/device/input/pointer-bal.mspx.

Miniotas, D. (2000). *Application of Fitts' law to eye gaze interaction.* Paper presented at the Conference on Human Factors in Computing Systems.

Miniotas, D., & Špakov, O. (2004). An Algorithm to Counteract Eye Jitter in Gaze-Controlled Interfaces. *Information Technology and Control, 1*(30), 65-68.

Miniotas, D., Špakov, O., & MacKenzie, I. S. (2004). Eye gaze interaction with expanding targets. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1255-1258.

Miniotas, D., Špakov, O., Tugoy, I., & MacKenzie, I. S. (2006). Speech-augmented eye gaze interaction with small closely spaced targets. *Proceedings of the Symposium on Eye tracking Research & Applications*, 67-72.

Mithal, A. K., & Douglas, S. A. (1996). Differences in movement microstructure of the mouse and the finger-controlled isometric joystick. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 300-307.

Myers, B. A. (1989). User-interface tools: introduction and survey. *IEEE, Software, 6*(1), 15-23.

Myers, B. A., Bhatnagar, R., Nichols, J., Peck, C. H., Kong, D., Miller, R., et al. (2002). Interacting at a Distance: Measuring the Performance of Laser Pointers and Other Devices. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Oh, A., Fox, H., Van Kleek, M., Adler, A., Gajos, K., Morency, L. P., et al. (2002). Evaluating look-to-talk: a gaze-aware interface in a collaborative environment. *Proceedings of the conference on Human factors in computing systems*, 650-651.

Oh, J. Y., & Stuerzlinger, W. (2002). Laser pointers as collaborative pointing devices. *Proceedings of the Graphics Interface, 2002*, 141-149.

Olsen Jr., D. R., & Nielsen, T. (2001). Laser pointer interaction. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 17-22.

Olson, J. R., & Nilsen, E. (1987). Analysis of the Cognition Involved in Spreadsheet Software Interaction. *Human-Computer Interaction, 3*(4), 309-349.

Partala, T., Aula, A., & Surakka, V. (2001). Combined Voluntary Gaze Direction and Facial Muscle Activity as a New Pointing Technique. *Proceedings of the Interact*, 100-107.

Posner, M. I. (1980). Orienting of attention. *The Quarterly Journal of Experimental Psychology, 32*(1), 3-25.

Potter, R. L., Weldon, L. J., & Shneiderman, B. (1988). Improving the accuracy of touch screens: an experimental evaluation of three strategies. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 27-32.

Reis, H. T., & Judd, C. M. (2000). *Handbook of Research Methods in Social and Personality Psychology*. Cambridge, United Kingdom: Cambridge University Press.

Salvucci, D. D. (1999). *Mapping Eye Movements to Cognitive Processes.* Carnegie Mellon University.

Salvucci, D. D., & Anderson, J. R. (2000). Intelligent gaze-added interfaces. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 273-280.

Salvucci, D. D., & Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. *Proceedings of the symposium on Eye tracking research & applications*, 71-78.

Schmidt, R. A., & Lee, T. D. (2005). *Motor Control and Learning: a behavioral emphasis* (4th ed.). Champaign, IL, USA: Human Kinetics.

Searle, L. V., & Taylor, F. V. (1948). Studies of tracking behavior: 1. Rate and time characteristics of simple corrective movements. *Journal of Experimental Psychology, 38*, 615-631.

Sears, A., & Shneiderman, B. (1991). High Precision Touchscreens: Design Strategies and Comparisons with a Mouse. *International Journal of Man-Machine Studies, 34*(4), 593-613.

Seow, S. C. (2005). Information Theoretic Models of HCI: A Comparison of the Hick-Hyman Law and Fitts' Law. *Human–Computer Interaction, 20*, 315–352.

Shannon, C. E., & Weaver, W. (1949). *The mathematical theory of communication*. Urbana, IL, USA: University of Illinois Press.

Sibert, L. E., & Jacob, R. J. K. (2000). Evaluation of eye gaze interaction. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 281-288.

Silfverberg, M., MacKenzie, I. S., & Kauppinen, T. (2001). An isometric joystick as a pointing device for handheld information terminals. *Proceedings of Graphics Interface*, 119-126.

Snoddy, G. S. (1926). Learning and stability. *Journal of Applied Psychology, 10*(1), 1-36.

Soukoreff, R. W., & MacKenzie, I. S. (1995). *Generalized Fitts' law model builder.* Paper presented at the SIGCHI Conference on Human Factors in Computing Systems.

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.

Špakov, O., & Miniotas, D. (2005). Gaze-based selection of standard-size menu items. *Proceedings of the 7th international conference on Multimodal interfaces*, 124-128.

Stampe, D. M., & Reingold, E. M. (1995). Selection by looking: A novel computer interface and its application to psychological research. In *Eye Movement Research: Mechanisms, Processes, and Applications* (pp. 467-478). Amsterdam: Elsevier.

Starker, I., & Bolt, R. A. (1990). A gaze-responsive self-disclosing display. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 3-10.

Surakka, V., Illi, M., & Isokoski, P. (2003). Voluntary eye movements in human–computer interaction. In J. Hyönä, R. Radach & H. Deubel (Eds.), *The Mind's Eyes: Cognitive and Applied Aspects of Oculomotor Research* (pp. 473-491). Oxford: Elsevier Science.

The New York Times. (2006). Palm to Settle With Xerox on Patent Infringement [Electronic Version]
from http://www.nytimes.com/2006/06/29/technology/29palm.html?_r=1&oref=slogin.

Velichkovsky, B. M., & Hansen, J. P. (1996). New technological windows into mind: there is more in eyes and brains for human-computer interaction. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 496-503.

Wallace, V. L. (1976). The semantics of graphic input devices. *The papers of the ACM symposium on Graphic languages table of contents*, 61-65.

Ware, C., & Mikaelian, H. H. (1987). An evaluation of an eye tracker as a device for computer input. *Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface*, 183-188.

Welford, A. T. (1968). *Fundamentals of skill*. London, UK: Methuen.

Wissen, M., Wischy, M. A., & Ziegler, J. (2001). Realisierung einer laserbasierten Interaktionstechnik für Projektionswände. *Proceedings of the Mensch & Computer*, 135-143.

Worden, A., Walker, N., Bharat, K., & Hudson, S. (1997). Making computers easier for older adults to use: area cursors and sticky icons. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 266-271.

Yamato, M., Monden, A., Matsumoto, K., Inoue, K., & Torii, K. (2000). Button selection for general GUIs using eye and hand together. *Proceedings of the working conference on Advanced visual interfaces AVI*, 270–273.

Young, L. R., & Sheena, D. (1975). Eye-movement measurement techniques. *American Psychologist, 30*(3), 315-330.

Zhai, S. (2003). What's in the Eyes for Attentive Input. *Communications of the ACM, 46*(3), 35-39.

Zhai, S. (2004). Characterizing computer input with Fitts' law parameters—the information and non-information aspects of pointing. *International Journal of Human-Computer Studies, 61*(6), 791-809.

Zhai, S., Conversy, S., Beaudouin-Lafon, M., & Guiard, Y. (2003). Human on-line response to target expansion. *Proceedings of the conference on Human factors in computing systems*, 177-184.

Zhai, S., Kong, J., & Ren, X. (2004). Speed–accuracy tradeoff in Fitts' law tasks—on the equivalency of actual and nominal pointing precision. *International Journal of Human-Computer Studies, 61*(6), 823-856.

Zhai, S., Morimoto, C., & Ihde, S. (1999). Manual And Gaze Input Cascaded (MAGIC) Pointing. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 246-253.

Zhai, S., Smith, B. A., & Selker, T. (1997). Improving Browsing Performance: A Study of Four Input Devices for Scrolling and Pointing Tasks. *Proceedings of the IFIP TC13 Interantional Conference on Human-Computer Interaction*, 286-293.

Zhang, X., & MacKenzie, I. S. (2007). Evaluating Eye Tracking with ISO 9241-Part 9. *Lecture Notes in Computer Science, 4552*, 779-788.

Table of Figures

Tables