Eye Gaze Tracking for Human Computer Interaction

Heiko Drewes

Dissertation
an der LFE Medien-Informatik
der Ludwig-Maximilians-Universität
München

München 2010

Erstgutachter:	Professor Dr. Heinrich Hußmann
Zweitgutachter:	Professor Dr. Albrecht Schmidt
Externer Gutachter:	Professor Dr. Alan Dix
Tag der mündlichen Prüfung:	18.3.2010

Abstract

With a growing number of computer devices around us, and the increasing time we spend for interacting with such devices, we are strongly interested in finding new interaction methods which ease the use of computers or increase interaction efficiency. Eye tracking seems to be a promising technology to achieve this goal.

This thesis researches interaction methods based on eye-tracking technology. After a discussion of the limitations of the eyes regarding accuracy and speed, including a general discussion on Fitts' law, the thesis follows three different approaches on how to utilize eye tracking for computer input. The first approach researches eye gaze as pointing device in combination with a touch sensor for multimodal input and presents a method using a touch sensitive mouse. The second approach examines people's ability to perform gestures with the eyes for computer input and the separation of gaze gestures from natural eye movements. The third approach deals with the information inherent in the movement of the eyes and its application to assist the user. The thesis presents a usability tool for recording of interaction and gaze activity. It also describes algorithms for reading detection.

All approaches present results based on user studies conducted with prototypes developed for the purpose.

Zusammenfassung

Mit einer wachsenden Zahl von Computergeräten um uns herum und zunehmender Zeit die wir mit der Bedienung dieser Geräte zubringen, haben wir ein großes Interesse daran, Interaktionsmethoden zu finden, welche die Benutzung der Computer erleichtern oder effizienter machen. Blickverfolgung scheint eine viel versprechende Technologie zu sein um dieses Ziel zu erreichen.

Die vorliegende Arbeit untersucht Interaktionsmethoden, die auf Blickverfolgertechnik beruhen. Nach einer Diskussion der Beschränkungen des Auges in Bezug auf Genauigkeit und Geschwindigkeit, die eine generelle Diskussion des Fitts' Law enthält, verfolgt die Arbeit drei verschiedene Ansätze wie Blickverfolgung für Computereingaben benutzt werden kann. Der erste Ansatz untersucht den Blick als Zeigegerät in Kombination mit einem Berührungssensor für multimodale Eingabe und stellt eine Methode mit einer berührungsempfindlichen Maus vor. Der zweite Ansatz untersucht die Fähigkeit von Menschen Gesten mit den Augen für Computereingaben durchzuführen und wie diese Blickgesten von natürlichen Augenbewegungen unterschieden werden können. Der dritte Ansatz beschäftigt sich mit der Information, die den Augenbewegungen entnommen werden kann, und ihrer Anwendung zur Unterstützung der Benutzer. Es wird ein Usability-Werkzeug zum Aufzeichnen von Interaktions- und Blickaktivität vorgestellt. Außerdem werden Algorithmen zur Leseerkennung beschrieben.

Für alle Ansätze werden Ergebnisse präsentiert, die auf durchgeführten Benutzerstudien mit speziell entwickelten Prototypen beruhen.

Eye Gaze Tracking for Human Computer Interaction

1 Introduction

With the invention of the computer in the middle of the last century there was also the need of an interface for users. In the beginning experts used teletype to interface with the computer. Due to the tremendous progress in computer technology in the last decades, the capabilities of computers increased enormously and working with a computer became a normal activity for nearly everybody. With all the possibilities a computer can offer, humans and their interaction with computers are now a limiting factor. This gave rise to a lot of research in the field of HCI (human computer interaction) aiming to make interaction easier, more intuitive, and more efficient. Interaction with computers is not limited to keyboards and printers anymore. Different kinds of pointing devices, touch-sensitive surfaces, high-resolution displays, microphones, and speakers are normal devices for computer interaction nowadays. There are new modalities for computer interaction like speech interaction, input by gestures or by tangible objects with sensors. A further input modality is eye gaze which nowadays finds its application in accessibility systems. Such systems typically use eye gaze as the sole input, but outside the field of accessibility eye gaze can be combined with any other input modality. Therefore, eye gaze could serve as an interaction method beyond the field of accessibility. The aim of this work is to find new forms of interactions utilizing eye gaze and suitable for standard users.

1.1 Motivation

Nowadays most eye-tracking systems work on video-based pupil detection and a reflection of an infrared LED. Video cameras became cheap over the last years and the price for a LED is negligible. Many computer devices come already with built-in cameras, such as mobile phones, laptops, and displays. Processor power still increases steadily and standard processors are powerful enough to process the video stream necessary to do eye tracking, at least on desktop and laptop computers. Head-tracking systems, which are necessary to give the users the freedom to move in front of their display, are also video-based. Such systems can be implemented unobtrusively with a second camera. If produced for the mass market, a future standard eye tracker should not cost much more than an optical mouse or a webcam today.

Some people interact with the computer all day long, for their work and in their leisure time. As most interaction is done with keyboard and mouse, both using the hands, some people suffer from overstressing particular parts of their hands, typically causing a carpal tunnel syndrome. With a vision of ubiquitous computing, the amount of interaction with computers will increase and we are in need of interaction techniques which do not cause physical problems. The eyes are a good candidate because they move anyway when interacting with computers. Using the information lying in the eye movements could save some interaction, in particular hand-based interaction.

Finding interface techniques utilizing the ease and speed of eye movements and giving the users some benefit in terms of computer interaction would establish eye tracking as additional interaction method for the future.

1.2 Eye Gaze and Human Communication

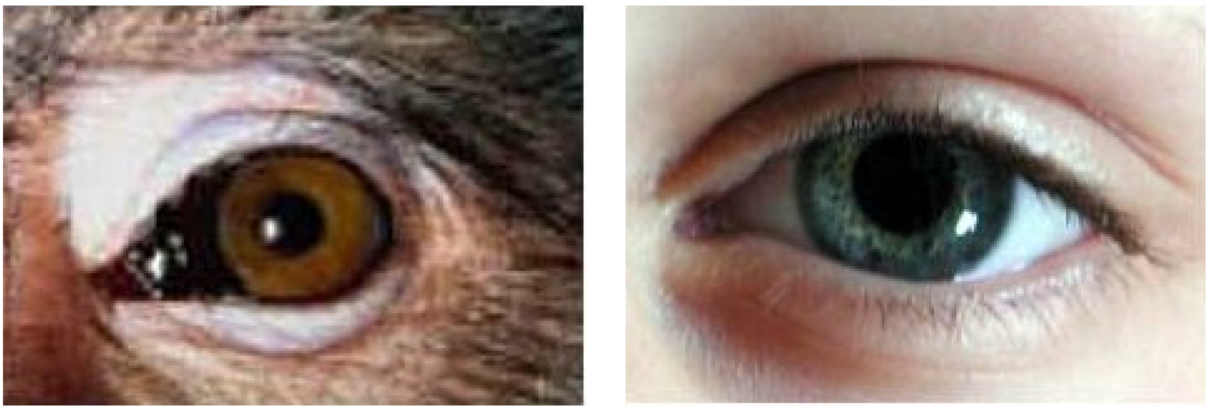
When we ask "What is this?" we often reference the object with the direction of our gaze and we assume that the person asked knows which object we are looking at. Bolt, a pioneer of gaze interaction research, expressed it this way:

'*Consider the case of asking the question "What is your favorite sport?" in the presence of several people, but looking at Mary, say, not at Frank, Judy, or Dave. Then you utter the selfsame words, but now looking at Dave. The question is a different question; the difference lies not in its verbal component, but in its intended addressee as given by eye.*' [Bolt 1982].

When we talk to somebody we normally address the person by looking at her or him, but staring at somebody without reason is not polite and should be avoided. These small examples on how we use the eyes for communication illustrate that our eyes move not only to enable vision. We are aware that others are aware where we are looking to and for this reason we need to have our eyes under control.

When comparing human with animal eyes the white in the human eyes is very distinctive (see Figure 1). In animal eyes, eyes of mammals in particular, which work in a similar way as human eyes, no white eyeball is visible. For this reason it is more difficult to determine the direction of an animal's gaze than of a human's. What role the awareness of the gaze's direction played in the evolution of our species is a field for speculation. What is certain is that we use our eyes for communication. Eibl-Eibesfeldt discusses the topic [Eibl-Eibesfeldt 2004, p. 621] and the meaning of eye contact in general. He observed that looking at somebody opens a communication channel, but all ethnics and cultures interpret staring at somebody as an act of aggression.

Figure 1: The eye of a chimpanzee and a human eye (taken from [Milekic 2003])



The eye tracking studies of Vertegaal et al. showed that when someone is listening or speaking to individuals there is indeed a high probability that the person being looked at is the person listening (p=88%) or spoken to (p=77%) [Vertegaal, Slagter, van der Veer, Nijholt 2001].

It is also worth mentioning that the cultural background seems to have influence on the movement of the eyes. Chua et al. [Chua, Boland, Nisbett 2005] presented a study comparing scene perception of Americans and Chinese. They state that Americans fixate sooner and longer on the foreground objects whereas Chinese looked more at the background. Even without a scientific study to refer to it is obvious that also the formation of a person has influence on the motion of the eyes. Presenting texts in different scripts, for example in Latin, Farsi, Hindi and Chinese characters, will produce a different eye response depending on whether the person is able to read the script. Eye movements involve high levels of cognition and depend on the current intention of a person, i.e. what she or he is looking for. Despite some universal aspects resulting from biology and evolution, the eye movements depend on very personal aspects. Every person sees the world with her or his own eyes.

The fact that we use the eyes for human-human communication leads to following conclusions for the field of HCI:

- We have the ability to control our eyes and use it for output

  When we look at a person's face to indicate we are talking to that particular person we use the eyes for output. To say it with the words of Bolt:

  "*At the user/observer interface level, interactivity is extended to incorporate where the user is looking, making the eye an output device*" [Bolt 1981].

  However, there are also sceptical voices on the use of the eye for output as expressed by Zhai et al.:

  "*We believe that many fundamental limitations exist with traditional gaze pointing. In particular, it is unnatural to overload a perceptual channel such as vision with a motor control task.*" [Zhai, Morimoto, Ihde 1999].

- For natural communication with computers the computer needs to know our gaze direction

  For a vision of natural interaction with computer devices in the same style as we interact with other people an eye-tracking technology is mandatory. For example, if we interact with the computer using a speech interface, the computer has to know whether we are talking to it or with our human neighbour. The social protocol for this is based on gaze direction. It would be impossible to integrate the computer into that protocol if it was not aware of it.

- Analysing the eye gaze reveals information on the person

  A person's eyes react differently to offered information according to the person's mood, intention, and life experience. This information would help a lot for the goal of smart computer interfaces which are able to assist the user on a personal level.

1.3 Eye Gaze for Computer Input

An eye-gaze interface seems to be a promising candidate for a new interface technique, which may be more convenient than the ones we use. Traditionally, disabled people who cannot move anything except their eyes use eye gaze interaction. These systems are designed to direct the computer solely by the eyes. Such systems work well and are a great help for people who need them, but for others they are cumbersome and less efficient than keyboard and mouse. This contradicts the fact that looking is an easy task and that eye movements are fast. Consequently, eye-gaze interfaces for the masses need a different design to bring benefit to the average user.

An eye-gaze interface might offer several potential benefits:

- Ease of use

  A benefit of eye tracking could be reduced stress for hand and arm muscles by transferring the computer input from the hand to the eyes. This need not necessarily put extra load on the eye muscles because for most interactions the eyes move anyway. For example, when clicking a button on the screen in most cases mouse and eyes move to the target (see 4.4.2 for a detailed discussion).

- Interaction speed-up

  Eye-tracking interfaces could speed up the interaction, as the eyes are quick. Although existing eye-typing systems are slower than traditional keyboard input, the combination of eye gaze with another input modality can provide fast interaction.

- Maintenance free

  Video-based eye tracking works contact free which means that no maintenance is necessary. There is no need to clean the device, which is a typical problem for keyboards and mouse devices. Placing the camera behind strong transparent material results in a vandalism-proofed interface, which is nearly impossible to realize for keyboards and mouse devices.

- Hygienic interface

  In environments with high hygienic demands, like an operation room for surgery, an eye-gaze interface would be useful because it allows interacting without anything to touch. Also for public interfaces, especially in times of pandemic threats, hygienic interaction is desirable.

- Remote control

  Another benefit resulting from eye tracking is possible remote control. Zoom lenses and high-resolution cameras make eye gaze detection possible over some meters of distance. Even the low-cost eye tracker used in the user studies of this thesis offers one meter distance which is longer than an arm length.

- Safer interaction

  Eye tracking not only guarantees the presence of a person but also his or her attention. A mobile phone, which requires eye contact for an outgoing call, will not call somebody because of accidentally pressed buttons while in a pocket. Eye tracking can ensure certain behaviour of the users. For example, a system can demand that a warning text is read before allowing the user to continue with other functions.

- More information on the users activity

The eyes tell a lot about what somebody is doing. Tracking the eyes provides useful information for context-aware systems. In the simplest form an eye tracker tells where the attention is, which already has a big potential for the implementation of context-awareness. Simple analysis of the eye tracker data can detect activities like reading. Analysis that is more sophisticated could reveal the physical or emotional condition of a user, her or his age, and degree of literacy.

Of course, there are also possible problems.

- Ability to control

The eyes perform unconscious movements and this might disturb their use as computer input. It is not clear to which degree people are able to control the movement of their eyes. The ability to control the eyes consists of both suppressing unintended movements and performing intended eye movements. It seems that we are at least able to control where we look because this is required by our social protocols. However, it is not clear whether we can train the motor skills of the eye muscles to the same extent as we can train the fingers for playing the piano.

- Conflict of input and vision

The primary function of the eyes is to enable vision. Using the eyes for computer input might result in conflicts. The well-known conflict is the Midas Touch problem – for the eye-gaze interface it is difficult to decide whether our gaze is on an object just for inspection or for invoking an action. Misinterpretation by the gaze interface can trigger unwanted actions wherever we look. The situation is similar when triggering actions by eye movements, i.e. gestures. The eye-gaze interface has to separate natural eye movements from intentional gaze gestures. Distraction by moving or blinking objects might also cause conflicts. The question how blinking advertisements on a web page interfere with eye gaze interaction is still a topic of research.

- Fatigue of the eye muscles

From other input devices we know that extensive use of particular muscles or muscle groups can cause physical problems called RSI (repetitive strain injury). There are fears that this might happen to the eye muscles too. The concern is justified and should be taken seriously, but as the eyes move constantly, even while we sleep, it might not turn out to be a problem.

1.4 Methods and Approach

The methods and approach used in this thesis are mostly experimental which is uncommon in computer science but appropriate in the field of HCI.

An iterative concept helps to get systematically closer to the development of new interaction methods based on gaze. Starting with a problem to solve the first step is to create ideas how to solve the problem. The resulting task is to realize the idea by creating a prototype. Typically, a pilot study gives the first hints for a redesign of the prototype and for the design of the user study. The data collected in the user study are the basis for an evaluation using statistical methods. Interviews with test users are the method to find out soft factors due to human aspects. Feeling comfortable with an interface or the fascination invoking commands by eye gaze are hard to measure by physical data. The results of a user study lead to further enhancements of the prototype and a follow-up user study. The iteration of modifying prototypes and conducting user studies leads to the final goal of understanding the implications of the tested prototype.

All prototypes used are based on an eye tracker and it's API (application programming interface). Each prototype uses software especially developed for the purpose. Additionally built hardware prototypes, such as touch-sensitive mouse devices, allow to research multimodal input.

1.5 Thesis Outline

This thesis was done at the media informatics group of LMU (Ludwig-Maximilian Universität) and belongs to the field of HCI (Human Computer Interaction). Within this field it researches how to make use of the information from an eye-gaze tracker for computer interaction. In contrast to the existing systems for disabled people, where eye-gaze input is the only input modality, this research focuses on eye gaze for regular users as an additional input modality for existing and future user interfaces. The thesis does not deal with eye-tracking hardware or low-level detection of the eye-gaze position but takes the technological basis as granted. The underlying assumption for this work is the availability of cheap and reliable gaze tracking technology in the near future. The main goal is to find new interaction methods and user interfaces, which utilize the information from an eye tracker for the benefit of the user. The thesis has seven chapters; an introduction, an overview on eye-tracking technology and general related work, four chapters on different aspects of eye-gaze interaction and a conclusion.

The introduction (this chapter) gives a motivation for the research done, discusses eye gaze as computer input, presents the methods and approaches, and finally lists the contributions to the field of research.

The overview and related work starts with a definition of eye tracking followed by its history. It describes existing systems and technologies and their application. It also presents the eye tracker used in this thesis. Furthermore, it discusses the current challenges and the scientific work done up to now on a general level. Special related work sections are part of the corresponding chapters.

The third chapter deals with the eye and its movements and presents data of two user studies. One user study reveals the limitations due to the low-level detection algorithms of the eye tracker used. The data also show the typical ranges for fixation times, saccade lengths and times as well as their statistical distribution during typical use of a computer. The other user study researches the accuracy and speed issue, and whether Fitts' law applies to eye movements.

The forth chapter researches the possibilities of eye gaze as a pointing device. Using eye gaze as a pointing device is obvious and suggests itself. This approach is conservative as it puts no additional motor task to the user but uses the intuitive motion of the eye. The chapter presents a prototype of a touch-sensitive mouse. Touching the mouse sets the mouse pointer at the gaze position. Because of the limited accuracy of the eye the mouse pointer may not hit the target but the mouse allows final positioning. Several user studies were done on eye gaze pointing, especially in combination with keys and a touch sensitive mouse.

The fifth chapter deals with the idea of directing the computer with complex eye gaze gestures. The user studies showed that people in general are able to perform complex gaze gestures at free will. The big advantage of gaze gestures is the low demand in accuracy and the absence of a calibration process. Further user studies researched possible applications for gaze gestures on mobile phones and for PIN (personal identification number) entry.

The sixth chapter sees eye gaze as context information and presents an extension of UsaProxy, a recording tool for web interaction, to additionally record gaze activity. The focus of the chapter lies on reading detection. Reading detection can make the user interface smart enough to avoid disturbing the user with popup messages while she or he is reading, or to let a message disappear automatically after it has been read. It also enables monitoring how well texts have been read or forcing compulsory reads for critical information.

The conclusion summarizes the results and discusses the potential of eye tracking. It presents some visions how gaze-aware systems of the near future might look like.

1.6 Contributions

Within an overview of the current state of research and technology this work discusses the accuracy of eye trackers and the need of a calibration process. The so-called calibration-free eye-tracking methods allow a simpler calibration process, which may not even be noticed by the user, but these methods still require calibration because of anatomic variability of individuals.

A further discussion focuses on theoretical aspects like Fitts' law and the related speed and accuracy issues. The discussion shows that there is no reason applying Fitts' law to eye movements. A general discussion of Fitts' law was published in [Drewes 2010].

The chapter on eye gaze pointing presents a touch-sensitive mouse as novel input device. Based on the observation that people have problems to locate the mouse pointer, the idea is to position the mouse pointer at the gaze position when touching the mouse. The approach leads to an improvement of Zhai's MAGIC (Mouse And Gaze Input Cascaded) pointing principle avoiding compensation methods needed for the MAGIC principle and giving the user a higher degree of intentional control. The core result of the chapter is that the use of gaze-assisted pointing saves an enormous amount of mouse movements. Recordings of mouse and gaze paths reveal that the eyes move to the target during classical pointing tasks and consequently gaze-assisted pointing does not put extra load to the eye muscles. Interviews with test users indicate that the eye movement is not perceived as a part of a motor task, which leads to a subjective feeling of higher speed for gaze-assisted pointing. The recorded data confirm a real speed benefit for gaze-assisted pointing when a user is not aware of the mouse pointer position. Theoretical considerations also predict a speed benefit for large displays. The corresponding publications are [Drewes, Schmidt 2006] and [Drewes, Schmidt 2009].

Furthermore, the thesis introduces gaze gestures as a novel method to direct computers. Gaze gestures are robust even on low accuracy and are not affected by calibration shift. Actually, gaze gestures do not need a calibration procedure and for this reason provide instant eye-gaze interaction even for different people. Gaze gestures work over a distance of several meters and could be a substitute for remote control devices. As gaze gestures do not need a further modality like a gesture key, they are a contact free input method, which provides an interface for high hygienic demands. The user studies confirm the claim that people are able to perform gaze gestures intentionally. Recordings of gaze activity during different tasks show that intentional gaze gestures are separable from unintentional natural eye movements. This thesis also presents an alphabet of four-stroke gestures, which provide 18 different gesture commands for gaze control. Further user studies on the use of gaze gestures for very small displays, as typically used for mobile phones, show that reliable separation of gaze gestures from natural eye movements requires six-stroke gestures. The major insight gained by a user study on gaze gestures for PIN (personal identification number) entry with the eyes, in combination with a gesture key, is that this input method is less error-prone than classical eye-gaze input using the dwell-time method. The corresponding publications on gaze gestures are [Drewes, Schmidt 2007] and [Drewes, Hußmann, Schmidt 2007] with the focus on the gaze gestures, and [Drewes, De Luca, Schmidt 2007] for general eye-gaze interaction on mobile phones and gaze gestures in particular and [De Luca, Weiss, Drewes 2007] for PIN (personal identification number) entry with the eyes.

Finally, this work researches how to utilize gaze information for context awareness. From experiences gained by implementing an extension for a web-activity recording tool (UsaProxy) to also record eye-gaze activity, it becomes clear that event handlers for gaze activity, similar to the ones for mouse activity, are desirable for next generation operating systems and Internet browsers. As the activity of the user is valuable context information, the thesis focuses on reading detection as one example of user activity. It presents new algorithms for reading detection. The user study confirms that the algorithms work well even for font sizes below the accuracy limit of the eye. One algorithm allows implementing a compulsory read by detecting whether the text was read completely. The other algorithm delivers numerical values, which makes it possible to decide whether a text was read carefully or only skimmed. An application for reading detection and recording of gaze activity with UsaProxy within e-learning environments was published in [Drewes, Atterer, Schmidt 2007].

2 Overview and Related Work

There is an immense knowledge on eye tracking and related fields such as the anatomy and physiology of the eye, the movement of the eyes and visual perception. The knowledge spreads over many disciplines of science including biology, medicine, psychology, and neurology. Even a rough overview on the topic could fill books. See Duchowski's textbook on eye-tracking methodology [Duchowski 2002] for a more detailed overview.

This overview starts with a definition of *eye tracking* followed by a short history of eye tracking and the fields of application for eye trackers. The next two sections explain the eye-tracking technology and present available eye-tracker systems. A further section explains the commercial eye tracker used for all experiments and user studies in this work. Finally, the last section mentions the scientific work on eye tracking for interaction done in the past, but only on a general level as the following chapters have their own related work sections.

2.1 Definition of Eye Tracking

The term *eye tracking* as it is used here means the estimation of direction of the user's gaze. In most cases the estimation of the gaze direction means the identification of the object upon which the gaze falls. In the case of a standard computer device, the coordinates on the screen identify the object of gaze. Interpretation of gaze direction is more complex for eye tracking in 3D virtual worlds and becomes difficult when interacting with the real world.

Eye trackers differ in the degrees of freedom which they can track. Simple eye trackers report only the direction of the gaze relatively to the head (EOG and systems rigidly mounted on the head) or for a fixed position of the eyeball (systems which require a head fixation). Systems that are more sophisticated allow free head movements in front of a stationary system. Such systems do some kind of (implicit) *head tracking*. In addition, wearable eye trackers for use in 3D virtual worlds have to report the direction of the gaze in space and not only relatively to the head. This work refers to such systems with the term *eye tracker* and does not explicitly name such systems as *eye-and-head-tracker*. (In the same way, this work refers to the eye and not to the eye and equilibrium organ although this would be more precise in some cases.)

Most video-based eye trackers deliver not only the direction of gaze but also the size of the pupil. The widening and narrowing of the pupil is an emotional response to the perceived scene and for this reason is quite interesting for research. However, as the main function of the pupil is the regulation of the amount of light entering the eye, such research requires stable light conditions. This work does not consider eye tracking in the sense of pupil size detection.

There are other forms of *eye tracking*, for example [Morimoto, Koons, Amir, Flickner 1999], [Ravyse, Sahli, Reinders, Cornelis 2000], [Ravyse, Reinders, Cornelis, Sahli 2000] and [Ying-li, Kanade, Cohn 2000], which track the eye as a whole and focus on the shape of the eye, whether it is closed and includes the eyebrows in the eye tracking. The terms *eye-gaze tracking* or *gaze tracking* instead of *eye tracking* would be more precise as these exclude explicitly such misinterpretation. However as most researchers of this field simply use the term *eye tracking* for *eye-gaze tracking* this thesis uses the term *eye tracking* as a synonym to *eye-gaze tracking*.

2.2 History of Eye Tracking

This overview on the history of eye tracking follows the essay “Methoden der Augenbewegungsmessung” of Erich Schneider and Thomas Eggert [Schneider@]. The authors refer to [Wade, Tatler, Heller 2003]. See also [Jacob, Karn 2003], which contains two pages on the history of eye tracking.

The first qualitative descriptions of eye movements date back to the 18th century (Porterfield 1737). At the end of that century Wells (Wells 1792) used afterimages, also called ghost images, which appear in the visual perception after staring some time on the same spot, to describe the movement of the eyes. In the 19th century Javal (1879) and Lamare (1892) observed the eye movements during reading and introduced the French originated word *saccade* for the abrupt movements of the eye. They used a mechanical coupling of the eyes and the ears using a rubber band to make the eye movements audible. Ahrens (1891), Delabarre (1898), and Huey (1898) were the first who tried to record the eye movements transferring the movements to a surface covered with soot by small levers fixed to the eyeball.

Dodge and Cline [Dodge, Cline 1901] made the first unobtrusive measurements in 1901. They used a photographic method and light reflections from the eye, recording eye movements in horizontal direction only. Some years later Judd, McAllister & Steel (1905) applied motion picture photography for eye movement recording. The invention of motion picture photography gave the possibility of frame-by-frame analysis of the eyes’ motion and enabled quantitative research on a solid basis.

Miles Tinker did studies of eye movements in reading in the 1930s and researched the effect of typeface, font size, and layout on the speed of reading.

In 1939, Jung measured vertical and horizontal eye movements simultaneously with electrodes applied at the skin close to the eyes. This method also called electrooculography (EOG) measures the electric fields of the eyeball which is a dipole. The method also gave the first (theoretical) possibility of real-time processing of gaze data by means of analogue electronics.

In 1947 Paul Fitts, who later became famous for his Fitts’ law, used motion picture cameras to record the eye movements of air force pilots when landing their airplane. His interest was how pilots use their cockpit controls. This was the earliest usability study using eye tracking [Jacob, Karn 2003], [Fitts, Jones, Milton 1950].

In 1948 Hartridge and Thompson invented the first head-mounted eye tracker for free head movements.

In the 1970s, there was a lot of improvement in eye-tracking technology. The eye trackers became less intrusive, provided better accuracy, and were able to dissociate eye from head movements by multiple reflections from the eye (Cornsweet and Crane, 1973). Psychology started to study perception and cognition.

In the 1980s, mini computers became powerful enough to do real-time eye tracking and this gave the possibility using video-based eye trackers for human computer interaction. Bolt presented such a vision in 1981 [Bolt 1981]. It was also the time of the first eye trackers to assist disabled users [Majaranta, Räihä 2002].

From the 1990s up to now, there has been a steady increase in the use of eye trackers. Falling prices for the tracking systems caused wider use typically for marketing research or usability studies. Scientists started to research the possibilities of eye trackers for human computer interaction.

2.3 Application Domains for Eye Tracking

Typical application domains for eye trackers are:

- Market research and advertising testing

The perhaps biggest field in terms of money is the use of eye trackers for market research. When designing posters for an advertisement campaign the marketing research department likes to test the materials. They present the posters to potential clients whose eyes are tracked to get answers to questions like “Did the person look at the product?”, “How much time did the gaze spend on the company’s logo?” and so on. With a portable eye tracker it is also possible to send people to a supermarket to find out which products the people notice and how much influence the form or colour or positioning of the product has on being noticed. See Figure 2 for examples of eye trackers in marketing research.

Figure 2: Advertisement for Tobii eye trackers showing a possible application for their product (scanned from a Tobii flyer)

- Usability research

Another field of commercial interest is usability testing. The first use of eye trackers, mentioned already in the section on the history of eye trackers, was done in 1947 for the American air force to find out the best positions for the controls in an aircraft cockpit. When offering a new device to somebody whose eyes are tracked, it is easy to see where the gaze moves in the expectation to find the control for solving the given task.

With the rise of the World Wide Web as a commercial platform the usability of web pages became an important topic. The user interfaces provided by web pages are very often the only contact of the company and the client and if only few percent of the users are irritated and do not click the “I order now” button the company loses sales. See [Pan, Hembrooke, Gay, Granka, Feusner, Newman 2004] for a scientific study with the title “The determinants of web page viewing behaviour: an eye-tracking study”.

- Eye control for accessibility

Another field for eye trackers is accessibility. Quadriplegics and people with diseases, causing a loss of control over the muscles, use eye trackers to interact with the world. Such eye tracker systems provide an eye-typing interface with a text-to-speech output. In addition, other types of eye-control, for instance directing the wheel chair or switching on the TV, are common in this field. See [Majaranta, Räihä 2002] for a scientific retrospective overview on eye typing. See [Hornof, Cavender, Hoselton 2004] for EyeDraw, a software program that enables children with severe mobility impairments to use an eye tracker to draw pictures with their eyes.

- Psychology and vision research

Eye trackers are a valuable tool for research on vision, perception, cognition, and psychology. One of the first things psychologists did after eye tracking became available was studying the reading process. See the publication list at [Deubel@] for examples of current psychology research on eye movements.

- Medical research, diagnostics and rehabilitation

Eye trackers are also a valuable tool for medical research and diagnostics to examine the function of the eyes especially on people with eye injuries or brain damage causing partial loss of vision. In these cases eye tracking is used to monitor the success of physical rehabilitation.

- Gaze interaction and car assistant systems

Future applications for eye-tracking technologies are gaze interaction for regular users. A TV set could switch on by itself when somebody is looking at it. The car industry does research in this field too, with the aim of developing assistant systems for cars. For example, an eye tracker in the car could warn the driver when she or he falls asleep while driving the car. This field of application for eye tracking is not yet commercially available and needs further research.

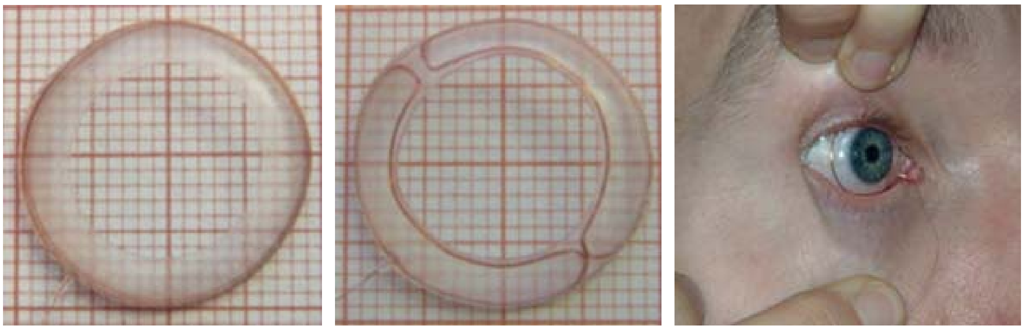
2.4 Technological Basics of Eye Tracking

2.4.1 Methods of Eye Tracking

There are three different methods to track the motion of the eyes.

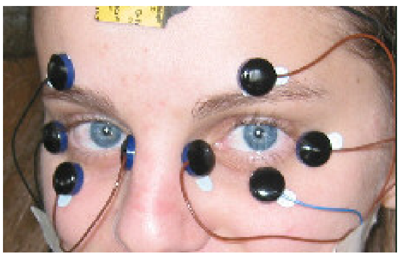
The most direct method is the fixation of a sensor to the eye. The fixation of small levers to the eyeball belongs to this category, but is not recommended because of high risk of injuries. A safer way of applying sensors to the eyes is using contact lenses. An integrated mirror in the contact lens allows measuring reflected light. Alternatively, an integrated coil in the contact lens allows detecting the coil's orientation in a magnetic field (see Figure 3). The thin wire connecting the coil with the measuring device is not comfortable for the subject. The big advantage of such a method is the high accuracy and the nearly unlimited resolution in time. For this reason, medical and psychological research uses this method.

Figure 3: Scleral search coils as offered by Chronos Vision (taken from [Chronos Vision@])



Another method is electrooculography (EOG) where sensors attached at the skin around the eyes measure an electric field (see Figure 4). Originally, it was believed that the sensors measure the electric potential of the eye muscles. It turned out that it is the electric field of the eye which is an electric dipole. The method is sensitive to electro-magnetic interferences but works well as the technology is advanced and exists already for a long time. The big advantage of the method is its ability to detect of eye movements even when the eye is closed, e.g. while sleeping.

Figure 4: Sensors attached at the skin around the eyes (taken from [metrovision@])



Both methods explained so far are obtrusive and are not suited well for interaction by gaze. The third and preferred method for eye-gaze interaction is video. The central part of this method is a video camera connected to a computer for real-time image processing. The image processing takes the pictures delivered from the camera and detects the eye and the pupil to calculate the gaze's direction. The following sections present several solutions to achieve this. The big advantage of video-based eye tracking is the unobtrusiveness. Consequently, it is the method of choice for building eye-gaze interfaces for human-computer interaction.

2.4.2 Video-Based Eye Tracking

The task of a video-based eye tracker is to estimate the direction of gaze from the picture delivered by a video camera. A possible way is to detect the iris using the high contrast of the white of the eye and the dark iris. This method results in good horizontal but bad vertical accuracy as the upper and lower part of the iris is covered by the eye-lid. For this reason, most video-based eye trackers work with the detection of the pupil.

There are two methods of illumination to detect the pupil – the dark and the bright pupil method. With the dark pupil method the image processing locates the position of a black pupil in the camera image. This can be problematic for dark brown eyes where the contrast between the brown iris and the black pupil is very low. The bright pupil method uses infrared light reflected from the retina and this makes the pupil to appear white in the camera image. The effect is well known as "red eye" when photographing faces with a flash. The bright pupil eye trackers require an illumination of the eye with infrared light coming from the same direction as the view of the camera. Therefore, an infrared LED has to be mounted inside or close to the camera, which requires mechanical effort. For differences in the infrared bright pupil response among individuals, see [Nguyen, Wagner, Koons, Flickner 2002].

All video-based eye-tracking methods need the detection of the pupil in the camera image. This is a task for image recognition, typically edge detection, to estimate the elliptical contour of the pupil (see the chapter 'Pupillendetektion' in [Schneider2@]). Another algorithm for pupil-detection is the starburst algorithm explained in [Dongheng, Winfield, Parkhurst 2005] and [OpenEyes Starburst@]. A further method estimates the parameters of the elliptical shape of the pupil using Hough transformation [Thomae, Plagwitz, Husar, Henning 2002]. This method is patented [DE102005047160].

Many eye trackers use a reflection on the cornea to estimate the direction of gaze. As the cornea has a nearly perfect sphere shape, a glint stays in the same position for any direction of gaze while the pupil moves. As the radius of the cornea differs from person to person this simple eye-tracking method needs a calibration for each person. See 2.4.3 for a detailed explanation of the method. More sophisticated eye trackers use stereoscopic views with two cameras or multiple light sources for multiple glints to avoid calibration and to allow free movement in front of the device. Such properties are desirable for HCI. For a description how such system works see [Ohno, Mukawa, Yoshikawa 2002], [Ohno, Mukawa 2004] or [Hennessey, Noureddin, Lawrence 2006].

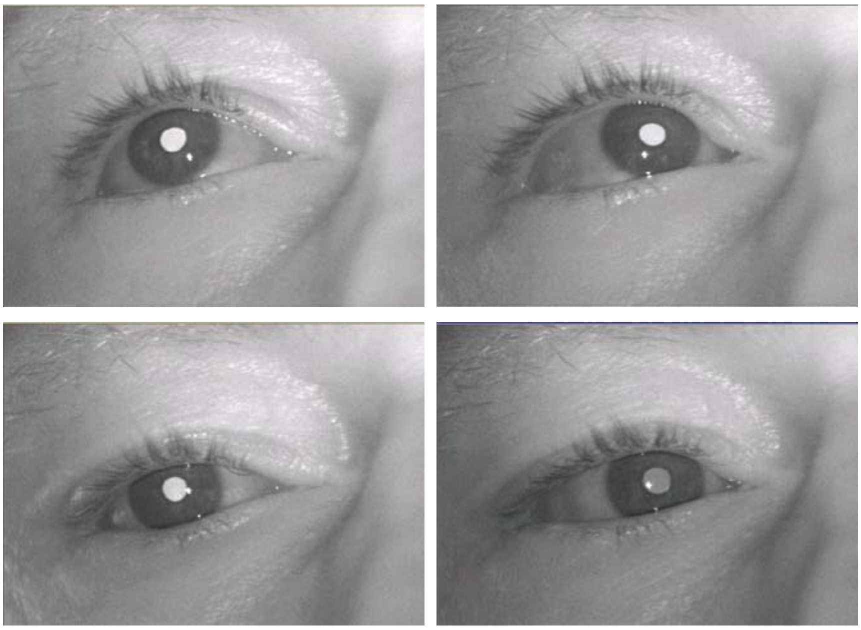
The corneal reflection can be used to build a simple attention sensor which Vertegaal calls ECS or Eye Contact Sensor [Dickie, Vertegaal, Sohn, Cheng 2005]. When looking directly at the combination of an eye tracker camera and an infrared LED, the glint appears in the centre of the pupil. The attention sensor only detects whether the glint is inside the pupil to know whether an eye is looking at it. There is no calibration needed for such an attention sensor. A commercial ECS called eyebox2 is available at [xuuk@] (see Figure 11).

There are eye trackers which also track the rotational movement of the eye but such systems are not very common. For research on this topic, see [Schneider 2004]. A nice application for such eye trackers is a head camera controlled by the eye movements using the motion stabilization of the eye to get a motion stabilized camera picture [Böning, Bartl, Dera, Bardins, Schneider, Brandt 2006], [Wagner, Bartl, Günthner, Schneider, Brandt, Ulbrich 2006], [Schneider2@], [DE 102 51 933 A1].

2.4.3 The Corneal Reflection Method

The ERICA eye tracker [ERICA@], which is the eye tracker used for all user studies presented here, shall serve as an example for a standard eye-tracking technique called the corneal reflection method. The eye tracker has an infrared camera with an infrared LED mounted inside the camera to illuminate the eye. The background of the eye, the retina, reflects the infrared light. Because the illumination has the same direction as the optical axis of the camera the camera sees the reflected light and the pupil appears white (see Figure 5). The same effect causes the red eyes on a photography taken with a flash. A white pupil provides a high contrast independent of the colour of the eyes (iris) and is easy to detect by the image processing software.

Figure 5: Looking to the four corners of the screen – the reflection stays in the same position



The infrared light also causes a reflection on the cornea (also called first Purkinje image). As the cornea has a perfect sphere shape, the glint stays in the same position independent of the gaze direction. The method does not work for people with deformations of the eyeball. It can also be problematic when people wear contact lenses. Glasses are less problematic because although the glasses may change the position of the glint the reflection stays in the same position. The calibration procedure described below compensates the optical distortion of the glasses. In rare cases, a reflection from the glasses can fool the image processing software, which misinterprets the reflection from the glasses with the reflection from the cornea.

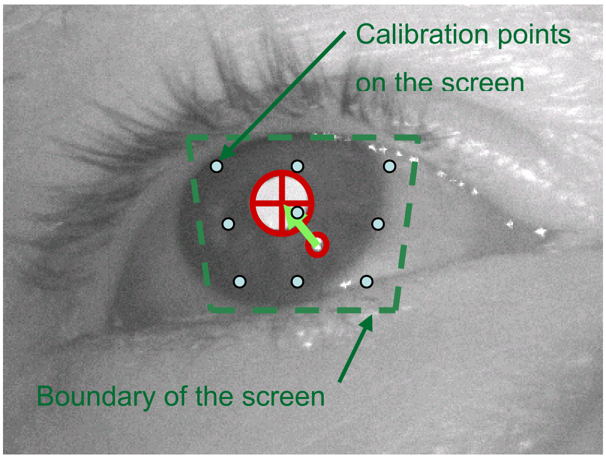
The image processing software detects the position of the glint and the centre of the pupil. The vector from the glint to the centre of the pupil is the basis for the calculation of the gaze's direction and finally the position of the gaze on the screen.

A direct calculation would not only need the spatial geometry of the eye tracker, the infrared LED, the display and the eye but also the radius of the eyeball, which is specific to the subject using the eye tracker. For this reason, a calibration procedure estimates the parameters for the mapping of the glint-pupil vector to positions on the screen. See also 3.2 for a discussion on the need for calibration.

An easy approach for the calibration procedure is to look at the four corners of the screen (see Figure 5). The four lines of sight form a pyramid and the intersection with the plane of the screen results in a trapeze on the camera image for the four pupil centres looking to the display corners (see Figure 6).

Although these four points are enough to do a calibration, many eye trackers use more points for the calibration to achieve better results as more points damp the inaccuracy which lies in the measurement for a single point.

Figure 6: The vector from the glint to the centre of the pupil and the positions of nine calibration points



2.5 Available Video-Based Eye Tracker Systems

2.5.1 Types of Video-Based Eye Trackers

The most common mechanical setup is a stationary eye tracker. Such systems are commercially available as a laboratory tool and are typically used in medical research or marketing research. These systems comprise of as a desktop computer with integrated eye tracker and with a software package for analyzing and visualizing eye-gaze data. While medical eye trackers often have a chin rest for head fixation, marketing research prefers systems with additional head tracking for free head motion in front of the device. Such systems often have two cameras for stereoscopic view but details on how the system works exactly are a business secret of the producing company.

Accessibility systems typically use a tablet computer with an external camera which both can be mounted on a wheel chair. Such systems normally come along with an eye-typing application and a speech synthesizer.

An alternative mechanical setup is a head-mounted eye tracker. With a laptop for the image processing in a backpack head-mounted eye trackers are wearable and allow free mobility of the user. Beside a camera which observes the eye such systems have an additional head-mounted camera which records the field of view.

The number of research eye-tracker systems is immense and they are normally not available for purchase. Such systems are typically purpose-built with properties a commercial eye tracker does not have or even to research new eye-tracking technologies.

2.5.2 Low-Cost Open Source Eye Trackers for HCI

For research in the field of HCI it is sometimes advantageous to build an eye tracker yourself. This allows modifications for research purposes, which are difficult with commercial systems as the company who builds the eye tracker normally tries to protect their intellectual property, and does not provide the necessary information. A self-built eye tracker is also much cheaper.

Figure 7: The Low-Cost Mobile Eye Tracker from the openEyes project

There are several instructions on how to build an eye tracker on the internet. The most popular project seems to be the openEyes project from Iowa State University (see Figure 7). See the paper "openEyes: a low-cost head-mounted eye-tracking solution" [Li, Babcock, Parkhurst 2006] for a scientific report, the technical report "Towards an open-hardware open-software toolkit for robust low-cost eye tracking in HCI applications" [OpenEyes Tech. Report@] or "Constructing a Low-Cost Mobile Eye Tracker" [OpenEyes Construct@] for instructions and [OpenEyes@] for downloading the software.

There are other projects also worth considering for example "Eye tracking off the shelf" [Hansen, MacKay, Hansen, Nielsen 2004] from IT University Copenhagen or the "opengazer" project from the Machine Intelligence Laboratory in Cambridge University Engineering Department [opengazer@], [opengazer sourceforge@].

See [COGAIN Low cost@] for a list of low-cost eye-tracker projects.

2.5.3 Commercial Eye Trackers for HCI

The number of available systems is continuously increasing. The selection presented here is not a recommendation and chosen only for the purpose of illustration. There is a list of available systems on [COGAIN Eyetrackers@] listing 28 different companies.

A typical eye tracker for the field of accessibility comes from ERICA [ERICA@]. The eye tracker consists of a tablet PC and an external camera and it can easily be mounted on a wheel chair. See Figure 12 in chapter 2.6 for a picture of the ERICA eye tracker.

Probably the most popular eye tracker manufacturer is Tobii [Tobii@]. The Tobii eye trackers (see Figure 8) allow free head movements in front of the device and can be found at psychology departments and marketing research companies.

Figure 8: The Tobii X120 (stand-alone) and the Tobii T60 (integrated into a display) (taken from [Tobii@])

The EyeFollower [interactive-minds@] eye trackers are an alternative option for the same type of application (see Figure 9).

Figure 9: The EyeFollower (taken from [interactive-minds@])

Head-mounted displays with eye trackers are available from Arrington [Arrington@] and SMI (Senso Motoric Instruments) [SMI@] (see Figure 10).

Figure 10: Head-mounted eye tracker from SMI (taken from [SMI@])

Figure 11 depicts a special type of eye tracker, which is especially interesting for the field of HCI and is called ECS (Eye Contact Sensor). It does not deliver accurate coordinates for the direction of the gaze but only signals eye contact.

Figure 11: The Eye Contact Sensor Eyebox2 from xuuk inc. (taken from [xuuk@])

2.5.4 Criteria for the Quality of an Eye Tracker

There are many different criteria for the quality of an eye tracker. The question how to weigh these criteria depends on the purpose the eye tracker should be used for. Beside the main criteria as the technology of the eye tracker (black or white pupil detection, etc.) and the mechanical setup (stationary or mobile) there are the following further aspects:

- Accuracy
  As explained later (see 3.2) there is an accuracy limit for the eye of about ±0.5° caused by the anatomy of the eye. Most eye trackers today state exactly this value in their data sheets. Such accuracy is sufficient for human-computer interaction. Research on micro saccades and tremor requires better accuracy.

- Time resolution and latency
  Every eye tracker reports the gaze direction in certain time intervals. For video-based eye trackers the time intervals are determined by the frame rate of the video camera. However, the frame rate does not necessarily correspond to the time resolution. For example, the ERICA eye tracker used for the user studies presented later reports coordinates with a frame rate of 60 Hz but does not report any data during a saccade.
  In many cases the latency, that means the delay in reporting the data, is more important than the time resolution especially in the field of HCI where instant response of a system is crucial.

- Robustness
  An eye tracker should work reliably in any situation and with all persons. A situation where an eye tracker does not work any more could be difficult light conditions. The same can happen if a person wears contact lenses or glasses.

- Low-level filtering
  The raw data from an eye tracker are noisy and have gaps while the eye is closed when blinking. For this reason the eye tracker filters the noise and smoothes the signal. In consequence, the eye tracker does not report small movements of the eye. It depends on the eye-tracker system to which extent it is possible to set parameters to control the smoothening of the signal.

- Application programming interface and data visualization
  Most eye trackers come along with a software package for visualization and evaluation of recorded eye movements. An eye tracker also provides some kind of application programming interface, which makes it possible to write applications for the eye tracker.

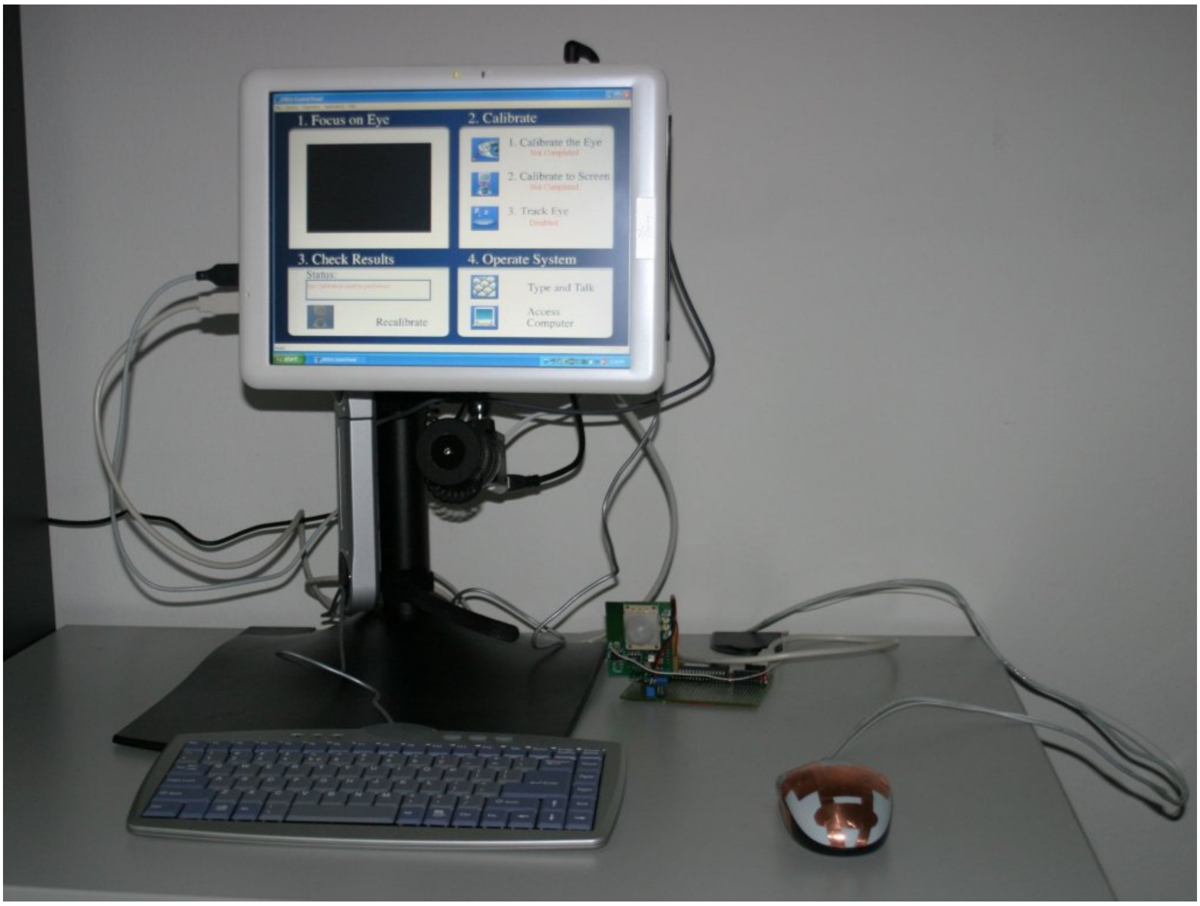
2.6 The ERICA Eye Tracker

The ERICA eye tracker is a low-budget commercial system of Eye Response Technologies [ERICA@] and was used for all presented studies. The ERICA system shall serve as an example to present an eye tracker in detail.

2.6.1 Specifications

The ERICA eye tracker is a system without head tracking and requires the fixation of the user's head with a chin rest. Although it provides some software for eye-gaze data analysis, it is a typical system for accessibility. The need of a chin rest does not matter for a disabled person who cannot move but changes the experience for a person without disabilities. It comes as a tablet PC with an external infrared camera, which both could be mounted at a wheel chair. For all studies the eye tracker was fixed on a stand and positioned on a table (see Figure 12).

Figure 12: The ERICA eye tracker



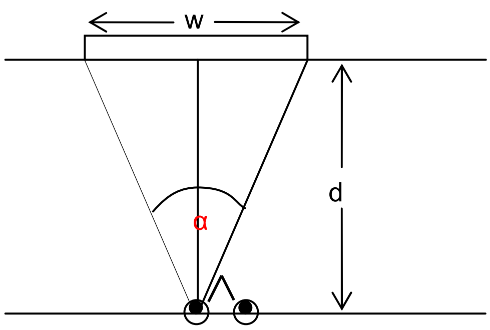
The tablet PC has a standard PC architecture with a 1.4 GHz processor and 512 Megabyte RAM. The operating system installed is Windows XP SP2. As it uses a time slice of 10 milliseconds, 10 milliseconds are the best accuracy that can be achieved for any system event e.g. mouse moves. The TFT display can switch pixels with about 20 ms.

The ERICA system uses the white pupil method and a corneal reflection (already explained in chapter 2.4.3). The camera delivers the image of one eye with a resolution of 640 x 480 pixels at a frame rate of 60 Hz or about one frame every 17 milliseconds. The ERICA system reports the gaze position with the same rate in the case the eye does not move (during fixations). During eye movements the eye tracker does not deliver any data. This means there is a time gap in the data during a saccade.

2.6.2 Geometry of the Experimental Setup

Figure 13 shows the geometry of the eye-tracker setup from a bird's-eye view. The tablet PC's display has a size of 246 mm x 185 mm (12') and a resolution of 1024 x 768 pixels. The distance *d* of the eyes from the screen, given by the position of the chin rest, is 48 cm ± 2 cm. From tan(α/2) = *w*/(2**d*) the angle α calculates to α = 28.7°. This means in average 0.028° per pixel or around 36 pixels for 1°. In the centre of the screen, which is flat and not sphere shaped, the pixels per degree are about 35 pixels. As the dependency of the pixels per degree from the viewing angle is small, all calculations for the evaluation of the user studies use the average value of 36 pixels for 1°.

Figure 13: Geometry of the experimental setup (top view), d = 48 cm, w = 24.6 cm, α = 28.7°



The following relations apply to all distances given in pixels within this thesis.

1° = 36 pixels

1 pixel = 0.028°

2.6.3 The ERICA-API

A simple SDK (software development kit) to access the ERICA system on Windows platforms by programs is available. It provides four functions defined in the header file:

```
// Used to tell ERICA where to send its messages
void PASCAL ericaSetReceiver( HWND hwnd, BOOL bWantSmoothedData );

// Used to get the name of the ERICA message
void PASCAL ericaGetMessageName( char *pchName );

// Used to get the low and high words of a lVal
void PASCAL GetSignedLowHigh( long lVal, int *pwLow, int *pwHigh );

// When processing an ERICA message, this is used to determine
//   if a blink occurred just prior to the received data
BOOL PASCAL ericaDidBlinkOccur();
```

The ERICA-API uses the Windows message concept to deliver the data from the eye tracker to a window which has to process the message. The use of the API within Windows programming projects is easy and straightforward. After requesting a message name (`ericaGetMessageName)`, registering the message at the Windows operating system (`RegisterWindowMessage`) and setting the receiver window (`ericaSetReceiver`), the ERICA system sends messages with the x-y position of the eye gaze and the x-y extend of the pupil in the message parameters (`WPARAM, LPARAM`).

The simplicity of the API is sufficient for many possible applications and makes it easy to start programming. There are some problems and bugs in the ERICA-API. The message stamp delivered with the ERICA messages is corrupt and this makes it necessary to use the system time (`GetTickCount`) at the time of message receipt instead of the time of message generation. This can influence the accuracy of timings but only if the system is very busy. Another problem is the delivery of coordinates relative to the client area of the window. It is an easy task to transform the coordinates into screen coordinates with a Windows API function (`ClientToScreen`). Moving the receiving window, however, causes a new gaze position relatively to the window and produces an ERICA message which appears as an eye movement. If the receiving window is minimized, wrong coordinates can be reported. It also seems that occasional signed-unsigned mismatch errors occur when delivering coordinate values. Once aware of the problem one line of code can fix this.

From the view of a programmer, it is a little bit cumbersome that only one application at a time gets the ERICA messages. This makes it impossible to run two applications which use the ERICA-API in parallel. A lost-signal-notification in the case the image processing cannot recognize the eye, for example because it moved out of the focus of the camera, is also desirable. From the view of a scientist, it is a pity that there is no access to the calibration data and only one Boolean parameter (`bWantSmoothedData`) for controlling the filter. Without access to the calibration data it is not possible to program alternative calibration procedures or to adjust the calibration while using the eye tracker. Without control of the filtering mechanism the existence of a minimum saccade length and strange effects as the gap in the data for saccade lengths (see 3.3.2) have to be accepted. The fact that the ERICA system does not report data during a saccade, which is a property of the filter mechanism, inhibits the prediction of the saccade end point and time when the saccade movement is on half of the way.

The blinking detection (`ericaDidBlinkOccur`) worked fine, but requires a longer closing of the eye than for the natural blink. Therefore, the use of blinking does not bring a speed benefit compared to the dwell method. According to informal talks with test users blinking to trigger commands does not feel comfortable. Especially repetitive blinking over a longer time, as it is typical for computer interaction, seems to create a feeling of nervous eye muscles. Therefore, the conducted user studies did not use triggering of actions by blinking.

2.7 Related Work on Interaction by Gaze

As mentioned already in the historical overview on eye tracking the processing power of computers became sufficient for real-time video processing in the early 1980s. This was the precondition for the use of eye tracking for human computer interaction and the early papers on such research date back to this time. Since then the annual number of publications on the topic increased steadily. Since 2000 there is a biannual international conference on eye tracking called ETRA (Eye Tracking Research & Application) [ETRA@]. Since 2005, the COGAIN (Communication by Gaze Interaction) initiative, supported by the European Commission, also organizes conferences and offers a bibliography of research papers in the internet [COGAIN Bibliography@].

In 1981, MIT researcher Bolt envisioned the very first application of eye gaze as interaction method for computers. His paper "Gaze-orchestrated dynamic windows" [Bolt 1981] is a vision of multimodal computer interaction which is not only ahead of his time but also of today. He describes a Media Room with a big display where 15 to 50 windows display dynamic content simultaneously which he named "World of Windows". His idea is to zoom-in upon some window the user is looking at and he discusses two different methods for the interaction. His first method is a "timing-out of how long you look at a window" and nowadays called dwell-time method. The second method is addressing the window with the gaze and using an additional modality, he mentioned joystick or speech command, to zoom-in. Bolt discusses windows with real-time content and non-real-time content separately. For non-real-time content, such as movies or slide shows, he suggests that it proceeds only when the user is looking at it. (See also the paper of Vertegaal et al. [Dickie, Vertegaal, Sohn, Cheng 2005].) One year later Bolt published the paper "Eyes at the Interface" [Bolt 1982] summing up the importance of gaze for communication and concludes the need of gaze-awareness for the interface.

In 1987, Ware and Mikaelian did a systematic research on eye trackers as an input device. Their paper "An evaluation of an eye tracker as a device for computer input" [Ware, Mikaelian 1987] reports about experiments to measure times the eyes need for selection. Ware and Mikaelian tested three different input methods which they called "dwell time button", "screen button", and "hardware button". The dwell-time button method is the standard method used today for eye typing. The gaze has to stay for a certain time, the dwell time, on the button to trigger the action associated with the button. The screen button method is a two-target task. The gaze moves to the chosen button and afterwards to the screen key to trigger the action. The hardware button method uses a key to press with the finger in the moment when the gaze is on the chosen button. The first two methods are gaze-only while the hardware button uses an extra modality. Unluckily, Ware and Mikaelian fitted their data against a 'modified' Fitts' law to compare their results with the experiments done by Card et al. [Card, Moran, Newell 1983] for mouse devices. Up to now there is a controversial discussion in the HCI community on the question whether Fitts' law holds for the eye (3.4.3 gives a detailed discussion).

Another important paper written by Jacob for the Naval Research Laboratory with the title "What You Look at Is What You Get: Eye Movement-Based Interaction Techniques" appeared in 1990 [Jacob 1990] and as a journal article in 1991 [Jacob 1991]. One big contribution of this paper is the identification of the Midas Touch problem which Jacob explained with these words:

“*At first, it is empowering simply to look at what you want and have it happen. Before long, though, it becomes like the Midas Touch. Everywhere you look, another command is activated; you cannot look anywhere without issuing a command*” [Jacob 1990].

Jacob’s work systematically researched the interactions needed to operate a GUI (graphical user interface) with the eyes – object selection, moving objects, scrolling text, invoking menu commands and setting the window which gets the keyboard input. The generality of this paper caused all further research to focus on single or more specialized problems of eye-gaze interaction. In 1995, Jacob wrote an overview on eye tracking in advanced interface design [Jacob 1995] where he explains another critical aspect of gaze interaction: providing a feedback cursor. Of course the user knows where she or he is looking but not with a one-pixel-accuracy and additionally there may be calibration errors which cause the gaze position to differ from the position reported by the eye tracker. Providing a gaze feedback cursor can result in chasing the cursor across the screen or as Jacobs expressed it:

“*If there is any systematic calibration error, the cursor will be slightly offset from where the user is actually looking, causing the user's eye to be drawn to the cursor, which will further displace the cursor, creating a positive feedback loop.*” [Jacob 1995].

In 2000 Jacob wrote a paper with the title “Evaluation of Eye Gaze Interaction” in which he utters to be sceptical about Fitts’ law for the eyes: “*… our overall finding is that our eye gaze results are more similar to those of Abrams, et al. (The mouse results are similar to other Fitts' studies.)*” [Sibert, Jacob 2000], see also [Sibert, Templeman, Jacob 2000]. In the same year, he also published “Interacting with Eye Movements in Virtual Environments” [Tanriverdi, Jacob 2000] where he describes eye gaze interaction as a faster interaction method compared to conventional 3D pointing but at the price of weaker recall of spatial information. Finally in 2003 Jacob wrote a 26 pages overview titled "Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises (Section Commentary)" [Jacob, Karn 2003]. For more contributions of Jacob, see his list of publications in the internet [Jacob@].

At the time of Jacob’s first publication, eye trackers were expensive and far away from being a standard interface for everybody. Nevertheless, eye-tracking interfaces established as a standard interface in the field of accessibility. The typical application in this field is eye typing, which displays a standard keyboard layout on the screen and enables the input of characters by looking at a key for a certain time (dwell-time). Majaranta and Räihä give an overview on this topic in their article “Twenty Years of Eye Typing: Systems and Design Issues” [Majaranta, Räihä 2002]. They researched also auditory and visual feedback for eye typing [Majaranta, MacKenzie, Aula, Räihä 2003] and the effects of short dwell times [Majaranta, Aula, Räihä 2004]. Text entry by gaze is a very important interface technique in the field of accessibility and there is research on alternative methods than dwell time on standard keyboards. The Symbol Creator [Miniotas, Spakov, Evreinov 2003], for example, demands less screen space than the display of a standard keyboard. There are also attempts to assist text entry by keyboard with gaze input for the special case of Chinese characters [Wang, Zhai, Su 2001].

See also the related work sections 4.2 for eye gaze as pointing device. See the related work section 5.1 for gaze gestures and 6.2 for eye gaze as context information including eye gaze as attention indicator, activity recognition such as reading and other forms of implicit interaction.

2.8 Current Challenges

The current challenges for eye tracking can be classified in technical aspects and human aspects.

- Technical aspects

Although eye-tracker technology exists for many decades now there are still many possibilities to improve it. Eye trackers use different methods for the pupil detection. The white and the black pupil methods (see 2.4.1) have different mechanical requirements for the illuminating infrared LED and the illumination seems to be one of the technical challenges. While indoor use of eye trackers works quite well, outdoor use can be problematic because of extreme light conditions. The sun is a bright infrared radiator and can fool the eye tracker, which needs the infrared reflection from an LED. A possible approach to solve the problem is synchronizing the LED with the frame rate of the camera and using differential pictures [Morimoto, Koons, Amir, Flickner 2000], [Zhu, Fujimura, Ji 2002]. The opposite illumination problem happens during night where an extra illumination is needed. The scenario of a driver steering her or his car through the night and getting occasionally floodlight from oncoming cars illustrates how problematic the illumination issue can be. Nevertheless, the night situation is less difficult to solve, as the infrared intensity of floodlight is much lower than the infrared intensity of bright sunlight. Rapid changing light conditions at daytime caused by travelling with high speed in a car or a train are most probably the biggest challenge. The shadows in the face from street lamp masts or telegraph masts cause a stroboscopic illumination, which requires eye trackers with very fast adoption to changing light conditions. There are scientists who dream to solve illumination issues with eye trackers working with the infrared light emitted by the body because of the 37°C body temperature. However, this requires cameras working with very low infrared light.

The technical progress in sensors for cameras is still ongoing and influences eye-tracking technology. Special cameras like ROI (region of interest) cameras, which can deliver high-resolution pictures at low frame rates and partial pictures at high frame rates, give potential for technical improvement of eye trackers. Eye trackers which detect a face in a wide range angle and over big distances, zoom into the face and detect the direction of the gaze are not commercially available yet and a topic for further research.

- Human aspects

One of the big challenges is how to deal with the low accuracy of eye tracking and the need of a calibration procedure. As explained later, the low accuracy and the need of calibration is not a deficit of the eye-tracking technology but lies in the anatomy of the human eye. Consequently, the challenge is not how to solve it but how to deal with it. See section 4.2 for the suggestions done so far on the accuracy issue and section 3.2, where the so-called calibration-free eye-tracking methods are mentioned. The chapter 4 on eye gaze as pointing device in this thesis examines a possible solution of the accuracy problem and chapter 5 introduces gaze gestures as a method for computer input which does not require calibration.

Another challenge, already mentioned in the related work section 2.7 as the Midas Touch problem, is how to solve input-output conflicts when using the gaze as the only modality for interaction. If the eye looks at something the question arises whether the eye does so for seeing what is there (input for the eye) or to trigger an action (output from the eye). One possibility to solve this problem is the use of an extra input modality, for example a key to press, but this makes the eye gaze interaction lose some of its benefits. With an extra key, eye gaze interaction is not contact free anymore, it is not suitable for remote control, and it is not applicable for the disabled. The use of eye gaze in combination with an extra input modality is not fully researched up to now. It is possible to combine gaze input with every other input modality, a property that sometimes is called orthogonal. This gives a big number of combinations especially when varying the methods for the input modalities.

The perhaps biggest challenge is the use of eye tracking for context awareness. Context awareness in general is a current challenge in the field of HCI. When thinking of all the proverbs, idioms, and expressions, using the terms look, eye or gaze – to get some very odd looks, to take a good look at, to look about or around or for, to gaze about, to have a keen eye at, with an uneasy eye, to eye up – it becomes clear that detecting eye gaze is a very promising method for recognizing context. Chapter 6 focuses on reading detection as an example for context-awareness.

All in all the challenge is to find new gaze-based interaction methods for implicit and explicit control of computer devices.

3 The Eye and its Movements

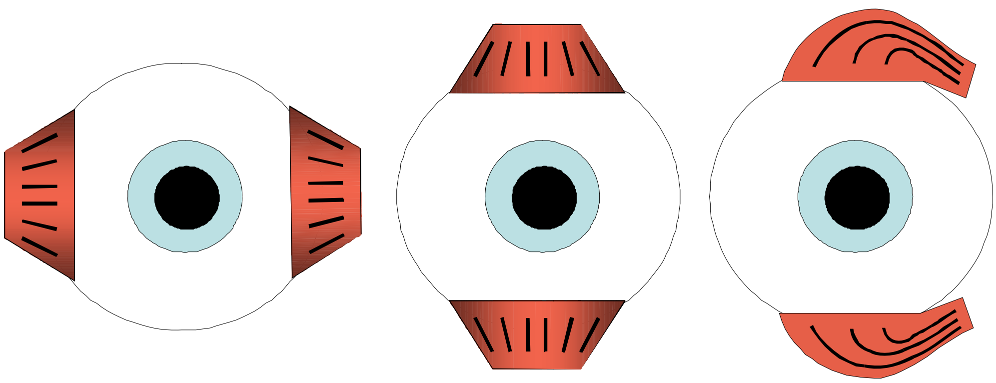
To research eye gaze based interfaces it is necessary to understand both sides of the interaction – the human eye and the eye tracker. The previous chapter introduced the eye trackers and this chapter presents general knowledge on the eyes as a preparation for interaction methods researched in the following chapters.

The first section gives an overview on the anatomy of the eye. The following section clarifies the accuracy and calibration issue. The subsequent section presents data from a user study where eye movements have been recorded while watching a video and while surfing the Internet. Statistics on the eye movement during these tasks for several individuals provide typical values and ranges for fixation times and saccade lengths and are the foundation for discussions in later chapters. The data also reveal typical limitations of eye trackers. The last section of the chapter discusses the speed and the accuracy of eye gaze and shows that there is no reason to apply Fitts' law to eye movements.

3.1 Anatomy and Movements of the Eye

In a simplified view and in the terms of engineering the eye and its muscles can be seen as a camera with image stabilization. Six muscles as depicted in Figure 14 connect the eye with the head. The muscles are organized as antagonistic pairs and give the eye 3 DOF (degree of freedom). One pair is responsible for horizontal movement, one pair controls the vertical movement, and the third pair allows rotational movement around the direction of view. Together the three pairs of muscles allow compensating all movements of the head. To achieve this task the nerves controlling the eye muscles are closely connected with the equilibrium organ located in the ear.

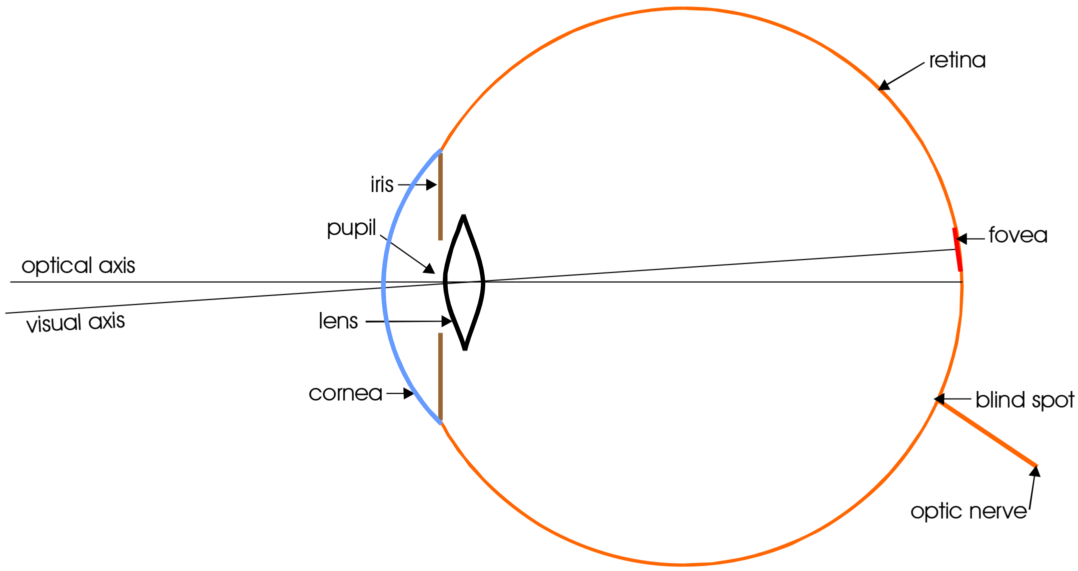
Figure 14: Three pairs of muscles can compensate all movements of the head



The construction of the eye itself is similar to that of a camera. There is a diaphragm, called iris for the eye, which allows setting the aperture. The hole in the iris is called pupil. In contrast to a camera that uses a lens with a fixed focal length and achieves focusing by changing its position, the lens of the eye can change the focal length by changing the lens form. The light sensitive area is the rear interior surface of the eye and called retina.

Figure 15 shows simplified schematics of the eye; for a more detailed picture see a standard textbook of anatomy, for example [Sobotta 2004, p. 366].

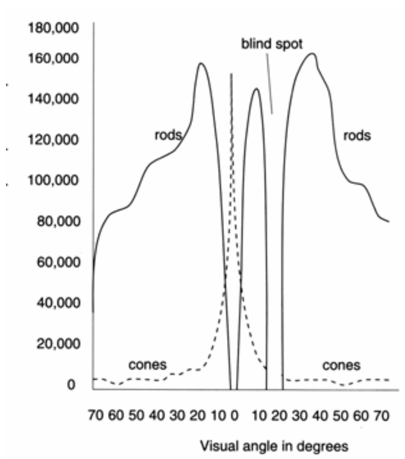
Figure 15: Schematics of the eye



The entire light-sensitive area is an ellipsoid with a horizontal extends of 180° and a vertical extends of 130°. There are two different types of light sensors on the retina – rods and cones. The rods are sensitive to brightness and have a high sensitivity to enable night vision. The cones are less sensitive and detect chromatic light. The rods and cones are not equally distributed over the retina – most parts of the retina have a very low density of light receptors and only a small circular area with a diameter of 1° to 2° called fovea has a high density of cones. The fovea contains around 150000 cones/mm^2 and only a small number of rods. Outside the fovea, the number of cones drops to 20000 cones/mm^2. Most of the visual field outside the fovea has a low resolution and its main task is the perception of ambient motion [Duchowski 2002, p. 33]. Figure 16 shows the density distribution of rod and cone receptors. Figure 17 visualizes the effect of the density distribution of the receptors for the image transmitted by the retina.

Figure 16 (left): Density distribution of rod and cone receptors (Taken from [Duchowski 2002, p. 34])

Figure 17 (right): Loss of resolution with eccentricity for the image transmitted by the retina. (Picture by Ben Vincent. From: Basic Vision, an Introduction to Visual Perception. Oxford University Press 2006. Taken from [Land 2006])



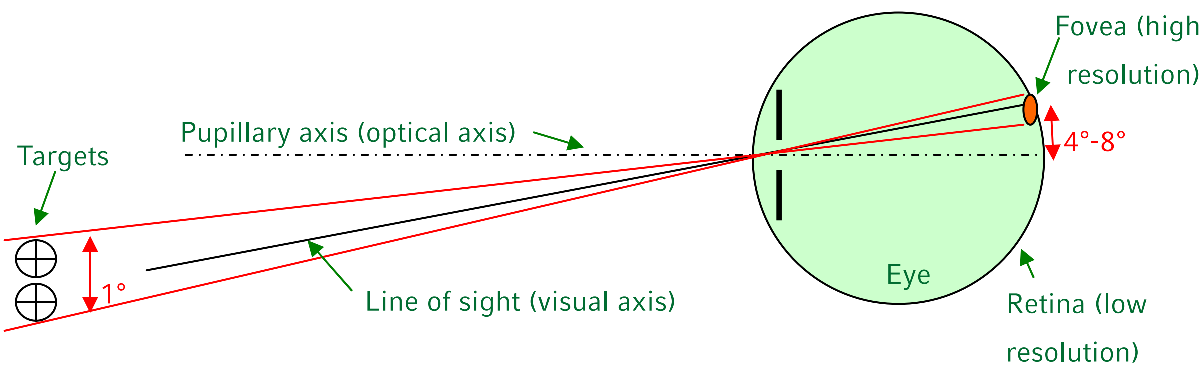
The eyes can move – and mostly do – in an unusual way compared to other parts of the body. First of all the eyes move synchronously. The eye needs a stable projection on the retina for image recognition. To achieve this three pairs of muscles compensate the movements of the head. The resulting movement of the eye is the inverse of the (smooth) head movement. The same kind of movement occurs when the eye follows a moving object. Such kind of movement is similar to other movements of the body. However, when there is no moving object and the head stays in stable position the eye gaze direction jumps in abrupt movements. These jumps are called saccades. After a saccade the eyes rest for a while called a fixation. The rest is the time of stable projection on the retina. The explanation for the movement in saccades lies in the size of the fovea. The fovea with its capability to deliver a high-resolution image is only around 1° visual angle in size. To see something clearly the eye has to turn to get the object of interest projected on the fovea and a quick movement, the saccade, does this. The time of rest, the fixation, is the time for the image processing which means the time the brain needs to recognize something. Depending on what was recognized the eyes perform a saccade to another spot of interest.

Looking at eye movements in smaller time and length scales shows drift, tremor, and micro saccades. Research on this kind of eye movement (see [Engbert, Mergenthaler 2006] as a pointer into literature) requires eye trackers with high angular and high time resolution. The eye tracker used for the user study of this thesis filters such movements and delivers a smoothed signal.

3.2 Accuracy, Calibration and Anatomy

Most commercial eye trackers state an accuracy of ±0.5° in their technical data sheet and need a calibration process before use. Both, the low accuracy and the need of a calibration, are obstacles on the way to a gaze-aware computer interface. For this reason the questions “Is a better accuracy possible?” and “Can we build an eye tracker which does not need a calibration?” are of big importance. Unluckily, the answer to both questions is “No!”

Figure 18: Size and position of the fovea



The reason lies in the anatomy of the eye, namely in the size and position of the fovea as illustrated in Figure 18. As mentioned in 3.1 the density of the light sensors (cones and rods) on the retina is not uniform. The fovea is a small spot of around 1° angular size and with a high density of light receptors. The rest of the retina has a low density. Our field of high-resolution vision, the 1° visual angle, is about the size of a thumbnail in arm length distance. Whenever we want to see something clearly, we move our eye to get the projection of the thing we want to see on the fovea. If the thing we want to see is smaller than the 1°, it is enough to have the projection somewhere on the fovea, not necessarily in its centre. There is no need for the eye to position its view more accurately. Additionally drift, tremor, and micro saccades of the eye movements, which are in a magnitude of 0.1°, contribute to the inaccuracy of eye pointing. All this behaviour of the eye, mostly the size of the fovea, limits its accuracy to around ±0.5°. Consequently, the accuracy issue is not a question of the eye tracker's accuracy, which could be increased by using higher resolution cameras. Ware and Mikaelian expressed it that way:

"*The research literature on normal eye movements tells us that it is possible to fixate as accurately as 10 minutes of visual angle (0.16 deg) but that uncontrolled spontaneous movements cause the eye to periodically jerk off target [5]. It is likely, however, that when an observer is attempting to fixate numerous targets successively, accuracy is considerably reduced and misplacements as as large as a degree may become common [7].*" [Ware, Mikaelian 1987]

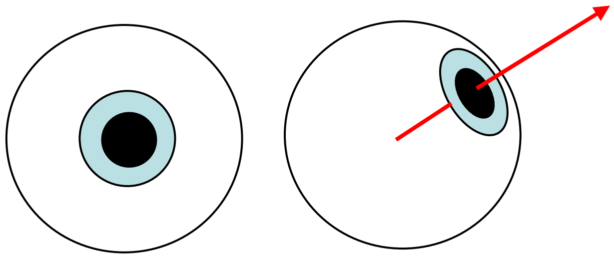
It is definitely interesting to find out the maximum accuracy for positioning tasks with the eye. However, even if it is possible to show that higher accuracy is achievable, it does not mean that this higher accuracy helps for human computer interaction. Most probably positioning tasks close to the possible accuracy limit will demand a high level of concentration and lead to mental fatigue after a short time. The size of the fovea seems to provide a natural and therefore reasonable size for the eyes' pointing accuracy. The situation is comparable with the pointing accuracy of fingers. Here the natural pointing precision is given by the size of the fingertip. Although we are able to type on small keyboards of mobile devices, we prefer keys in the size of our fingers, especially when interacting for longer periods.

Within this thesis it was not possible to research whether enhanced interaction technologies could achieve a better pointing precision with the eye because the eye tracker's precision was not good enough. However, it seems to be clear that we cannot achieve the same precision for pointing with the eye as we do with a mouse. With a mouse and current screen resolutions it is possible to address a single pixel. The 0.16° mentioned in the citation as the best positioning accuracy of the eye are six pixels for the eye tracker used for this work. This means that it is difficult to position a text cursor accurately by gaze. Therefore, it is not possible to replace the mouse by an eye tracker without any changes in the graphical user interface.

The low accuracy of the gaze has impact for the calibration process (see 2.4.3). The position data for the calibration are already inaccurate and consequently the calibration is inaccurate too. To get a better calibration it is necessary to offer calibration targets several times and average over the corresponding gaze positions.

This leads to the second question whether a calibration procedure is necessary. It is possible to estimate the optical axis of the eye by the shape of the pupil. When looking directly towards the camera the pupil has the shape of a circle. For any other direction the pupils' shape is an ellipsis. It is possible to calculate the optical axis from the parameters of this ellipsis (see Figure 19) by applying Hough transformations to the picture delivered from the camera. The method is patented [DE102005047160] and [Thomae, Plagwitz, Husar, Henning 2002] presents an eye tracker working on this principle. [Shih, Wu, Liu 2000] presents another so-called calibration-free gaze tacking technique based on 3D computer vision to estimate the direction of the gaze. The system needs two cameras and was not really built but evaluated by computer simulations. The method is also patented [EP 1 228 414 B1]. Morimoto et al. used a similar approach [Morimoto, Amir, Flickner 2002] but instead of multiple cameras they used several light sources to create several glints and a single camera. This method is also patented [US 6578962]. Noris et al. recently presented a further calibration-free eye-gaze direction detection using Gaussian processes [Noris, Benmachiche, Billard 2008].

Figure 19: Direction of view and the elliptical shape of the pupil



The reason why such eye trackers still need calibration lies in the position of the fovea. The fovea is not located directly opposite of the pupil but 4° to 8° higher (see Figure 18). This causes that the optical axis (or pupillary axis) is not the same as the visual axis (or line of sight). Because the offset of the fovea differs from person to person, a calibration is still necessary and such eye trackers are not calibration-free. In contrast to the eye-tracking method presented in 2.4.3, however, these eye-tracking methods based on estimating the optical axis need only a single target for calibration.

It is possible to hide a calibration process from the user by presenting an introduction screen with non-static content such as animations or blinking targets which attract the eye gaze. As the so-called calibration-free eye tracking methods need only one point for calibration, they can do this implicit calibration much more efficient than the eye trackers using the standard corneal reflection method, which needs at least three points.

Lessons learned:

- The accuracy of an eye tracker for interaction does not need to be better than ±0.5° (size of fovea)
- All eye tracking devices need a calibration procedure (position of fovea)

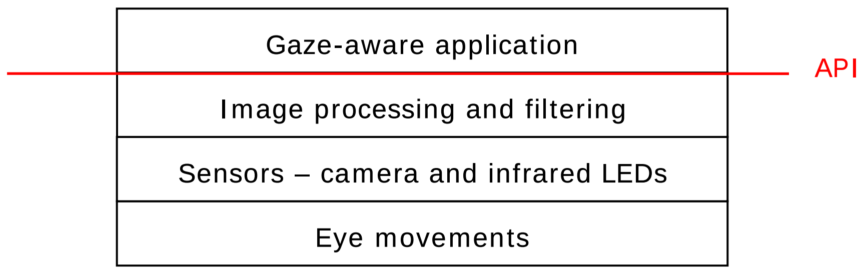
3.3 Statistics on Saccades and Fixations

As mentioned in the introduction, a lot of research on eye movements was done already. Statistics on saccades and fixations are no novelty. The main reason to present such statistics here is to demonstrate the capabilities and limitations of the eye tracker used for the studies. The statistics also provide reasonable values and ranges for times, distances and speeds of eye movements. Such data are valuable for the design of gaze-aware user interfaces and for the discussion on the feasibility of eye gaze interaction methods. In contrast to general research on eye movements the statistics presented here explicitly focus on computer activities. It shows that different computer activities result in different numerical values for statistical measures.

The situation for commercial eye trackers is that they come as a complete system and the way in which they function is partly a business secret. Especially the source code for the image processing with the detection of fixations and saccades and the calibration data is normally not available. Figure 20 shows the situation in a layer model. All what happens below the API (application programming interface) is unknown. Thus, the statistics presented here should not be seen as statistics on the eye movements but on the data the eye-tracker system delivers, keeping in mind the warning of Salvucci and Goldberg:

"*However, identification (of fixations) is often a critical aspect of eye-movement data analysis that can have significant effects on later analyses.*" [Salvucci, Goldberg 2000]

Figure 20: Layer model for gaze-aware applications



Typically, the coordinate values coming from image processing are very noisy and the eye tracker smoothes the coordinate data before delivering them to the API. As long as the current coordinates of the gaze position are within a threshold distance from the last coordinates reported, an eye tracker does not report new coordinates. Therefore, an eye tracker typically reports only saccades and fixations even when the eye gaze smoothly follows a moving object.

The citation of Salvucci and Goldberg show that it is not even clear what is a fixation. The literature sometimes speaks about meaningful fixations lasting about 300 milliseconds, which is the time the brain needs to recognize something. Normally fixations are defined by a minimum time of eye rest but for the statistics presented here, a fixation is just what the eye tracker reports as a fixation. This can also be a fixation of zero time. There is no meaning for human perception put on the term fixation here. It is introduced in this way to have a terminology to discuss the data delivered by the eye tracker.

3.3.1 The Data

Within the user study “gaze gestures” (see 5.4.3) eight people watched a three and a half minute video and surfed three to four minutes in the Internet while their eye movements were recorded. The aim was to find out how natural eye movements do look like. Two different tasks were chosen as the details of eye movements depend very much on the task.

The ERICA eye tracker used for the studies delivers x- and y-coordinates of the gaze position during fixations at a frame rate of 60 Hz but it reports no data during a saccade. The time interval between reporting two different positions is a saccade and the time interval of reporting the same position is a fixation. In some cases the ERICA eye tracker reports a single x-y value which leads to a fixation time of zero by this way of calculation. It is also possible to assume that such fixations last the time length of a frame of the camera (17 milliseconds) which in consequence leads to a subtraction of one frame for the time of a saccade. This could result in saccades of zero length time. As the ERICA eye tracker does not report data during saccades (see also Figure 51 and Figure 53 in chapter 4) the time of the data gap is the duration of the saccade and the distance of the points reported before and after the gap is the saccade length.

As mentioned above, the data presented here depend on the properties of the low-level filters and for this reason may be ERICA-specific. Nevertheless all the other eye trackers face the same general problem of noise filtering and the data of the ERICA system are not expected to be very different from the data another eye tracker would have produced.

3.3.2 Saccades Lengths

Figure 21, Figure 22 and Figure 23 show the number of saccades occurred, plotted over their length.

The length interval used is one pixel. The data basis is the sum of eight people watching a video or surfing the internet respectively and both for a total time of half an hour. In Figure 21 it is easy to see that the eye tracker does not report saccades with a length below 16 pixels (0.45°) which is also the accuracy stated in the technical description of ERICA. This threshold filters tremor, drift, and micro-saccades. The figure also shows that many saccades do not exceed 45 pixels (1.25°) in length.

The two peaks at 16 and 28 pixels and the gap in between are surprising and most probably caused by the low-level filter of the eye-tracking software and are not a property of eye movements. A request at the manufacturer did not bring a satisfying answer. The strange gap in the data for the saccade lengths confirms the warning of Salvucci and Goldberg given above.

Figure 21: Saccade statistics for watching a video with full vertical scale

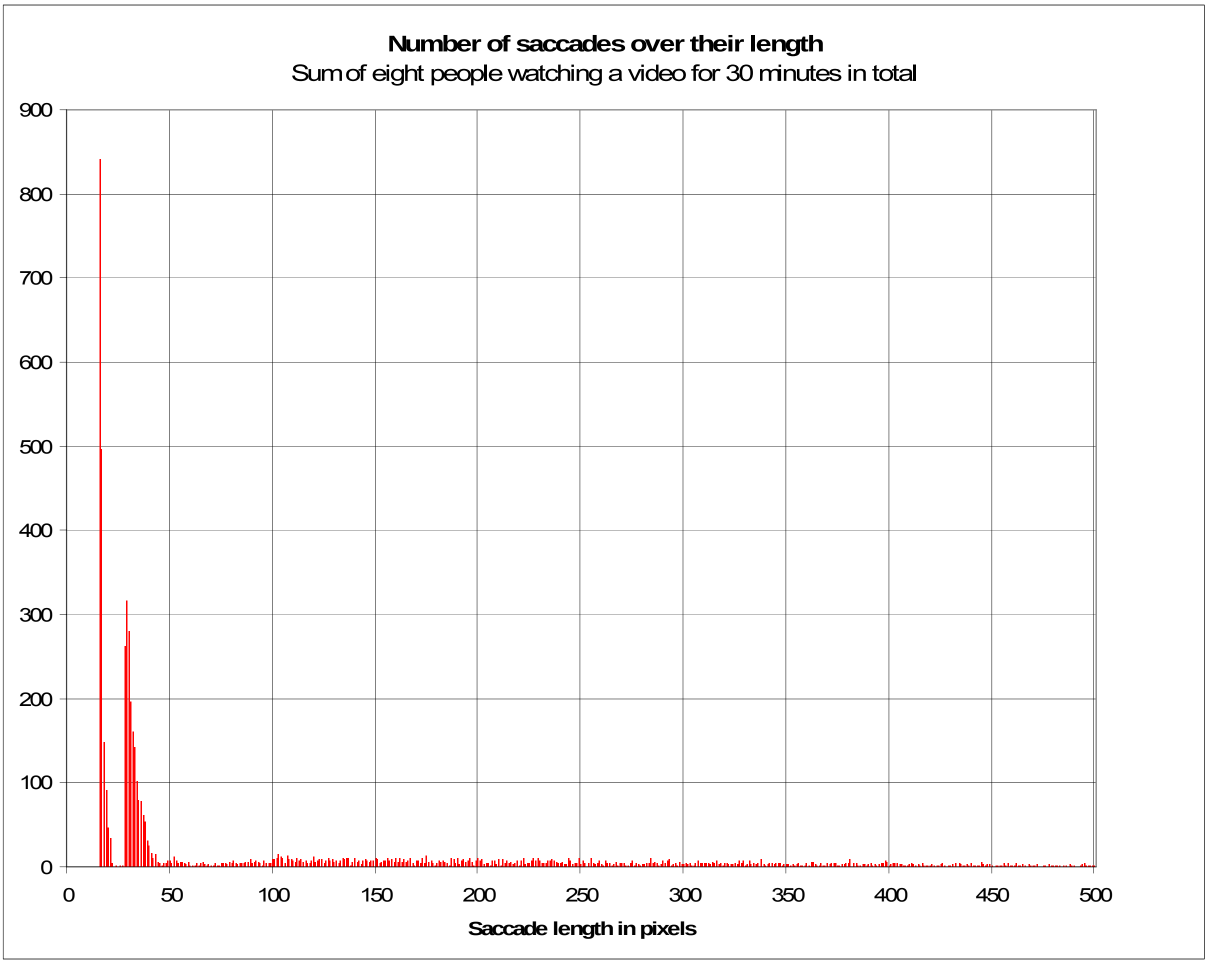


Figure 22 shows the same statistics as Figure 21 for watching a video but with a partial vertical scale. Figure 23 shows statistics for the eye movements while surfing in the Internet. It does not look different at the first glance but there are some differences, which will be discussed in 3.3.5.

Figure 22: Saccade statistics for watching a video on a partial vertical scale

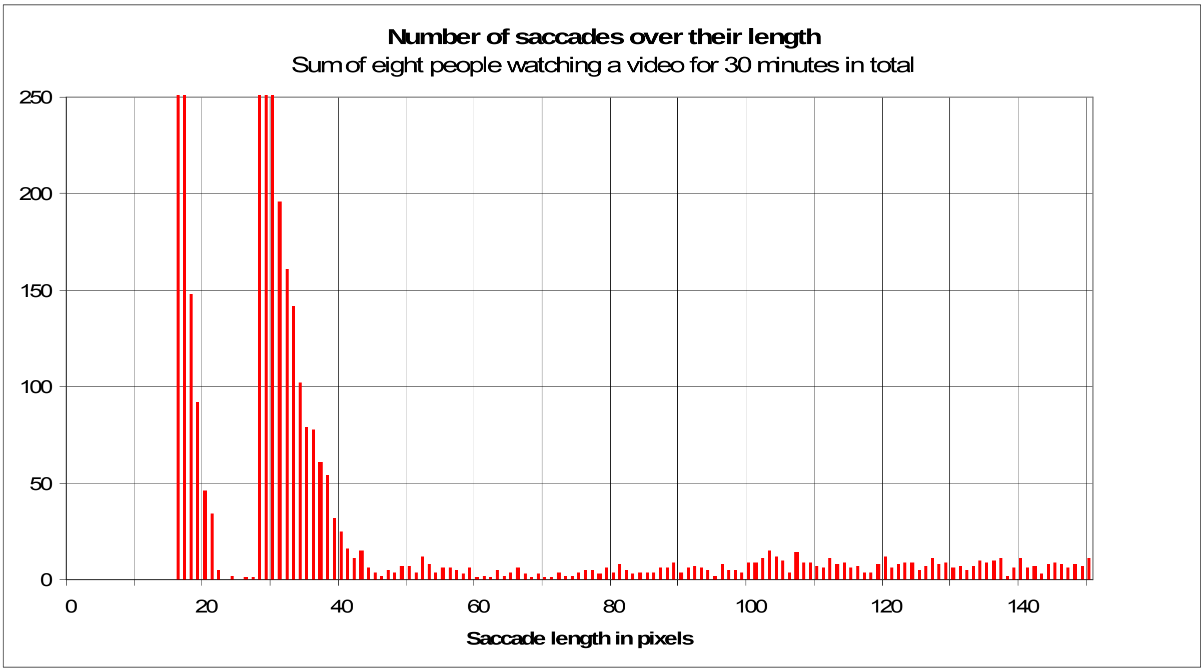
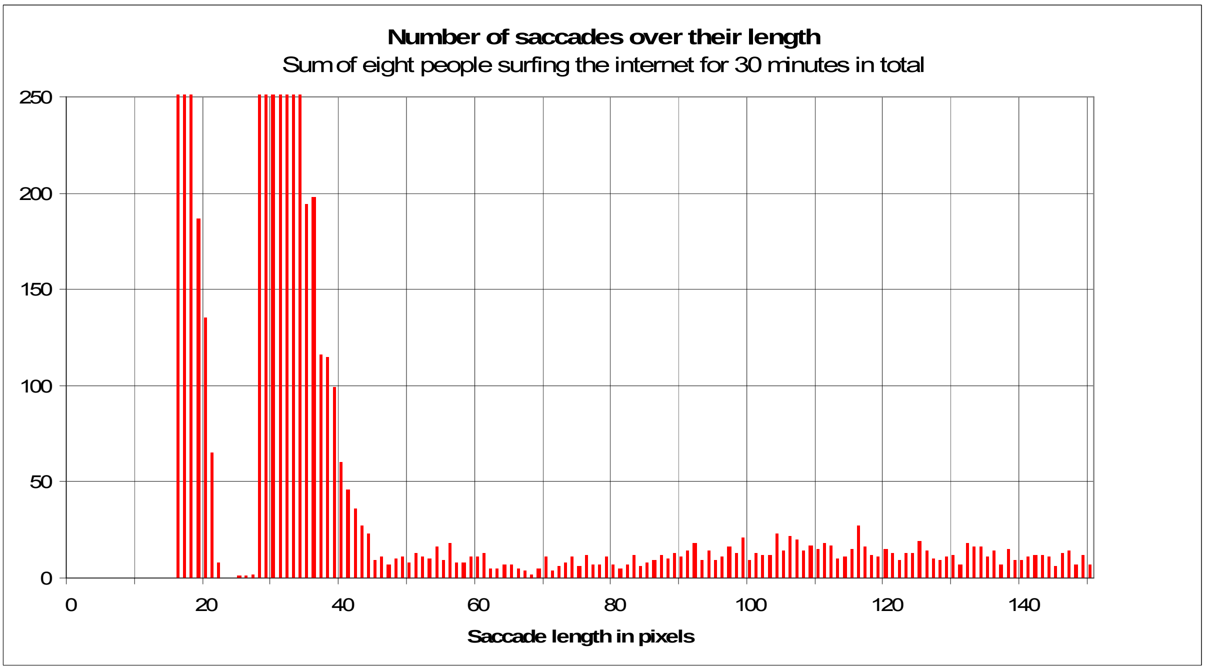


Figure 23: Saccade statistics for surfing the Internet on a partial vertical scale



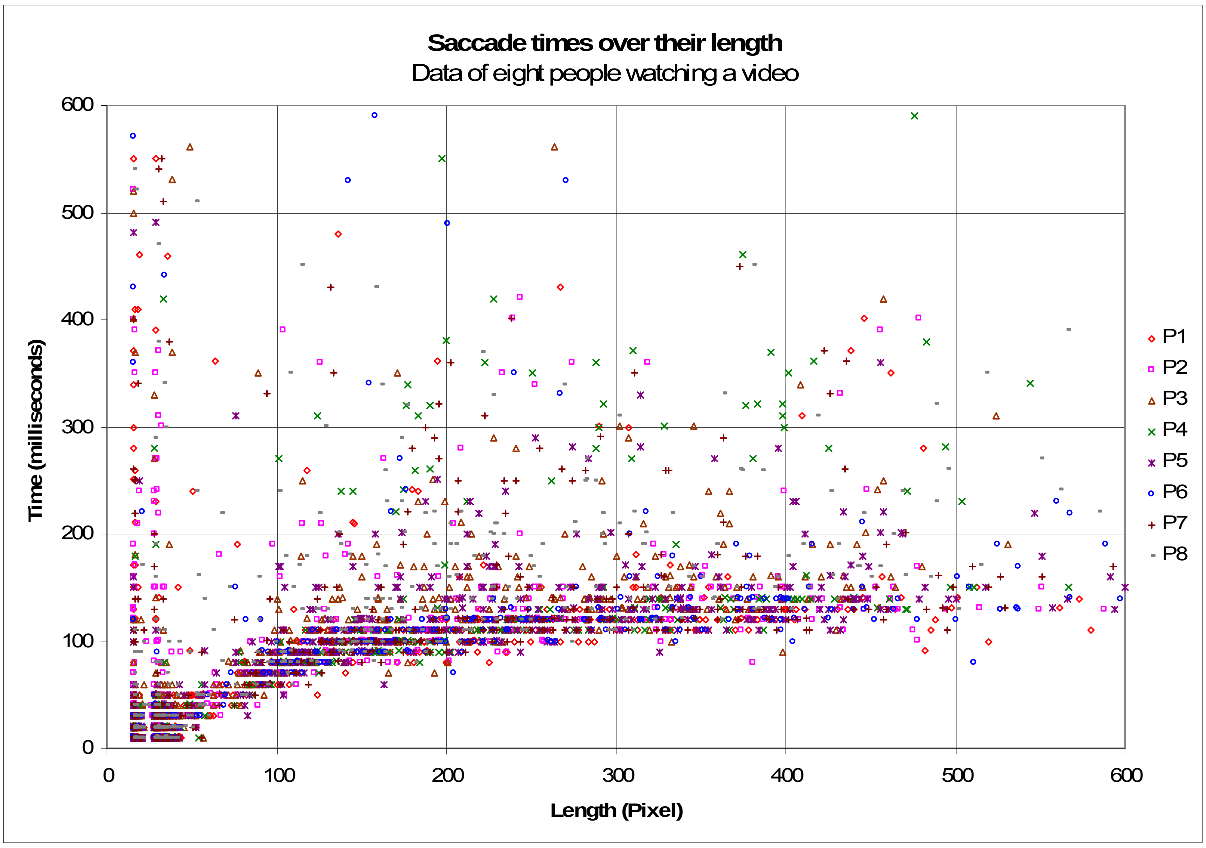
Lessons learned:

- There are no reported saccades below a threshold length (16 pixels / 0.45° for ERICA)
- Do not trust in values for saccades with lengths close to the threshold (below 36 pixels / 1.0° for ERICA)
- The majority of saccades are below 45 pixels / 1.25° in length

3.3.3 Saccade Speed

Figure 24 shows a plot of saccade times over their lengths for eight participants watching a video. On the length scale the data show the filter cut-off at 16 pixels and the gap from 23 to 28 pixels. On the time scale the 10 millisecond resolution of the operating system is recognizable. The data also show that the individual differences of the eight participants (P1-P8) are small.

Figure 24: Saccade times over their length



The plot has three areas, one area where there are no data, one area where the data are dense and another area where the data are scattered. The area with the dense data represents the cases where the eye moves at maximum acceleration. There are no eye movements faster than that and this explains the empty area below. The area above with the scattered data can be either slow or composite eye movement, which the eye tracker reports as a single saccade.

The curve describing the area with dense data has a very flat slope for saccade lengths above 180 pixels (5°). Longer saccade lengths can be performed with nearly no increase in time. This is an important fact for the size of gaze gestures presented in chapter 5. Chapter 3.4 discusses the functional relation of saccade time and length.

Lessons learned:

- Saccades of 180 pixels (5°) length need about 120 milliseconds.
- There is little increase in time for longer saccades.

3.3.4 Fixation Times

Figure 25 and Figure 26 show the number of fixations over the fixation time.

Figure 25: Number of fixations over fixation time (watching half hour video)

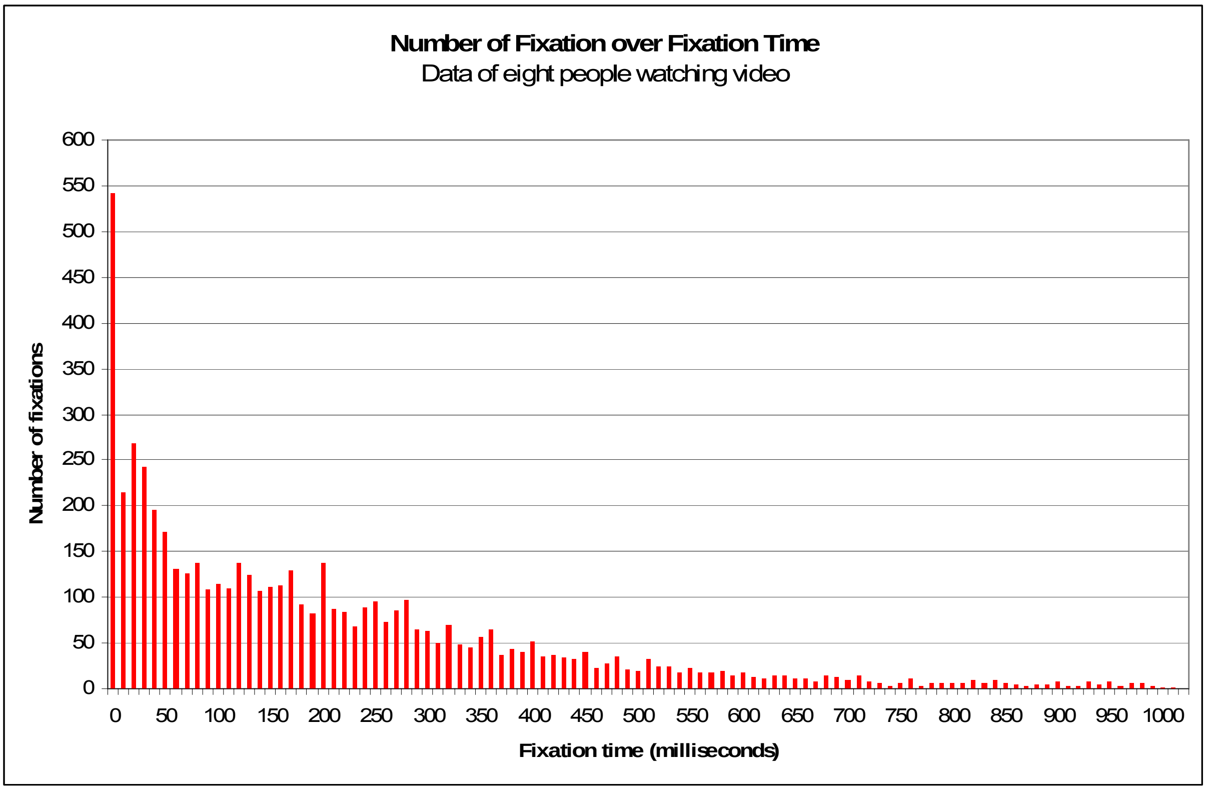
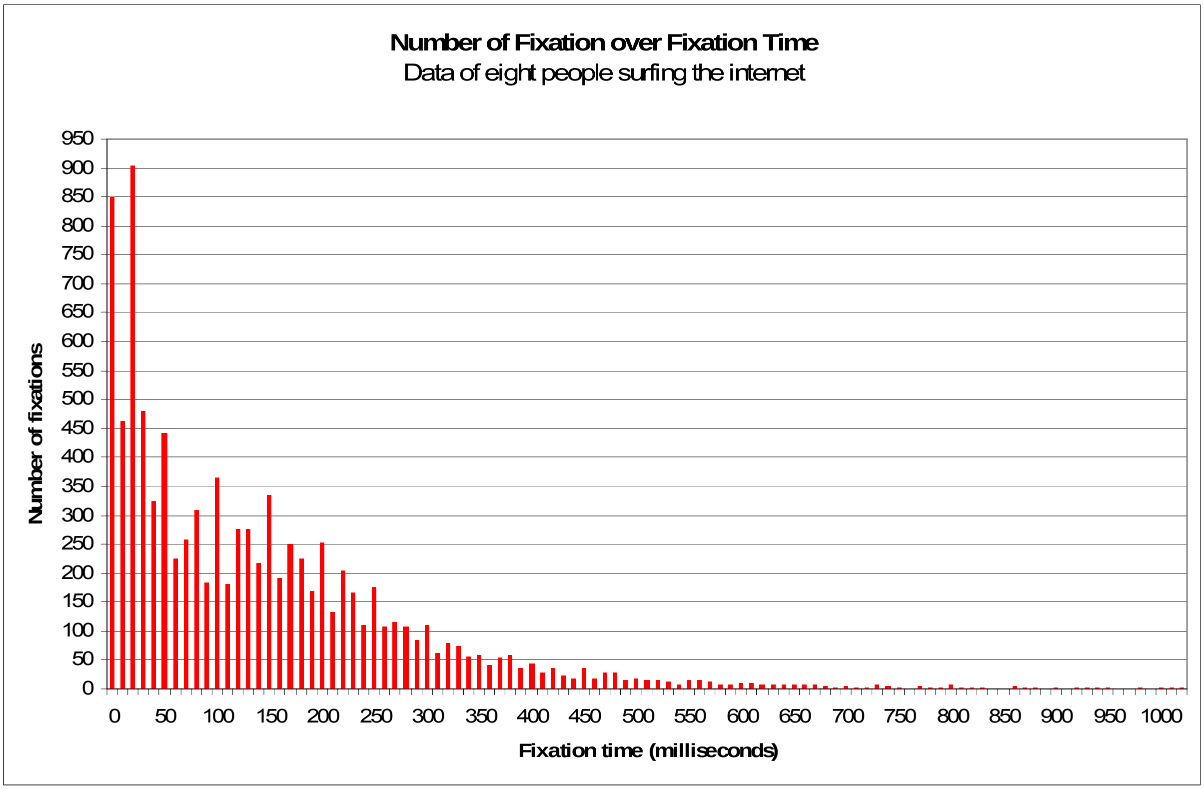


Figure 26: Number of fixations over fixation time (surfing half hour in the internet)



As explained in 3.3.1 fixations with zero time are a consequence of the definition to calculate the time of a fixation. It is also possible to see such fixations as fixations lasting about the time of a frame in the video stream. The low fixation times may have also errors from the interference of the 17 milliseconds frame time and the 10 milliseconds time slice of the operating system.

The number of fixations declines with higher fixation times and long fixation times become very unlikely. This is the reason why the dwell-time method works. However, the figures also show that long fixation times occur occasionally, especially in the relaxed situation of watching a video. This explains that people who operate an eye-gaze interface based on dwell-time method for the first time report that they feel stressed.

When surfing the internet more but shorter fixations occur than watching video. The next chapter will give an explanation.

Lessons learned

- The longer the fixation time the less likely it occurs

3.3.5 Summary of Statistics

The two tasks, watching a video and surfing the internet, are different in character. Watching a video can be seen as passive task. The motion presented in the video forces the speed of the eye movement and the positions to look at. Surfing the internet is an active task. Most web pages have static content and it is the user who decides where to look at and when. Therefore, the eye movements are expected to be different for the two tasks.

Table 1 and Table 2 show mean values for both tasks such as the average number of saccades performed per second, the average saccade time and length and the average fixation time. They also show the percentage of saccades below the ‘trusted length’ of 36 pixels (1°) and 45 pixels (1.25°). See Figure 22 and Figure 23 to understand why 45 pixels were chosen.

Table 1: Mean values of gaze activity parameters for all participants (watching video)

Video	Sac. / sec	Pixel / Sac.	Av. Sac. Time	Av. Fix. Time	≤ 36	≤ 45	Total Time
unit	1 / sec	Pixel	ms	ms	%	%	s
P1	3.68	101.9	67.8	204.0	59.1%	63.2%	216.9
P2	3.43	87.2	69.3	221.9	66.8%	69.7%	219.2
P3	3.45	94.1	71.7	213.9	61.0%	65.6%	231.3
P4	2.74	107.5	76.3	282.0	59.7%	63.6%	228.3
P5	3.24	131.2	73.5	225.6	49.6%	53.8%	216.6
P6	2.79	134.1	99.1	259.3	52.4%	56.7%	225.1
P7	3.49	102.0	73.9	212.5	59.7%	63.1%	217.4
P8	2.76	111.3	91.1	270.6	52.6%	57.1%	219.9
Mean	**3.20**	**108.7**	**77.8**	**236.2**	**57.6%**	**61.6%**	**221.8**
Std. dev.	0.38	16.6	11.2	29.8	5.6%	5.3%	
Std. dev. / Mean	11.8%	15.3%	14.4%	12.6%	9.8%	8.6%	

Table 2: Mean values of gaze activity parameters for all participants (surfing in the internet)

Internet	Sac. / sec	Pixel / Sac.	Av. Sac. Time	Av. Fix. Time	≤ 36	≤ 45	Total Time
unit	1 / sec	Pixel	ms	ms	%	%	s
P1	5,36	73,0	44,9	141,5	70,6%	75,9%	234.0
P2	5,10	73,2	43,9	137,2	69,2%	74,4%	229.4
P3	4,54	109,4	70,3	150,0	56,2%	61,8%	228.9
P4	5,51	69,0	41,5	140,0	74,2%	80,7%	229.8
P5	4,58	105,9	70,2	145,6	62,0%	67,6%	291.1
P6	4,59	106,7	54,7	156,0	54,8%	60,8%	264.0
P7	4,78	108,0	66,0	143,3	54,4%	59,6%	238.2
P8	3,17	123,3	104,5	177,1	51,2%	56,0%	374.7
Mean	**4,70**	**96,1**	**62,0**	**148,8**	**61,6%**	**67,1%**	**261.3**
Std. dev.	0,72	20,9	20,9	12,9	8,7%	9,0%	
Std. dev. / Mean	15,4%	21,8%	33,7%	8,6%	14,2%	13,4%	

Table 3: t-test for “watching video” versus “surfing in the internet”

	Sac. / sec	Pixel / Sac.	Av. Sac. Time	Av. Fix. Time	≤ 36	≤ 45	
t-test	0.00040	0.13136	0.05318	0.00003	0.19918	0.09981	

A look at the standard deviations reveals that in the case of watching a video the individual differences are in the range of ±15% around the means. The eye movements are driven by the same input. For surfing the internet the freedom where and when to look is much higher and consequently the individual differences are bigger. None in all the individual differences in eye movements is extreme. When designing gaze-aware user interfaces expected values in a range of ±50% around their means should be sufficient for nearly all users.

Table 3 shows the values for a paired Student's t-test comparing both tasks. The meaning of this value is the probability that the data sets compared are taken from a distribution with the same mean. The values that differ significantly ($p = 0.04\%$ and $p = 0.003\%$ means extremely strong significance) are the saccades per second and the average fixation time (see also Figure 25 and Figure 26). At the first moment it seems to be a little bit paradox that people perform less eye movements when watching a video full of action than surfing on static pages in the internet. The reason for that lies in the process of reading (see chapter 6.4.1) which consists of saccades with short fixations in between. When reading the eye moves as fast as possible while when watching a movie the eye waits for something to happen.

The results of the statistics are good news for attempts to use eye-gaze activity for context awareness (see also chapter 6). The strong significance proves that a gaze-aware system can guess the users activity quite well. The low individual differences in the gaze activity parameters justify the hope that such activity recognition will work with universal threshold values and does not need an adaptation for individual users.

Lessons learned:

- The saccades per second and the average fixation times depend strongly on the task.
- The individual differences in gaze activity are not extreme (below 35%).
- Activity recognition from eye movements is possible.

3.4 Speed and Accuracy

The standard textbooks on eye movements state that the eyes perform ballistic movements, that the saccadic movement is pre-programmed, and that there is no visual feedback during a saccadic movement as the speed of a saccade is quicker than the reaction time of the photoreceptors on the retina. The speed of eye rotations can reach 700°/s while the process of photoreception takes about 20 ms to respond [Land 2006]. This means that the eye is nearly blind during a saccade (saccadic suppression) [Goldstein 2002, p. 354], [Duchowski 2002, p. 44]. If the eye is blind during a saccade, there is no control-feedback loop and consequently no speed-accuracy trade-off. This means that the time for a saccade depends on the distance (rotation angle) and the acceleration of the eye muscles only.

Nevertheless, some voices of the HCI community suggest the validity of Fitts' law for eye movements (see 3.4.3). Fitts' law assumes a dependency of the movement time not only on the amplitude of the movement but also on the target size. The reason for this assumption lies in the fact that Fitts' law successfully describes the performance with a pointing device like the popular computer mouse. As the eye also does pointing operations, it is natural to assume the same law, but also naïve.

The HCI community already heavily researched Fitts' law and further contributions should be done only for good reasons. However, with a chapter on eye gaze as pointing device in this thesis, it cannot be ignored here.

The following sections present models for eye speed as used in psychology, introduces Fitts' law and its variations and comments on the debate in the HCI community whether Fitts' law is valid for the eyes. The three subsequent sections present experiments and data from user studies and discuss how well they fit to the possible models. The last section concludes that Fitts' law does not apply to the eyes and provides design rules for size and distance of interaction elements in gaze-aware interfaces.

3.4.1 Eye Speed Models

In 1977, Carpenter measured the amplitude and duration for saccades [Carpenter 1977]. He found a linear relation between the time T of a saccade and its amplitude A:

$$T = 2.2\ \text{ms}/° \cdot A + 21\ \text{ms}$$

For angles above 3° this formula works well for the data presented in the previous section. The data in Figure 24 suggest a factor of 3 ms/° and an offset of 70 ms for the quickest saccades above 3° length. This is not a contradiction because considering slower and shorter saccades changes the numerical values; it is a question on how to average the data.

In 1989 Abrams, Meyer, and Kornblum [Abrams, Meyer, Kornblum 1989] suggested a model where the muscle force is constantly increasing over time which means the acceleration $a(t)$ is constantly increasing with the time:

$$a(t) = k \cdot t$$

with *k* as a constant for the increase of acceleration. Abrams et al. state that this relation is valid for angular movements up to 15°. The distance *s(t)* (strictly the rotation angle) covered within time *t* can be calculated by integrating twice over time.

$$s(t) = k/6 \cdot t^3$$

In this model a saccade would be a movement of constantly increasing acceleration up to half of the distance and a constantly increasing deceleration for the second half. The relation between the time *t* and the covered distance *s* would be (with *c* just another constant value):

$$t = c \cdot s^{1/3}$$

The sections 3.4.4 and 3.4.6 deal with the comparison of Fitts' law versus the approach of constantly increasing acceleration.

3.4.2 Fitts' Law

Since Fitts published his famous paper "The information capacity of the human motor system in controlling the amplitude of movement" in 1954 [Fitts 1954] we know that speed and accuracy of muscle movement depend each other.

Fitts deduced his law from concepts of Shannon's information theory [Shannon 1948] and therefore it relates to hard science. It is one of the few human-centric interaction formulas. Understanding the nature of rapid aimed movements is of high relevance for human computer interaction as this is what we do when clicking a target with the mouse. For these reasons, much research has been done on the application of Fitts' law to pointing devices for example [Card, English, Burr 1978], [MacKenzie, Sellen, Buxton 1991] just to mention some. Fitts' law is very popular in the HCI community and researched and referenced in (too) many papers. MacKenzie published a "Bibliography of Fitts' Law Research" on the internet [MacKenzie@ 2002]. It has 310 entries and the last update is from 25-Jun-2002. MacKenzie also published the articles "Movement Time Prediction in Human-Computer Interfaces" [MacKenzie@ 1995], [MacKenzie 1995] and "A Tool for the Rapid Evaluation of Input Devices Using Fitts' Law Models" together with Buxton [Buxton@] in the internet which both introduce the issue.

The existence of an ISO standard [ISO 9241-9@], which explains how to measure the quality of pointing devices based on Fitts' law, reflects the high importance of this law. For a re-analysis of the experiments done by Card et al. [Card, English, Burr 1978] in terms of the ISO standard see [MacKenzie, Soukoreff 2003]. Unfortunately, the ISO defines a single constant, the throughput, while Fitts' law has two constants to describe the quality of a pointing device. For a detailed discussion of this topic, see [Zhai 2004]. It is also questionable whether the ISO standard reflects the performance of a pointing device; Fitts' law constants reflect the performance of the nervous system of humans.

Fitts' law is often introduced as a speed-accuracy trade-off, which describes the relation of the time *t* to hit a target of size *W* (from width) in the distance *A* (from amplitude). Although this is true, it was not Fitts' message. He wanted to find out whether the performance of a motor task is a question of physical strength or of the nervous system's information processing capacity. In one of his experiments, where the subjects had to tap with a stylus on metal stripes of different widths, he used styluses with different weights. The fact that the performance did not change with the weight of the stylus proved that the performance is not a question of physical strength.

Fitts assigned an index of difficulty $ID = \log_2(2 \cdot A/W)$ to each of his motor task which is measured in bits. The assumption that there is a maximum capacity for information processing, measured in bits processed per second, leads to Fitts' law.

Derivation of Fitts' Law

Many derivations of Fitts' formula give only a vague reference to Shannon's theory. However, there is also the discrete-step model for a derivation of Fitts' formula. The model is very simple and can be found in different variations, for example on Dix' internet site [Dix@].

The model starts with a step-wise movement towards the target. Every step consists of aiming at the target, moving the pointer to the target, and estimating how close the pointer to the target is. The derivation uses two basic assumptions:

- the distance to the target after each step is proportional to the distance at the beginning of the step
- every step takes the same amount of time

The first assumption reflects the scalability of nature. The second assumption reflects a constant information processing power.

Figure 27: Step-wise movement towards target

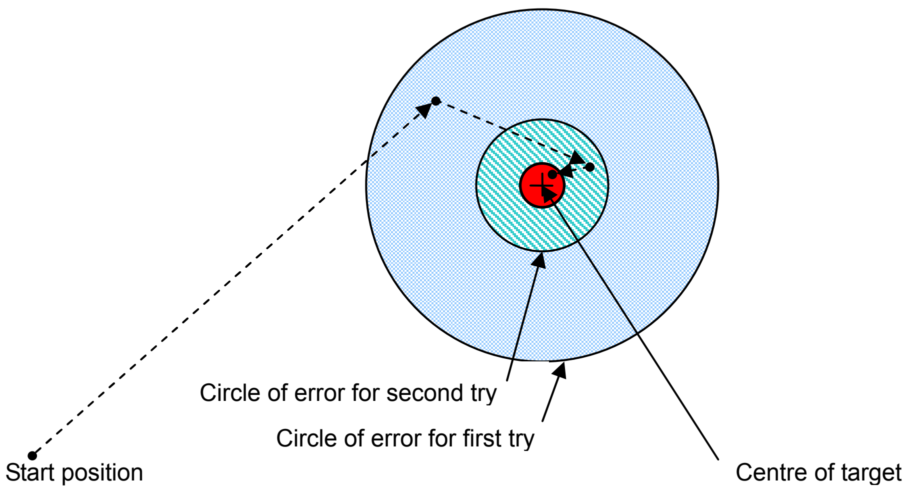


Figure 27 shows discrete steps of sensing and movement. With each step, the pointer gets gradually closer to the target. Figure 27 shows circles of error, which indicate the area where the pointer most probably will end. These circles of error represent a probability density for the inaccuracies of movement. The probability density can be symmetrical but this is not a demand. The derivation only demands that the expectation value for the distance to the target after a step is proportional to the distance at the beginning of the step. This can be the case for unsymmetrical probability densities for example when grabbing a cup where overshooting is not allowed.

Let the distance to the target at each stage be A_i with the initial distance $A_0 = A$. After each step, the average distance to the centre of the target A_{i+1} is a constant fraction λ of the distance A_i at the beginning of the step.

$$A_{i+1} = \lambda \cdot A_i$$

and consequently:

$$A_i = \lambda^i \cdot A$$

The process stops after n steps when the distance to the target centre is less than the radius R of the target:

$$A_n = \lambda^n \cdot A < R$$

From this follows:

$$n = \log(R / A) / \log(\lambda)$$

Each step takes a fixed time τ and there will be some initial time *a* for the brain to get started. The total time T to reach the target is:

$$T = a + \tau \cdot n = a + \tau \cdot \log (R /A) / \log (\lambda)$$

$$T = a + b \cdot \log (A / R)$$

where $b = - \tau / \log (\lambda)$. As the pointer gets closer to the target with each step, λ is smaller than 1 and $\log(\lambda)$ is negative so b is positive.

The model is very simple and has its problems. For example, the speed of the pointer has its maximum at the initial move. However, the pointer is in rest at the beginning and a physical object cannot accelerate within zero time. The model also neglects many factors which influence pointing actions of a subject. A real movement shows inaccuracies by tremor which is not related to the target distance. The model does not cover the subjects' mood, their nervousness, their eyesight, chemical substances like alcohol or caffeine in their blood, or distraction by the environment.

Of course, every model has its limits and does not explain reality completely. However, it is not always helpful to emphasize that in reality the things are much more complicated. The benefit of a model lies in its simplification. The models in physics, for example, allow giving a formula for the motion of a perfect sphere on a perfect surface, but not of a stone thrown down a hill.

Variations of Fitts' Law

Fitts did not give a Fitts' law formula in his publication. He only introduced the index of difficulty ID. Subsequent researchers suggested different variations for a formula of Fitts' law.

Frequent variations are:

$$t = a + b \cdot \log_2(2 \cdot A / W) \qquad (1)$$

$$t = a + b \cdot \log_2(A / W + 0.5) \qquad (2)$$

$$t = a + b \cdot \log_2(A / W + 1) \qquad (3)$$

$$t = a + b \cdot \log_2(A / W) \qquad (4)$$

In these formulas *t* is the mean time to hit the target. The constants *a* and *b* are measured in a Fitts' law experiment. Constant *a* is called the non-informational part and has the unit seconds. Constant *b* is the informational part in seconds per bit.

Formula 1 uses an ID as defined by Fitts. Formula 2 was introduced by Welford [Welford 1968] and formula 3 by MacKenzie [MacKenzie 1989]. Factoring out the 2 in formula 1 is legal and leads to formula 4, however with a different value and interpretation for the *a*-constant (see below).

It is worth to mention that the definition of the distance to the target influences the formula. Let A_c be the distance to the centre of the target and A_e the distance to the edge of the target. The different distance definitions relate to each other by:

$$A_c = A_e + W/2$$

Consequently, the index of difficulty ID turns into:

$$\log_2(2 \cdot A_c / W) = \log_2(2 \cdot A_e / W + 1)$$

For this reason reports on a Fitts' law experiment should clearly state which definition of distance it uses. Sadly, many publications miss a clear statement.

Interpretation of the a- and b-Constants of Fitts' Law

The *b*-constant is called informational part and has the unit seconds per bit. It states how much time is needed to process a bit. The *a*-constant has the unit seconds. For formula 1 it is the reaction time. The derivation explicitly added this constant as reaction time. In Formula 4 the interpretation of the *a*-constant has a different meaning:

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

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

$$= a' + b \cdot \log_2(A/W) \text{ with } a' = a + b$$

The calculation shows that the *b*-constant does not change by factoring out the 2. As the value of the a-constant changes according to the used formula, many authors call it non-informational part.

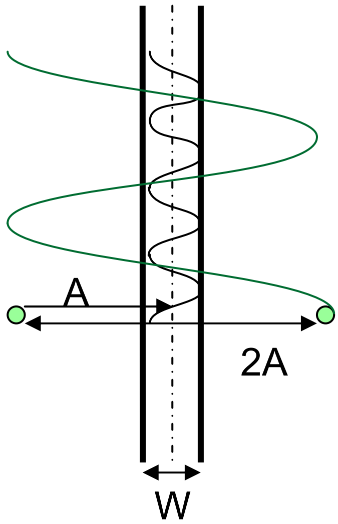
The Confusion on Fitts' Law

It was Fitts himself who started the confusion by the way he introduced the factor 2:

"*The use of 2A rather than A is indicated by both logical and practical considerations. Its use insures that the index will be greater than zero for all practical situations and has the effect of adding one bit ($-\log_2 1/2$) per response to the difficulty index. The use of 2A makes the index correspond rationally to the number of successive fractionations required to specify the tolerance range out of a total range extending from the point of initiation of a movement to a point equidistant on the opposite side of the target.*" [Fitts 1954, p. 267].

Actually, when just looking at the formula, the factor 2 does not ensure that the ID will be positive; adding 1 can achieve this. Nevertheless, Fitts is right. Perhaps Fitts did not express it in an elegant way, but within the analogy to information theory, he mapped the amplitude of the noise to the width of the target. However, the width of the target corresponds with the difference from peak to peak. In consequence, he took also the peek-to-peek value for the amplitude of the movement, which is 2A (see Figure 28).

Figure 28: Signal and noise amplitudes



Comparing Fitts' definition of the index of difficulty with the derived formula of the discrete-step model, it turns out that both definitions are equal 2A/W = A/R. With this understanding the problem with negative IDs dissolves and Fitts is right with his statement "*the index will be greater than zero for all practical situations*". For a negative ID the distance to the target has to be smaller than the radius. This means that the pointer is already inside the target and the entry to the target happened in the past. Nothing is wrong with formula 1.

The confusion in the HCI community on which formula to use dates back to 1991 where MacKenzie et al. [MacKenzie, Sellen, Buxton 1991] published a paper which mentions three different formulas (The numbers for the formulas in the following citation are equal to the numbers used in this document);

"*More formally, the time (MT) to move to a target of width W which lies at distance (or amplitude) A is*

$$MT = a + b \log2(2A / W) \qquad (1)$$

where a and b are empirical constants determined through linear regression. A variation proposed by Welford (1968) is also widely used.

$$MT = a + b \log2(A / W + 0.5). \qquad (2)$$

…

There is recent evidence that the following formulation is more theoretically sound and yields a better fit with empirical data (MacKenzie, 1989):

$$MT = a + b \log2(A / W + 1). \qquad (3)$$

In an analysis of data from Fitts' (1954) experiments, Equation 3 was shown to yield higher correlations than those obtained using the Fitts or Welford formulation. Another benefit of Equation 3 is that the index of difficulty cannot be negative, unlike the log term in Equation 1 or 2 Studies by Card et al. (1978), Gillan, Holden, Adam, Rudisill, and Magee (1990), and Ware and Mikaelian (1987), for example, yielded a negative index of difficulty *under some conditions. Typically this results when wide, short targets (viz., words) are approached from above or below at close range. Under such conditions, A is small, W is large, and the index of difficulty, computed using Equation 1 or 2, is often negative. A negative index is theoretically unsound and diminishes some of the potential benefits of the model.*" [MacKenzie, Sellen, Buxton 1991]

From the discussion above it is clear that negative IDs cannot happen with formula 1. A negative ID for formula 1 means that the pointer is already inside the target. Otherwise it means that target size or distance are not proper defined and not that there is something wrong with the formula.

In addition, there are doubts whether formula 3 is more theoretically sound. MacKenzie writes:

"*Shannon's Theorem 17 expresses the effective information capacity C (in bits × s-1) of a communications channel of band B (in s-1) as*

$$C = B \log2((P + N) / N) \quad (4)$$

where P is the signal power and N is the noise power (Shannon & Weaver, 1949, pp. 100-103).

It is the purpose of this note to suggest that Fitts' model contains an unnecessary deviation from Shannon's Theorem 17 and that a model based on an exact adaptation provides a better fit with empirical data. The variation of Fitts' law suggested by direct analogy with Shannon's Theorem 17 is

$$MT = a + b \log2((A + W) / W)$$

It is revealing to examine the source Fitts cites in his paper at the point where he introduces the relationship (Fitts, 1954, p. 368). His derivation is based on Goldman's Equation 39 (Goldman, 1953), which is similar to Fitts' law except in its use of the terminology of communications systems:

$$C = B \log2(P / N) \quad (6)$$ " [MacKenzie 1989].

MacKenzie does not give an explanation why Fitts' model contains an unnecessary deviation. He also does not explain what is analogue to what and why it is legitimate to apply an analogy.

The simple questions regarding the direct analogy are:

- Why does Shannon's Theorem 17 apply and not another theorem? Theorem 17 delivers capacity and not time.
- Why does the power of the noise map to the target width, which means the diameter, and not to the radius? Amplitudes should map to radius or half of the target width respectively.
- What happened to the square? The power of the noise is proportional to the square of the noise amplitude (or variance of the noise, see also the footnote in Fitts' publication [Fitts 1954]). In the case of Goldman's Equation the square can be drawn out of the logarithm and doubles *B*.

Is it possible that the direct analogy – take the distance as the signal power and the diameter as the power of the noise – is a little bit too direct or in other words naïve?

MacKenzie continues:

"*Fitts recognized that his analogy was imperfect. The "2" was added (see Equation 1) to avoid a negative ID when A = W ; however, log2(2A / W) is zero when A = (W / 2) and negative when A < (W / 2). These conditions could never occur in the experiments Fitts devised. Other researchers, however, have reported experimental conditions with ID less than 1 bit (Drury, 1975), or with a negative ID (Crossman & Goodeve, 1983; Ware and Mikaelin, 1987). It is noteworthy that, in the model based on Shannon's theorem (see Equation 5), ID cannot be negative.*" [MacKenzie 1989]

Actually, Fitts did not state that his analogy is imperfect. Again and as already discussed above, the ID cannot become negative in Fitts' formula (as long as the pointer is outside the target). It is also not clear what level of understanding the citation addresses. People who are familiar with information theory do not need an explanation under which conditions log(2A/W) is negative. Doubts that MacKenzie's formulation is more theoretically sound seem to be justified.

MacKenzie claims that empirical data provide a better correlation with his formula. The question whether a better correlation proves a formula is left to the statistics experts. However, it is always possible to find a formula that matches recorded data. Such formula is good for engineering purposes but definitely, it does not provide new insights or deeper understanding. There is also doubt whether the correlation used by MacKenzie is a legitimate value, as it seems that he used the correlation on pre-averaged data (see also the discussion in the next section).

MacKenzie shows that the correlation gets better when adding 0.5 to the argument of the logarithm and that it gets even better when adding 1. Following his idea (despite the doubts whether this is legitimate) raises the question whether the correlation will increase further when adding values bigger than 1. Table 4 shows the correlations given in MacKenzie's publication and correlations calculated from a data record in the user study presented in 4.4. The explanation why the correlation in the data from the user study is so low will be given in the next section. The data show that the correlation from the user study is best when adding 2. The user study used only two different target sizes and it was not designed with the intention testing Fitts' law. Therefore, the data presented here do not serve as a proof for anything. The reason to present it here is to justify the raised question and definitely not to suggest a further variation of Fitts' formula.

Table 4: Correlations presented by MacKenzie and of data recorded within this work

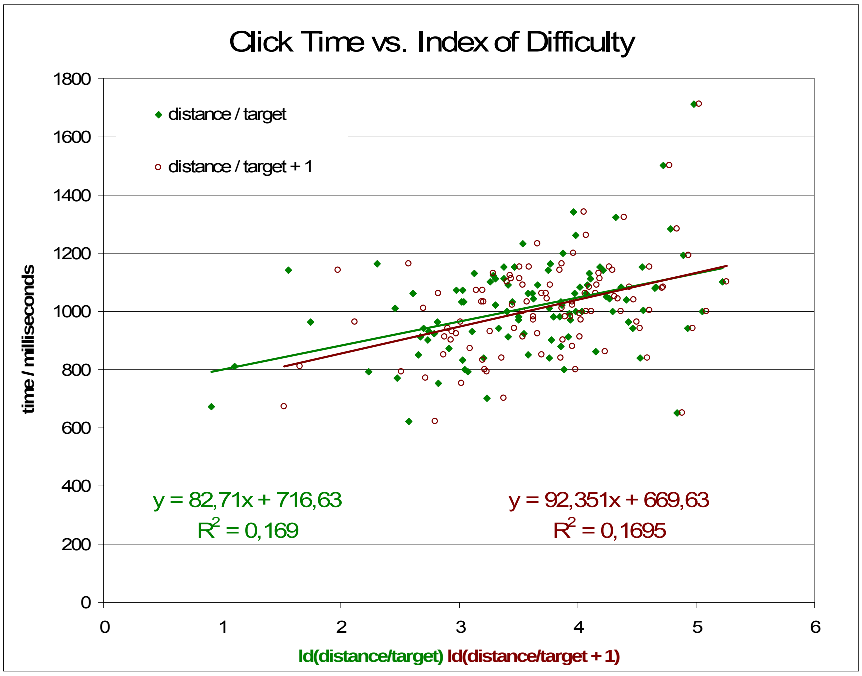
Formula	**MacKenzie**	**user study presented in 4.4**
$\log_2(2\,A / W)$	0.9831	0.1690
$\log_2(A / W + 0.5)$	0.9900	0.1693
$\log_2(A / W + 1)$	0.9936	0.1695
$\log_2(A / W + 2)$	-	0.1696
$\log_2(A / W + 3)$	-	0.1695

It is possible to see this data as a hint that Fitts' formula is not perfect. However, there is no evidence that a better formula should be any of formulas given in the last four rows of Table 4. A better formula might have an additional term, which depends only on the distance, reflecting the time it needs to accelerate parts of the body (see the remark after the derivation of Fitts' formula from the discrete-step model).

The Effect of Different Fitts' Law Formulas

In a setup to measure Fitts' law the distance to the target *A* is normally much bigger than the width *W* of the target. Consequently *A/W* is much bigger than 1 and the additive term does not contribute much to changes in the value of *b*. For the cases where the width of the target is bigger than the distance to the target it is questionable whether Fitts' law is still valid. In this case it needs at least a precise definition of the distance to the target. For completeness of the discussion Figure 29 compares the effect of different definition for the ID. The solid points are for the ID without additive term and the hollow points are for the ID which adds 1. The effect is not extreme but changes the value for the *b*-constant of about 10 percent in the given example.

Figure 29: Performance of one participant for the click-the-target task with a classical mouse. Comparison of different definitions for the index of difficulty



Conclusion

Fitts' introduction of factor 2 is correct. However, the factor 2 should be interpreted as belonging to the target width so that the formula uses the target's radius instead of its diameter. Using the radius of the target instead of the diameter clarifies the problem with negative IDs. If the distance to the target is smaller than the target radius it means that the pointer is already inside the target – the entry to the target happened in the past. Using Fitts' formula for Fitts' law has also the advantage that the *a*-constant is the reaction time and not only some non-informational constant. In the case of a Fitts' law experiment with a mouse device, where people have to click into a target, the reaction time is not only the time for the brain to get ready to move the mouse but also the time to click the mouse key.

We should keep in mind that Fitts' law is a statement on mean values and the noise in the data is high. Because of the high noise, it is difficult to decide from experimental data which variation of Fitts' law is the correct one. The variations in the timings of a single individual are much higher than the difference of times resulting from the different formulas. For practical purposes it does not matter which formula to choose. However, it would be nice if the HCI community could reach an agreement on one formula for Fitts' law. There is no reason not to take Fitts' formula for Fitts' law. It is not satisfying to have a free choice for the variation of Fitts' law on one hand, but strict demands for scientifically valid statistical evaluations on the other hand.

For practical purposes it would be more useful to state the accuracy of the measured constants. A statement like "Our b-constant has the value 0.12 ± 0.03 s/bit" would be more helpful than the statement "Our data show a correlation r = 0.9910".

Lessons learned:

- There are variations of Fitts' law formula.
- The interpretation of the a-constant in Fitts' law depends on the chosen formula.
- There is doubt about MacKenzie's formula.
- An agreement on one formula for Fitts' law is desirable.

3.4.3 The Debate on Fitts' Law for Eye Movements

The question whether Fitts' law applies for eye movements is a topic for discussions in the HCI community. Fitts himself did eye-tracking research [Fitts, Jones, Milton 1950] before he published his famous work, but for his law on the capacity of the motor system he experimented with hand movements – eye movements are not mentioned.

It is worth mentioning that Fitts argumentation was built on the fact that the subjects had the same performance with styluses of different weights, and for this reason the physical strength can not be the limiting factor (within a reasonable range of weights). A similar experiment for the eyes is hard to imagine because it would be difficult to fix weights on the eyes.

The first application of Fitts' law to eye movements was done 1987 by Ware and Mikaelian [Ware, Mikaelian 1987]. They evaluated the possibilities of eye gaze as a pointing device for computer input. To provide values comparable to other pointing devices they fitted their data to Fitts' law. The data fit quite well but this is not necessarily a proof as explained later in this section. Ware and Mikaelian did not claim that Fitts' law applies to eye movements. They wrote: "*This is a theoretical construct designed to account for eye-hand coordination. We use it here only as a convenient way of summarizing the results, not because we wish to make any theoretical claims.*" [Ware, Mikaelian 1987]

In 1999 Zhai et al. [Zhai, Morimoto, Ihde 1999] expressed themselves sceptical on the validity of Fitts' law for the eyes, but nevertheless they present values for the *b*-constant of Fitts' law for the combination of eye and mouse movement.

Miniotas did the next statement proving Fitts' law for eye movements in a CHI Student Poster [Miniotas 2000] in the year 2000.

In the same year Sibert and Jacob published a paper on eye gaze interaction [Sibert, Jacob 2000] in which they expressed themselves sceptical on that issue and referred to the work of Abrams, Meyer and Kornblum [Abrams, Meyer, Kornblum 1989].

In 2005 Zhang and MacKenzie measured the throughput of the eye as a pointing device according to the standards of the ISO9241-9 [Zhang, MacKenzie 2007], [MacKenzie@ 2007]. Using this method implicitly assumes the validity of Fitts' law for the eyes.

Also in 2005 Ashmore et al. referred to the confusion of Fitts' law for the eyes:

"*Eye pointing is, in principle, similar to manual pointing, although the debate regarding its predictability by Fitts' Law complicates this metaphor. If Fitts' Law were a suitable model, then any innovations developed for manual interaction could potentially transfer to eye pointing. Ware and Mikaelian [27] found that eye input does indeed follow Fitts' Law, although Zhai et al. [29] reported only low correlation to the model (r2 = .75). More recently, Miniotas [19] reported fairly high correlation (r2 = .98) with the following Fitts' variation, expressing mean selection time (MT) as: MT =a+blog2 (A/W + 0.5), with a = 298 (ms) and b = 176 (ms/bit).*" [Ashmore, Duchowski, Shoemaker 2005].

The papers referenced in this citation correspond to the references [Ware, Mikaelian 1987], [Zhai, Morimoto, Ihde 1999], [Miniotas 2000] of this document.

In 2008, Vertegaal did a Fitts' law comparison of eye tracking and manual input in the selection of visual targets:

"*Although manual pointing typically provides an excellent fit to the Fitts' Law model, there has been some debate as to whether the same would be true for eye input. Although they did not report a correlation value, Ware & Mikaelian [33] found that eye input does indeed follow Fitts' Law. Zhai [35] found correlations that were fairly low, in the order of r2=.75. The most likely cause for this was the presence of eye tracker noise in their experiment. Perhaps most interestingly, Sibert and Jacob reported little variance in movement time over distance [24]. Miniotas [17] reported the highest fit with an r2 = .98.*" [Vertegaal 2008]

The papers referenced in this citation correspond to the references [Ware, Mikaelian 1987], [Zhai, Morimoto, Ihde 1999], [Sibert, Jacob 2000], [Miniotas 2000] of this document.

The citations above give the impression that a correlation of 0.7 is a bad value. However, the squared correlation coefficient R^2 depends on the number of IDs used for the experiment. Actually, the correlation is only an indicator for a linear dependency. It needs a further test (with null hypothesis that the correlation is zero) to find out the significance of the calculated correlation. It is possible that a lower correlation yields a better significance if the correlation was calculated from more data pairs. The argumentation with correlation is common for Fitts' law in general and not specific to the question whether Fitts' law applies to eye movements. The citations above are the reason why a discussion of correlation is presented in this section.

The Correlation of Fitts' Law Data

Figure 30 plots the performance of one participant for a target acquisition task with a classical mouse against the index of difficulty. The data are from the user study presented in 4.4. The measurement was done with two different target sizes; 50 clicks for each target size and randomly chosen distances.

Figure 30: Performance of one participant for the click-the-target task with a classical mouse

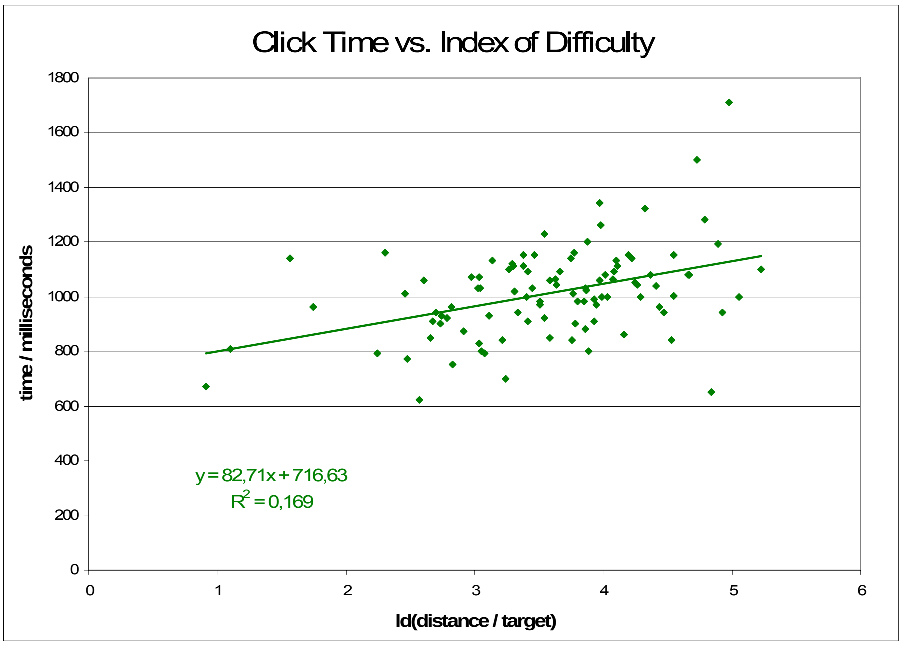
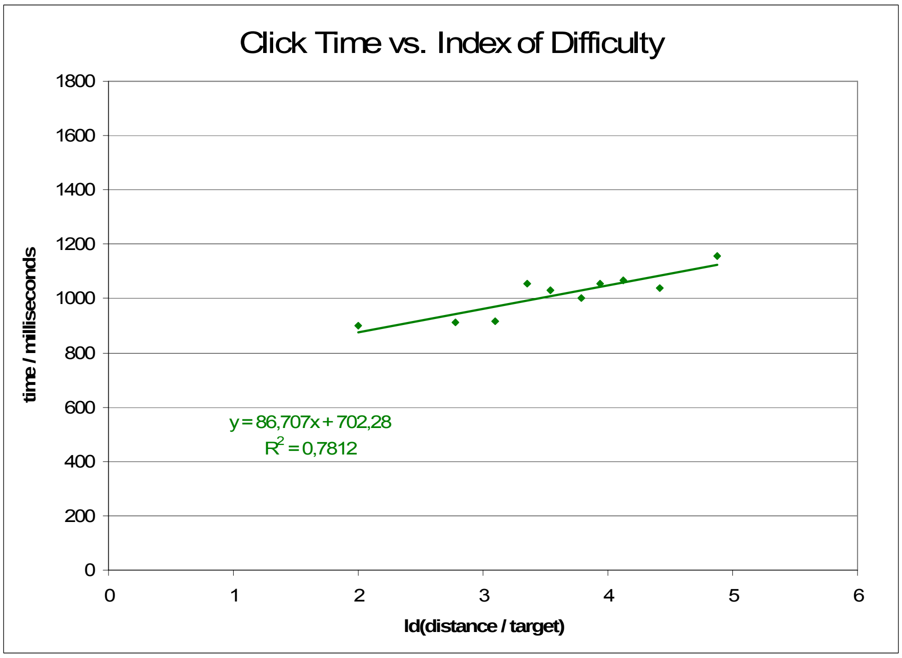


Figure 31: Performance of one participant for the click-the-target task with a classical mouse. One point represents an average of ten clicks.



In Figure 30 the *b*-constant, which is the slope of the trend line, calculates to 82 milliseconds/bit. However, the correlation coefficient R^2 = 0.169 is very low, while publications on Fitts' law typically report correlations above

0.9. This raises the question why the data presented here have such low correlation. The reason is that the data presented here use a continuous scale for the IDs.

Fitts used a discrete set of values for the index of difficulty. The reason for this lies in the mechanical setup of Fitts' experiments with metal stripes of discrete sizes. Experiments with a computer mouse allow arbitrary values for target sizes and distances and can provide data on a continuous scale. However, the publications on Fitts' law for pointing devices present also small discrete sets for the ID, typically four target sizes and four distances resulting in seven different IDs as in [MacKenzie, Sellen, Buxton 1991]. The reason seems to lie in better looking correlation values when first averaging times for the same ID and calculating the regression line for the mean times. To simulate the effect the data from Figure 30 are sorted by the ID and grouped in ten sets of ten data points. Figure 31 plots the mean time and the mean ID from every set. The value for the *b*-constant is 86 milliseconds/bit and nearly unchanged. However, the squared correlation coefficient R^2 now is 0.781. With fewer points the R^2 value is much better. This is not surprising as with two points only the regression line would fit perfectly. This means that the correlation alone, without stating the number of data pairs from which it was calculated, does not tell how good the data fit to Fitts' law.

Figure 32: Different data sets which yields the same correlation when averaged over IDs first.

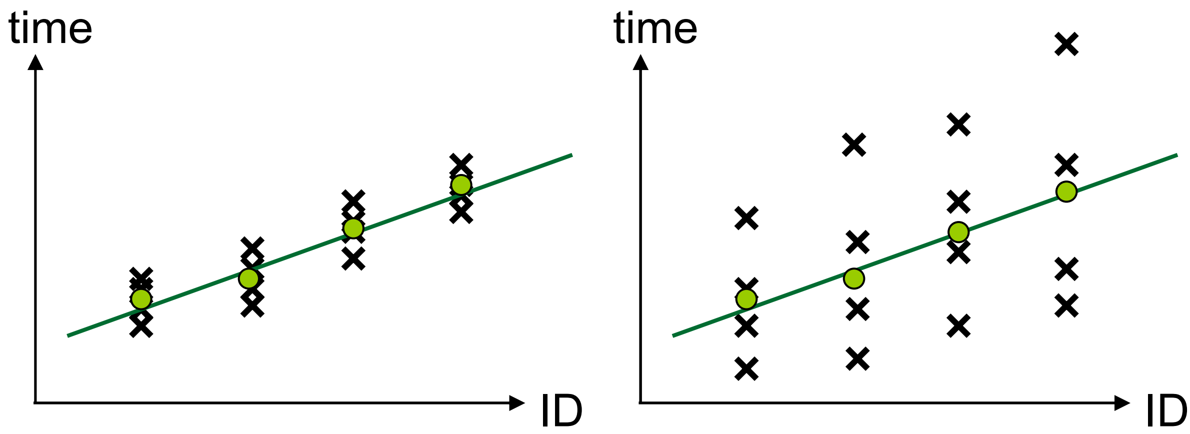


Figure 33: Correlation of data averaged over IDs with different number of data for each ID.

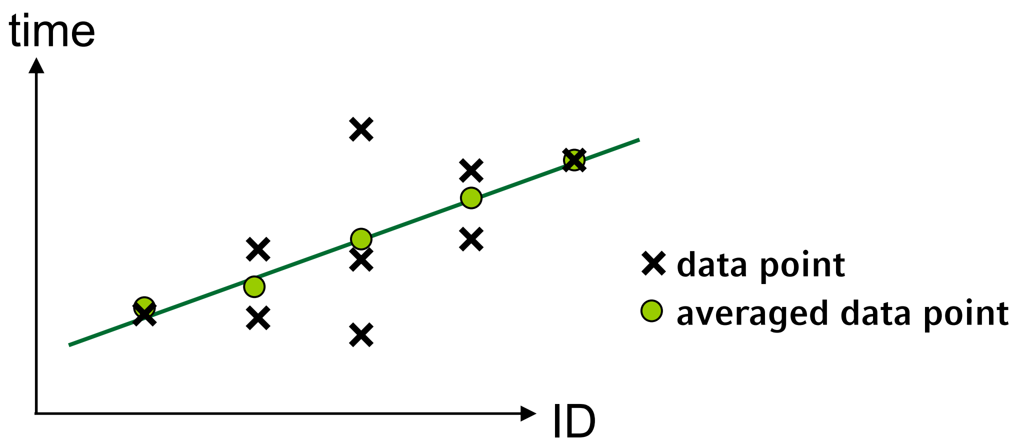


Figure 32 and Figure 33 give examples for what happens when the data are averaged first for each ID and the correlation is calculated afterwards. The two schematic data distributions in Figure 32 yield the same correlation value if calculated with this method. However, the first data set fits much better than the second does. The calculation with averaged data loses the information on the variance of the means.

Figure 33 shows a situation which is typical for Fitts' law experiments where the distances and target sizes are chosen in a way that different combinations of target size and distance result in the same ID. In an experimental condition where such distances and target sizes are fully crossed with one another, the number of recorded data is not equal for the IDs (see also Table 5 below). In consequence, the averages for an ID do not have the same weight and beside an irrelevant value for the correlation this causes also wrong values for the *a*- and *b*-constant.

The Evaluation of Fitts' Law for the Eyes

All the publications mentioned above which state that Fitts' law is valid for eye movements have methodological deficits in several aspects.

The first aspect is a missing discussion of the meaning of target size to the eye. Figure 24 shows the time of saccadic eye movements while watching a video. This raises the question what are targets for the eye when watching a video and what are the sizes of these targets. Does a target need a minimum contrast to the surrounding or does it need a special contour? Estimating a target size involves object recognition. However, the eye has only a small region in the visual field with a high resolution. This means that the eye may have problems to recognize a target outside of this region. The saccadic movement to the target takes place for the purpose of object recognition and this means that the eye knows the exact target size after the movement.

There is a lower cut-off for the target size caused by the eye's inaccuracy. In terms of speed-accuracy trade-off, the precision of the movement will not become better than the accuracy of the eye by spending more time. It is also questionable whether it makes sense to offer the eye a bigger target. Does the eye recognize a big target or does it select just a detail of the target for the positioning task?

Before assuming that Fitts' law applies to eye movement, there should be answers to the following questions:

- What is a target for the eye?
- How does the eye estimate the target's size?
- When does the estimation of target size take place – before or after the saccadic movement?

The second aspect is an implicit assumption of the validity of Fitts' law when evaluating the data with IDs (index of difficulty). Averaging the data for the same ID forces the outcome that Fitts' law is valid even if it is not the case.

Let us assume that Fitts' law is not valid and that the formula of Carpenter (see 3.4.1) is correct. This means the positioning speed for the eye is independent of the target size and the time to hit the target has a linear dependency on the distance. Even then a typical Fitts' law evaluation will produce a nearly perfect regression factor $R^2 = 0.97$ as demonstrated below. This means that a good correlation does not prove Fitts' law.

Typical Fitts' law user studies use four target sizes with the ratios 1, 2, 4 and 8 and four distances with the same ratios. This results in seven different ratios of distance and target size as shown in Table 5.

Table 5: All combinations for four distances and four target widths

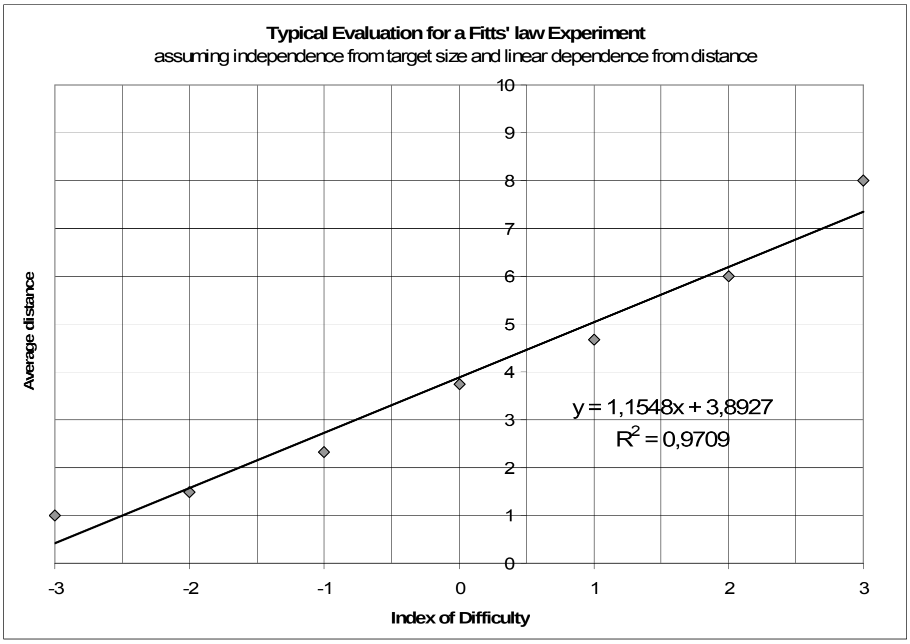
d/w	1	2	4	8
1	1	2	4	8
2	1/2	1	2	4
4	1/4	1/2	1	2
8	1/8	1/4	1/2	1

Typically, the subjects in such a user study perform positioning tasks over all combinations of target sizes and distances, which results in data probes of uniform distribution over all combinations. Now assume a linear relation for execution time and distance (without dependency on the target size). With a linear relation it is possible to calculate the average execution time from the average distance. Table 6 shows all possible IDs and the resulting average distances.

Table 6: All possible IDs and the corresponding average distance

d/w	$\log_2(d/w)$	#	$\sum$ d	$\sum$ d / #
1/8	-3	1	1	1
1/4	-2	2	1+2	3/2
1/2	-1	3	1+2+4	7/3
1	0	4	1+2+4+8	15/4
2	1	3	2+4+8	14/3
4	2	2	4+8	6
8	3	1	8	8

Figure 34: Averaging the data for the ID forces the validity of Fitts' law



The correlation does not change when applying a linear transformation to the data. This means the correlation does not change when using another unit for the axis or adding an offset to the data. As the formula of Carpenter is a linear transformation, this means that the correlation for the time over ID is the same as for distance over ID. Figure 34 shows the trend line for the average distance over ID as given in Table 6. The regression factor R^2 = 0.97 is very close to 1.0 although the assumption for the calculation was that Fitts' law is not valid.

As there are good reasons that the positioning speed of the eye does not depend on the target size and the formula of Carpenter is a good approximation, it seems that exactly the effect explained above happened to the research of Miniotas. Miniotas used only two target sizes and four distances:

"Controlled variables were target amplitude (A = 26, 52, 104, and 208 mm) and target width (W = 13 and 26 mm) fully crossed with one another … Each A-W combination initiated a block of 8 trials." [Miniotas 2000]

With four distances but only two target sizes the calculations from above lead to a regression factor $R^2 = 0.945$. Using the Fitts' law variation $MT = a + b \cdot \log_2(A/W + 0.5)$ as used by Miniotas leads to $R^2 = 0.964$ (R = 0,982). However, Minotas presented a nearly perfect value R = 0.9910 for the evaluation of his data.

Zhai et al. [Zhai, Morimoto, Ihde 1999] used two target sizes (20 and 60 pixels) in combination with three distances (200, 500 and 800 pixels). Zhang and MacKenzie [Zhang, MacKenzie 2007] used two target sizes (75 and 100 pixels) in combination with two distances (275 and 350 pixels). In both cases, the ratios of target size and distance do not fall together and the effect demonstrated above does not apply directly. However, in both cases the small IDs correspond to small distances and big IDs to long distances and this causes a positive slope even if the law of Carpenter is valid. The good regression factor is a consequence of the small number of IDs as demonstrated above.

The third aspect is a missing comparison with existing competitive models, i.e. the model of Carpenter or Abrams et al. It may be possible that data fit quite well to one theoretical model but the data may also fit well to another theoretical model. This means that the formula produces good values but it does not prove the theoretical model.

Lessons learned:

- There is a controversial discussion on Fitts' law for eye movements in the HCI community.
- It is not clear what is a target for the eye and when and how the eye esimates the target size.
- The correlation coefficient R^2 alone without the number of data pairs (ID, t) for which it was calculated does not tell how good the data fit to Fitts' law.
- Building the same IDs from different target sizes is an implicit assumption of Fitts' law. It leads to a confirmation of Fitts' law even if it is not valid.
- Assuming Fitts' law for eye movements is a contradiction to eye speed models of psychology.

3.4.4 The Screen Key Experiment

The user studies conducted for this thesis provide large data sets on eye movements. With two competitive models for eye movements, Fitts' law and the model of increasing acceleration, it is a good idea to have a look at these data to see whether they favour one of these models.

Chapter 4 researches eye gaze as a pointing device and describes a user study similar to the experiments of Mikaelian and Ware [Ware, Mikaelian 1987]. Mikaelian and Ware introduced the term screen key for an interaction method where the gaze is activated by looking at a target to select the input command and looking at a second target, called the screen key, to activate that command. Looking at the first target brings the gaze to an initial position. The second target has a known distance to the initial position and a known size. This makes it possible to interpret the data of the experiment with the assumption of Fitts' law. The data set consists of 50 completion times of a single subject. The experiment was not designed for proving Fitts' law and was done with a single target size of 100 pixels. Consequently, the data cannot tell anything about the dependency on target size but they could tell something about the functional relation of distance and performance time.

Figure 35 presents the data in a plot with a logarithmic scale for the ratio of distance (centre to centre) to target size (index of difficulty) and a linear scale for the performance time. Such plots are used for the evaluation of Fitts' law where the data should lie on a straight line. The slope of the line in seconds per bit is the informational part of Fitts' law (the *b*-constant). The data in Figure 35 are more a cloud than a straight line but this does not indicate a violation of Fitts' law, as the data for the classical mouse look similar.

Figure 36 presents the same data in a plot where both scales are logarithmic. Such plotting brings the data in a straight line if they obey a potential law. The slope of the line is the value of the exponent in the potential law. For the model of Abrams et al. this slope should have the value 1/3. The regression line has a slope of 0.23 but a slope of 0.33 according to the model of Abrams, Meyer, and Kornblum (dashed line) would not violate the data. From the presented data it is not possible to decide which model fits better.

Figure 35: Time to move the gaze from one target to another target as log-lin plot.

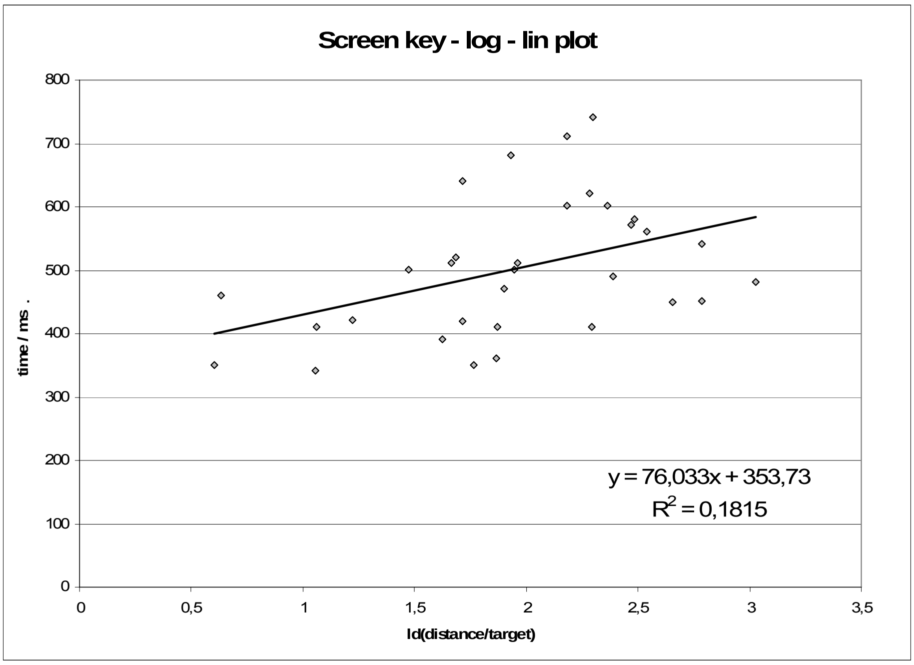
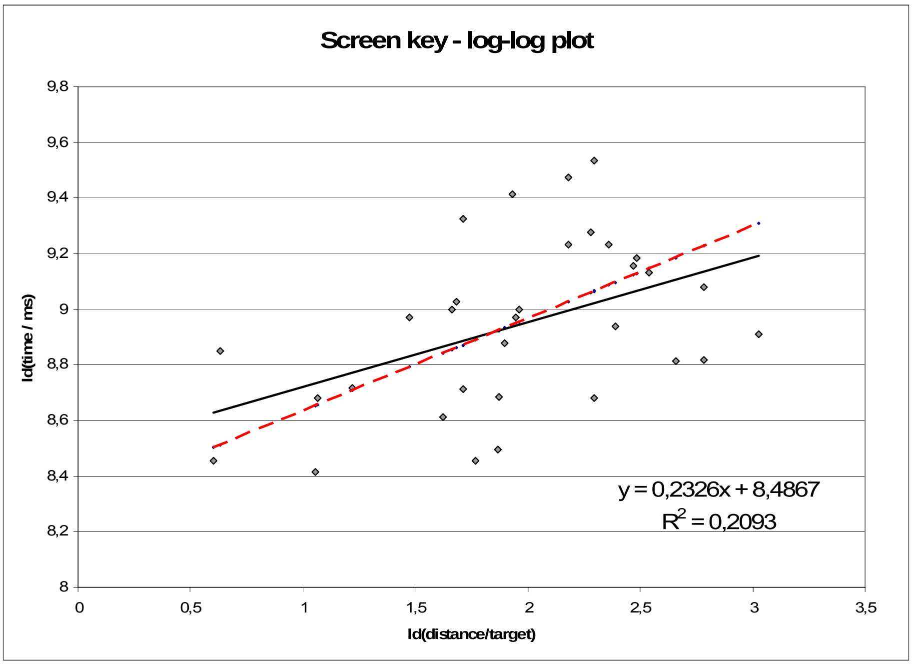


Figure 36: Time to move the gaze from one target to another target as log-log plot. The dashed line has the slope 1/3 according to the model of Abrams, Meyer, and Kornblum.



Lessons learned:

- The data for the eye as pointing device are as noisy as the data for a mouse as pointing device.

3.4.5 The Circle Experiment

The existing data could not clarify whether Fitts' law holds for the eyes. This was the reason to set up another experiment. The paper of Sibert and Jacob [Sibert, Jacob 2000] describes an experiment, which the authors called the circle experiment. The experiment presents circles on the screen and the test subject has to look at them. Their experiment used a single circle size. The authors suggested repeating the experiment with different circle sizes to measure Fitts' law. The setup for the experiment presented here was mostly the same as in [Sibert, Jacob 2000] with the difference that no dwell time (zero dwell time) was used. The user study was stopped after measuring five participants. One reason to stop the user study was that some participants gave up when the target size was below the eye's accuracy. Typically, when the participants realized that they were not able to hit the target they started to shake the head. With a liberal head fixation by a chin rest only, this spoiled the calibration and worsened the problem. Another reason was that the data did not allow an evaluation for Fitts' law as explained later. Nevertheless, some of the data are presented here because they give some insights on the topic. Figure 37 shows the gaze paths of one participant recorded during the circle experiment.

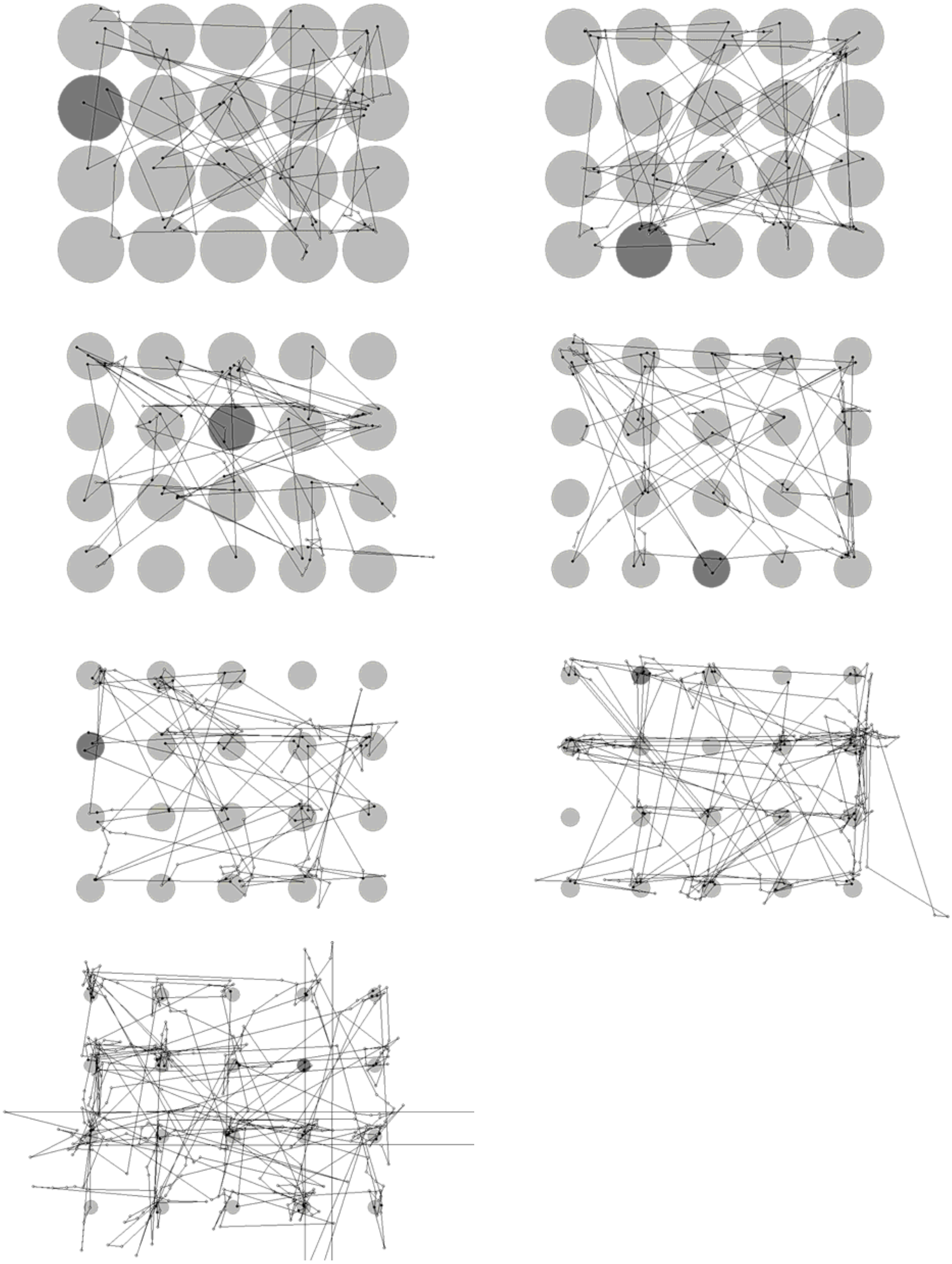
When thinking about a positioning task for the eye the first question is whether the eye hits the target with a single saccade or in a sequence of saccades. A sequence of saccades would lead to a scenario as used for the derivation of Fitts' law (see 3.4.2) but with the difference that the steps are not constant in time. The time for a saccade depends on the length (see 3.3.3) and this would lead to a different formula.

Figure 37 shows that the gaze moves to a target with a single saccade as long as the target size is bigger than the eye's accuracy. Small saccades happen mostly within the target and not very often in between the targets. Consequently, the question is whether the amplitude-time relation of a saccade obeys Fitts' law.

Figure 37 also shows that the gaze does not move to the centre of the targets – it moves to the edge of the targets. This means that it is questionable whether it makes sense to offer the eye a big target – the eye searches its own target size area. It seems that the concept of target size for pointing tasks with the eye has problems. At the beginning of the pointing task the target lies in the area of peripheral vision where the resolution is low. This brings up the question what the eye will recognize as a target and how it estimates its size. It is questionable whether the eye tries to position its view to the centre of a target. It seems more likely that the eye tries to position its view to a spot and the demand for the positioning task is that this spot finally will be within the size of the fovea. Consequently, the target size for the positioning task is the size of the fovea. It also means that the speed of the positioning task is independent of the target size. The evaluations in 3.4.6 are built on this understanding.

The fact that the gaze does not move to the target centre also caused the problem of evaluating the data. The performance time for the first three target sizes was nearly equal for all participants, however, with a small tendency to higher execution time for smaller targets. The grid size for the targets stayed constant and consequently the distance of the target centres stayed constant too for different target sizes. The small increase in performance time for smaller targets could be used as an argument for a positioning time dependent on the target size. However as the gaze moved from edge to edge the total distance covered by the gaze increases with smaller targets as the gaps between the targets become bigger. In this situation the result of the evaluation depends on the definition of the distance – task defined distance or performed distance. Changing the definitions after looking at the data is not a scientific approach.

Figure 37: Gaze paths (starting at the dark target) for seven different target sizes (radius of 70, 60, 50, 40, 30, 20, and 15 pixels) in the circle experiment. The last three sizes are below the accuracy limit.

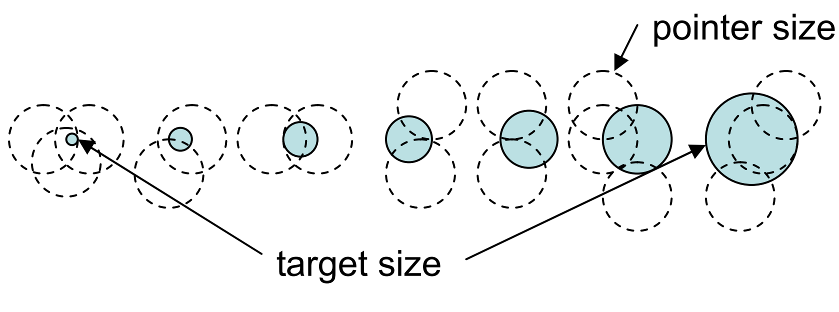


Although the presented experiment does not give a proof, it indicates that the performance time for a positioning task for the eye does not depend on the target size. This is in accordance to the opinion of psychologists. A personal email to Professor Deubel [Deubel@], asking whether Fitts' law applies to eye movements, brought the answer that the time for a saccade depends only on its amplitude. This answer did not encourage setting up a redesigned user study to find out something already known.

As shown in 3.3.2, the ERICA eye tracker does not report eye movements below the 16-pixel threshold and saccades with lengths up to 28 pixels are not reported correctly. The property of the eye tracker to damp the noise disallows fine positioning with the eye, even if the eye could do it. If the eye tries to get closer to the target, the eye tracker first does not report any movement and finally reports a movement to the other side of the target; the reported gaze position jumps across small targets. Therefore, the problems hitting the small targets are mainly an effect of the limited eye tracker's accuracy. The last three sizes in the circle experiment seem to be affected by the eye tracker's inaccuracy. In the case of missing the target, the participant's eye moved away from the target and tried to hit the target again. After some attempts the eyes hit the target by chance. The situation is similar to throwing a stone at a small target and this picture describes the situation (for this type of eye tracker) in the circle experiment for small targets below the accuracy quite well. Of course, it is possible to give a relation for the expected time for a successful hit depending on the target size and distance. However, the character of such a stochastic process is very different from a feedback-controlled process.

The existence of an accuracy limit creates a situation which differs from the experiments conducted by Fitts. Fitts used a stylus as a pointing device with a nearly perfect tip. It is not possible to do the same for the eye. It does not make sense to do positioning tasks for targets smaller than the accuracy of the pointer. When doing so the effective width of the target is the accuracy of the pointer. A pointing task with the fingertip has an accuracy of about ±0.5 cm because of the size of the fingertip. Pressing buttons with sizes of 0.1 cm, 0.2 cm, 0.4 cm, and 0.8 cm with the finger will not reflect target sizes but the size of the fingertip plus the size of the target. See Figure 38 for an illustration.

Figure 38: Target sizes and accuracy for pointing tasks for a pointer of a fixed size.



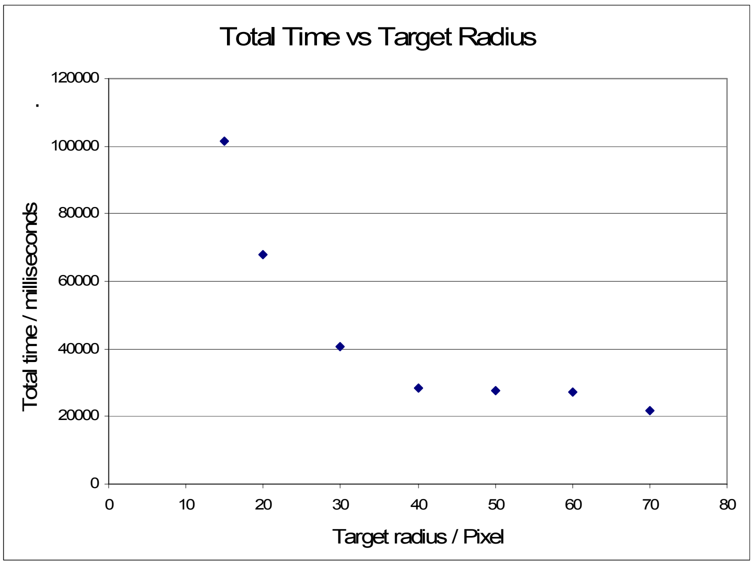
The difference of the situation in a Fitts' law experiment with the eye and a fingertip lies in the fact that the eye tracker reports only the coordinates of a single pixel and the normal hit condition is that this position is inside the target. When changing the hit condition, so that a hit is triggered when the gaze is not further away from the target than the accuracy, the situation for the fingertip and the eye will be the same. For gaze-aware interfaces this means that it is possible to provide targets smaller than the accuracy if the hit condition takes the accuracy into account. Then, the distance between the targets must be bigger than the accuracy.

Although the data were not evaluated, they are presented here for completeness of the discussion. Another reason is to provide an explanation for the size dependency measured by Ware and Mikaelian [Ware, Mikaelian 1987, Figure 3]. Table 7 shows the completion times for the circle experiment. Figure 39 presents the plot of the data from the participant (P1) whose gaze path is depicted in Figure 37. The figure shows a dependency on target size. However, the discussion above makes clear that the size dependency for the small targets is an effect of the eye tracker's accuracy. The small size dependency for the big targets is an effect of distance definition.

Table 7: Total time of one participant to perform the circle experiment

Participant	P1	P2	P3	P4	P5
Radius / pixel	Time / ms	Time / ms	Time / ms	Time / ms	Time / ms
70	21701	19488	22853	23314	19638
60	27339	21161	23023	23875	20349
50	27760	20990	24416	30424	21912
40	28321	26648	29052	61499	29703
30	40658	33128	30053	60988	34510
20	67778	98471	41921	104841	55590
15	101606	183904	86264	210303	73997

Figure 39: Total time to perform the circle experiment in dependency of target radius.



Lessons learned:

- The eye can hit a target bigger than 1 - 2° in size with a single saccade.
- The size of the target does not matter, as the eye does not aim for the centre of the target but the edge. The time to hit depends only on the distance to the target (edge).
- There are problems to hit targets smaller than 1 - 2° in size. The problem is a combination of the eye's and the eye tracker's inaccuracy.

3.4.6 Ballistic or Feedback-Controlled Saccades

The question how a saccade selects a target and how this relates to object recognition (see [Deubel, Schneider 1996] as a pointer into literature) is not trivial. Without a model for the object recognition, it is difficult to define the size of a target and without a target size, it is not possible to assign an ID. The presented data from watching a video raises the question what are the targets and what are their sizes. Without the concept of size, the eye just chooses a spot for the purpose of object recognition and positions the eye with a saccadic movement so that the spot gets into the area of the fovea. In this context, the idea to apply Fitts' law to eye movements analogous to mouse movements seems to be naïve. Nevertheless, the questions whether saccadic movement is ballistic or feedback controlled seems to be open. Duchowski writes:

"*… during saccade execution, there is insufficient time for visual feedback to guide the eye to its final position (Carpenter, 1977). One the other hand, a saccadic feedback system is plausible if it is assumed that instead of visual feedback, an internal copy of head, eye, and target position is used to guide the eyes during a saccade (Laurutis & Robinson, 1986; Fuchs, Kaneko, & Scudde, 1985). Due to their fast velocities, saccades may only appear to be ballistic (Zee, Optican, Cook, Robinson, & Engel, 1976).*" [Duchowski 2002, p. 45-46]

Following the idea of feedback-controlled saccades where the performance is limited by the informational capacity of the controlling process leads to Fitts' law with a constant target size (size of fovea). The alternative is the model of Abrams et al. based on the increasing acceleration by the eye muscles, which means the saccadic movement is ballistic. The following evaluation compares both approaches.

Figure 40: Saccade times over their length, linear plot with three manually fitted curves corresponding to the three models introduced

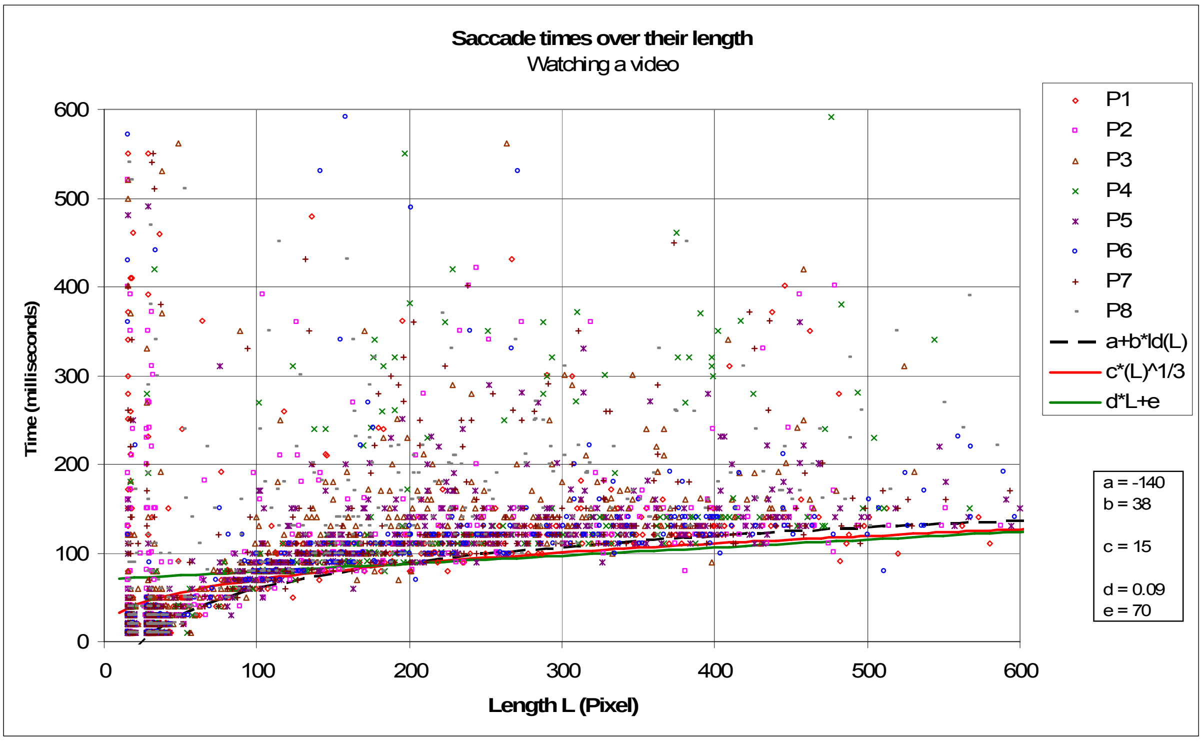


Figure 40 shows a plot of saccade times over their length. The data, already presented in 3.3.3, are from the third task of the user study presented in 5.4.3 where eight people watched a video. The plot shows varying times for a saccade of a certain length. There are several reasons for that: different individuals need different times, the eye can move slow or fast for one individual and the eye tracker can report different times because of previous or succeeding saccades below the threshold (see 3.3.2). The minimum of the times, representing the maximal speed of the eye, form a curve. The question is whether this curve represents Fitts' law or the model of increasing acceleration. The functional relation for Fitts' law (assuming fixed target size) is $a + b \cdot \log_2(L)$ and for increasing acceleration it is $c \cdot L^{1/3}$, with L is the distance to the target. Figure 40 shows a manually fitted curve for both cases and for the linear model of Carpenter.

In a logarithmic plot Fitts' law will result in a straight line while a double logarithmic plot will result in a straight line for a potential law with a slope of the exponent. Figure 41 and Figure 42 show the corresponding plots. It is hard to decide which one fits better.

Both plots show the two gaps on the length scale as reported already in 3.3.2 and mark the length below which the data cannot be trusted. Figure 42 reveals the 10-millisecond time slice of the operating system and show that the data for low times also should not be trusted. For the lengths above 80 pixels, both models fit well and it would be the lower lengths from 40 to 80 pixels which could make the difference. For lengths from 40 to 80 pixels, the corresponding times are around 50 milliseconds. With a frame rate of 60 Hz of the eye tracker camera the time inaccuracy is 17 milliseconds and the 10-millisecond time slice of the operating system adds additional inaccuracy. Concluding something from the narrow interval from 40 to 80 pixels where the accuracy is only 50% (27 milliseconds inaccuracy for 50 milliseconds values) seems to be risky. It should also be considered that the assumption of constantly increasing acceleration is a simplified model that could be elaborated. Consequently, it is not possible to decide from the data which model fits better.

Lessons learned:

- The presented data are not good enough to decide whether the saccadic movement is ballistic or feedback controlled. It needs a better eye tracker (better spatial and time resolution) to measure better data.
- The differences in time values predicted by the competitive models are small. For practical purposes in the field of HCI it does not matter which model is correct.

Figure 41: Saccade times over their length, log-lin plot

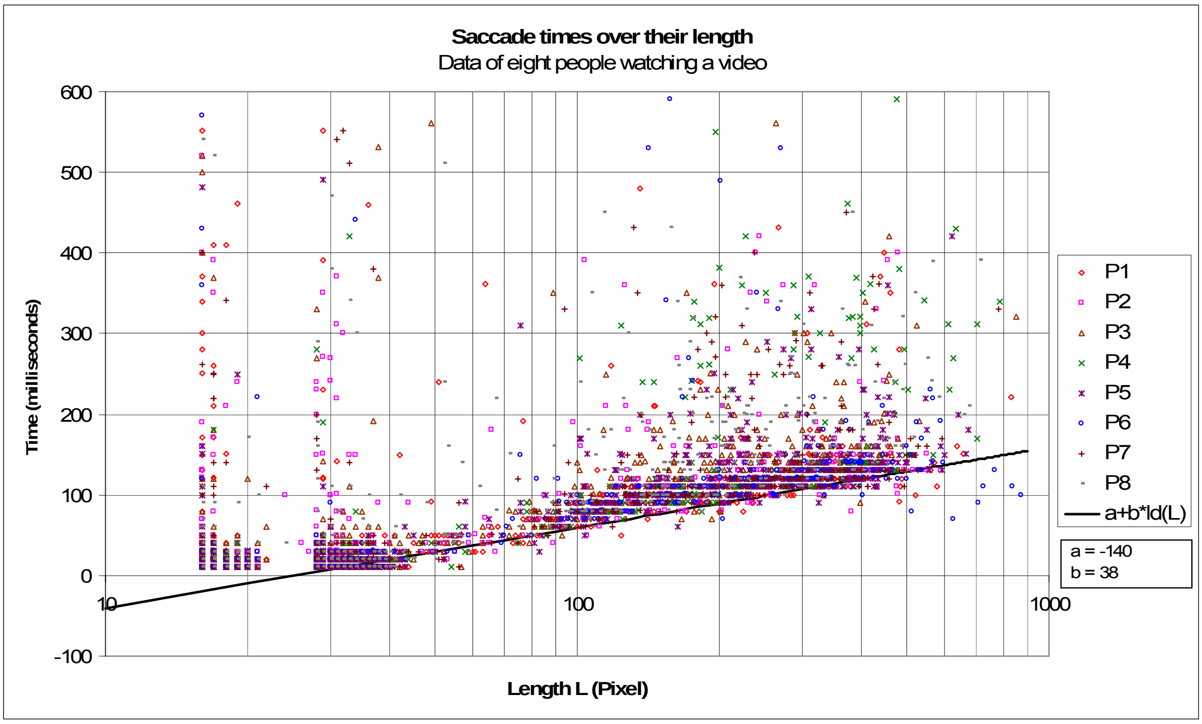
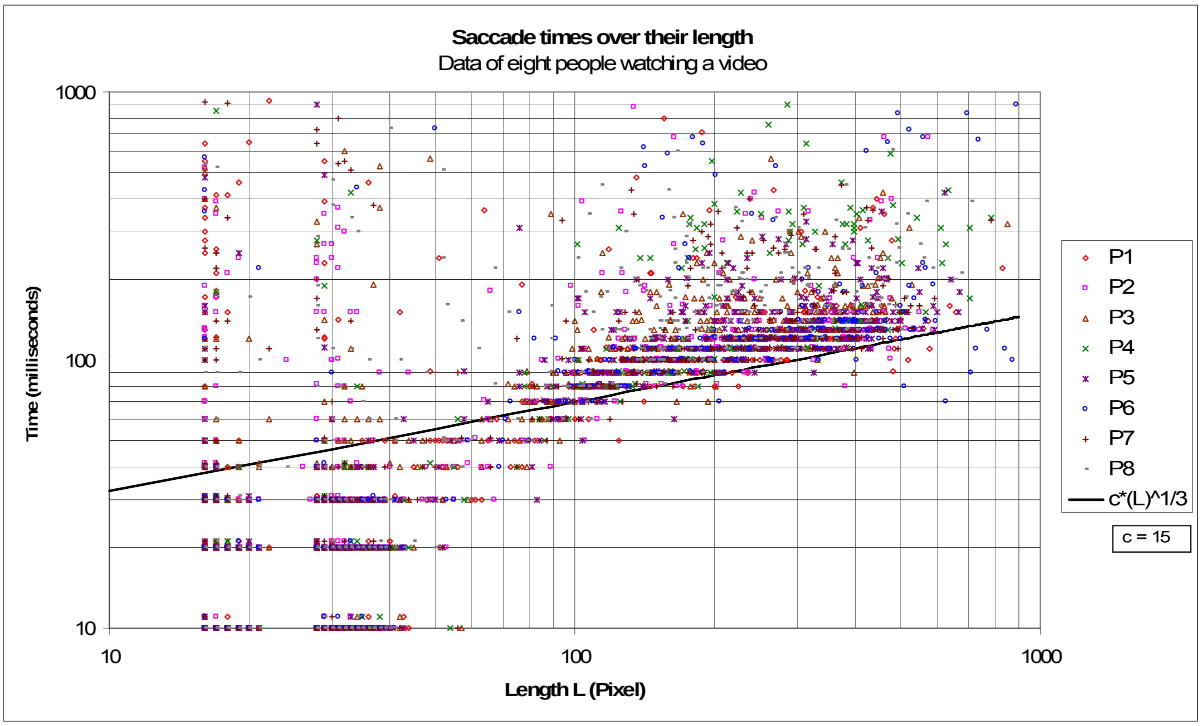


Figure 42: Saccade times over their length, log-log plot



3.4.7 Conclusions on Fitts' Law for the Eyes

With the presented data it is not possible to decide whether the maximal positioning speed of the eye is limited by the physical strength of the eye muscles or by the informational capacity of a controlling nervous system. However, there is no reason to assume that the eyes' positioning speed depends on target size. It seems that the eyes do not use a concept of target size, but only know spots of interest. Consequently, Fitts' law with a concept of target size does not apply to eye movements. In consequence, we cannot transfer the results for pointing devices to the eyes in general, especially not the consequences of Fitts' law for mouse devices.

Assuming the validity of Fitts' law for eye movements means to ignore the standard knowledge of psychology. None of the authors who applied Fitts' law to eye movements showed that there is a dependency on target size, which is not an effect of distance definition or accuracy. Additionally none of them mentions the results of Carpenter and discusses the contradiction to his research. The evaluation and statistical methods used to prove Fitts' law are also questionable.

Although Fitts' law does not apply to eye movements, it is possible to discuss the consequences for gaze-aware user interfaces in terms of size and distance of interaction objects in the same way as for a mouse. Fitts' law says that it takes more time to hit a target if the target is further away. It also says that it takes more time to hit a small target than to hit a big target. Both statements are not surprising and could be used without Fitts' results just by common sense. From the first statement it is possible to create a design rule: "Try to arrange the interaction elements for subsequent actions close to each other. Do not spread them over the space." From the second statement follows the design rule: "Try to make your interaction elements big. Do not make them too small." From the first design rule it is clear that the interaction elements should be close to each other and the limiting factor for the distance is the size of the interaction elements. When making the sizes of the interaction elements smaller the performance gets better because of smaller distances but also worse because of smaller sizes. Therefore, we need a balance of target size and target distance. It takes a small user study to find this balance for a practical situation. This means we can go quite far with only common sense and without any assumption on how the nervous system for motion control works. Actually, the argumentation given above goes far beyond Fitts' law, as it is still valid when Fitts' law is not applicable anymore, as it is the case for the eyes.

For the eyes, as long as the target size is above the accuracy, the positioning speed is independent of the target size. As soon as the target size is below the accuracy limit, it is very difficult to hit the target. This means that the time for positioning diverges towards infinity. Consequently, the interaction elements must have a minimum size. On the other hand it does not make sense to design big interaction elements for fast performance. This means that targets with sizes clearly above the accuracy limit are a good choice for interaction by gaze.

The positioning speed of the eye depends on the distance to the target and this means that for fast interaction the interaction elements should be close to each other. However once the distance of the targets is above 5° the time to hit the target hardly changes (see Figure 24). The dependency on the distance follows a curve, which is quite flat at the end, and within plausible ranges it does not matter whether this curve is logarithmic or a cubic root. As the corresponding curve for a mouse does also depend on the target size, it is not possible to state that the curve for the eye is more flat. However, the user studies presented in chapter 4.4 show that the eyes are much earlier on the target than the mouse. The proximity of interaction elements for gaze interfaces is not as important as it is for mouse devices.

For most purposes of HCI it is enough to know that the eye gaze jumps (saccade) to a spot of interest and waits (fixation) for the result of the object recognition. The time to perform the jump depends only on the distance and the object recognition may take longer than the time to perform the saccade.

Lessons learned:

- Assuming the validity of Fitts' law for eye movements means to ignore the standard knowledge of psychology. The eye does not 'think' in targets with size but in spots of interest.
- An accuracy-speed trade-off in the sense of Fitts' law does not exist, as there is a limit for the accuracy. Spending more time for positioning will not achieve better accuracy. Interaction objects for gaze-aware interfaces must be bigger than the accuracy of eye gaze.
- Target sizes much bigger than the accuracy of eye gaze do not bring any speed benefit.
- The distances of interaction objects for gaze-aware interfaces do not contribute much to performance time, as the eyes are quick.
- The exact functional relation for time and length of a saccade does not matter, as there is a high variability in the data. Knowing the range of the variability seems to be of more importance than knowing the exact functional relation. Empirical values are sufficient for most practical purposes in HCI.

4 Eye Gaze as Pointing Device

Pointing is a very natural way to address an object. The introduction of GUIs (graphical user interfaces) using pointing for interaction was a big step forward in the usability of computers and its use by the masses. Pointing at an object you can see is much easier than typing in its name. Most GUI interactions, such as setting the text cursor, choosing a menu item, selecting a piece of text, following a hyperlink, and dragging an object involve pointing. In traditional systems these pointing operations are usually performed using a mouse, a trackball, a joystick, a track point, a touchpad or a touch screen. The speed and accuracy of these input devices has been studied in depth over the last 25 years [MacKenzie, Soukoreff 2003].

It is worth mentioning that pointing normally is done in combination with a mouse click that means that pointing normally is combined with an extra modality. One exception is the tool tip mechanism (or balloon help) which uses pointing without any extra input. Tool tips appear only when the mouse pointer rests for a while on a target to avoid disturbing the user with unwanted tool tips during normal mouse activities. Eye tracking applications for the disabled, who have the pointing of the gaze as the only input modality, use the same mechanism to avoid the Midas Touch problem. In the context of gaze-aware applications this method has the name dwell time method.

4.1 Overview on Pointing Devices

Eye gaze for pointing is the most obvious approach using an eye tracker for interaction. Pointing with the gaze means to look at something and this is considered as an intuitive and easy task, which humans can perform with high speed. As most people use a GUI to work with the computer and pointing is the basic operation for this, it is obvious that eye gaze pointing is seen as a promising technique. For this reason most research on interaction using an eye tracker was done for eye gaze pointing.

4.1.1 Properties of Pointing Devices

Beside accuracy and speed, pointing devices differ in space demand, provision of feedback, possibility for multiple pointers and the way to click. Table 8 gives an overview on these properties.

Table 8: Properties of pointing devices

	Mouse	Trackball	Track point	Touchpad	Touch screen	Eye gaze
Speed	faster	fast	medium	fast	fast	Very fast
Accuracy	time	time	time	time	size of finger	size of fovea
Space demand	much	little	little	little	none	none
Feedback	yes	yes	yes	yes	no	no (see 4.1.3)
Method	indirect	indirect	indirect	indirect	direct	direct
Multiple pointing	2 hands	2 hands	2 hands	10 fingers	10 fingers	1 pair of eyes
Intrinsic click	no	no	no	yes (no)	yes	no

Speed and accuracy of pointing devices are related according to Fitts' law; achieving a higher accuracy for a pointing operation demands more time. In the case of a touch screen or eye gaze, anatomic sizes limit the accuracy, and the question of how big the pointing targets should be is answered by the size of the finger or fovea and not by Fitts' law.

A mouse needs some space for its movements on the table. As sufficient space often is not available when sitting in a train or plane, mobile devices typically use a track ball, track point, or touchpad. A touch screen does not need additional space but the precision of a finger is low because of the size of the fingertip, which hides the visual information. To achieve high precision in pointing on a touch screen people use a pencil with a thin tip. The situation for gaze pointing regarding space demand and accuracy is similar to pointing with the finger on a touch screen. However, the option to increase the precision with a pencil does not exist for eye gaze but the eye gaze does not hide the pointing target for that.

The provision of feedback by a mouse pointer is mandatory for pointing devices which work indirectly. For direct pointing on a touch screen feedback is not necessary. Gaze pointing is also a direct method and does not need feedback. The reason to desire a feedback pointer for eye gaze comes from a possible calibration shift but introducing a gaze pointer can be counter-productive. For a discussion, see 4.1.3 (feedback for eye pointing).

The use of multiple pointers is a topic of current research. There are many discussions on two-handed interaction and the use of all fingers for pointing. For the eyes it is clear that both eyes move synchronously and we are not able to point with both eyes independently. Multiple gaze pointers only make sense for multiple people.

Many GUI operations use pointing in combination with a click on the mouse key, which means that there is a need of an extra modality. This is not the case for the touch screen where the touch provides a position and a click event. The touch pad does not give the same possibility, as the indirect method does not allow touching a target directly. The touch happens before steering the feedback pointer to the target and consequently the touch event is not useful to trigger an action on the target. A touch pad can use an increased pressure as click event but commercial devices with touch pad normally provide an extra mouse key. From the traditional pointing devices the touch screen is most similar to gaze pointing. The big difference is that a finger can be lifted to move to another location while the gaze cannot do that. Consequently, the eye gaze cannot produce a click event like the finger.

4.1.2 Problems with Traditional Pointing Devices

The hands do most computer input and the permanent repetition of the same movements of the hands can cause physical problems. Many frequent computer users suffer from the carpal tunnel syndrome. With the trend to bigger displays and multiple monitor set-ups the distance covered by the mouse and the necessary movement of the hand will increase.

Additionally people have problems to find the position of the mouse cursor on the display. There are several solutions to this problem within commercial GUIs. One solution is to highlight the mouse cursor; another solution is to position the mouse cursor where the user expects it. The popular Windows GUI offers three options: (1) ‚Show location of pointer when I press the CTRL key' (2) ‚Automatically move pointer to the default button in a dialog box' (mouse options in the control panel) and (3) the API knows the DS_CENTERMOUSE flag for the creation of dialogs. Current research provides other solutions for the problem for example the use of head tracking to place the mouse pointer in a multiple monitor set-up [Ashdown, Oka, Sato 2005].

4.1.3 Problems with Eye Gaze as Pointing Device

Using the eye gaze for pointing is intuitive and the first idea when thinking about eye tracking for computer input. Nevertheless, there are differences to traditional pointing devices operated with the hand:

- Eye gaze has a low accuracy

A standard mouse device has a high accuracy. Most people are able to address a single pixel with a mouse on a display with standard resolution. The smallest interaction objects on GUIs are only three pixels wide. Typical examples for an interaction which requires a 3-pixel-accuracy is the positioning of a text cursor in between two small letters or to drag the frame of a window. Expressed in visual angle such interaction objects are .1° in size. In contrast, the interaction objects for eye gaze have to be 1° degree in size to allow accurate positioning. Consequently, a GUI for eye gaze pointing needs bigger targets than the targets used in the GUIs of today. The other way round, it means that the GUIs of today cannot be operated with eye gaze as the only pointing device.

- Eyes can not press buttons

The fact that the gaze cannot be lifted like a finger makes a big difference. To select an item in a GUI people point with the mouse at the item and click the mouse button. The clicking cannot be done with the eye. Blinking with the eye is not an option. Blinking occurs as natural movement and it is difficult to decide whether a short blinking occurred just to keep the eye wet and clean or whether it was an intended action. A longer lasting blink is time consuming and people do not feel comfortable with it. Additionally, a video-based eye tracker cannot deliver coordinates while the eye is closed.

One possibility is to use the eye for pointing and a key press for selection. This way of eye input is fast but it is problematic because of missing feedback. The need of an extra input modality also takes away some of the advantages of eye gaze interaction such as being contact free and hygienic and usable as remote control. Another possibility to press a button with the eyes is to stare at it for a certain time (dwell time). This solution is time consuming, typical dwell times are 500 to 1000 milliseconds, and eats up the speed benefit that could result from the quick movement of the eyes.

- Feedback for eye pointing is problematic

A pointer feedback for the eye, similar to a mouse pointer, does not work. When inspecting the content of the display an eye pointer would obscure the area of attention and during a pointing operation a moving eye pointer is distracting. In the case there is a little offset for the eye pointer (caused by calibration drift) the eye moves towards it which again causes the eye pointer to move. As a result the eye will chase the eye pointer over the display (see also the citation of [Jacob 1995] in 2.7). As shown in chapter 3.3.2 the ERICA eye tracker does not report saccades with lengths below 0.5° and, as long as the drift is smaller than this value, the chasing of the eye pointer does not occur. However, this is not a solution to the problem because a user experiences it as an alternately hanging and jumping eye pointer. When the eye pointer is very close to a small target, it first refuses to move into the target and finally it jumps across.

- Eyes are needed also for inspection

If the hand operating the mouse is needed for some other task, for example keyboard use, the hand can be taken away from the mouse. For the eye gaze the situation is very different. The main task of the eyes is to see and if the eyes are also used for directing the computer it is hard to decide whether the eyes looking at a button want to read the button text or want to invoke the associated action. Jacob identified this problem and called it Midas Touch problem [Jacob 1990].

4.2 Related Work for Eye Gaze as Pointing Device

Most research on eye gaze for computer input sees the gaze as pointing device as this is very intuitive. The big obstacle for eye gaze as pointing device is the low accuracy. For this reason the focus of much research is on how to deal with low accuracy.

Raw and Fine Positioning

Zhai et al. made a suggestion to handle low accuracy and named it MAGIC (Mouse And Gaze Input Cascaded) pointing [Zhai, Morimoto, Ihde 1999]. MAGIC pointing uses the gaze for raw positioning and a traditional mouse device for the fine positioning. The next chapter discusses the MAGIC pointing in detail, as an enhancement of MAGIC pointing is a part of this work.

Adding Intelligence

In 2000 Salvucci and Anderson presented their intelligent gaze-added interfaces [Salvucci, Anderson 2000]. They start with a system where the gaze tracker delivers x-y positions to a standard GUI that highlights the interaction object the user is looking at. A gaze key analogous to a mouse key offers the user the possibility to trigger the action. To solve the accuracy issue the system interprets the gaze input in an intelligent way – it maps the gaze points to the items which the user is likely attending. To find these items the system uses a probabilistic algorithm which determines the items by the location of the gaze, i.e. the items close to the reported gaze point, and the context of the task, i.e. the likeliness of a command after a previous command.

Expanding Targets

Another approach solving the accuracy problems is the use of expanding targets. Balakrishnan [MCGuffin, Balakrishnan 2002] and Zhai [Zhai, Conversy, Beaudouin-Lafon, Guiard 2003] researched the use of expanding targets for manual pointing and showed that this technique facilitates the pointing task. Miniotas, Špakov and MacKenzie applied this technique to eye gaze pointing [Miniotas, Špakov, MacKenzie 2004]. In their experiments the target expansion was not visually presented to the users but the interface responded to an expanded target area. They called this technique static expansion. In a second publication Miniotas and Špakov studied dynamically expanding targets [Špakov, Miniotas 2005] that means targets where the expansion is visible to the users. The research was done for menu targets and the results showed that the error rate for selecting a menu item reduces drastically for the price of an increased selection time.

In the same year Ashmore and Duchowski published the idea to use a fisheye lens to support eye pointing [Ashmore, Duchowski, Shoemaker 2005].

In 2007 Kumar et al. presented an eye-gaze interface called EyePoint [Kumar, Paepcke, Winograd 2007]. This interface technique also uses expansions of the interaction targets and it also uses a key as additionally needed input modality. When pressing down this key the screen area where the gaze looks at becomes enlarged. Within this enlarged screen area, the user selects the target with the gaze and the action is triggered in the moment the user releases the key.

Use of further input modalities

The inaccuracy of gaze pointing means that a pointing action has ambiguities on the target if the targets are close to each other. This gave Minotas et al. to the idea to specify the target with an additional speech command [Miniotas, Špakov, Tugoy, MackKenzie 2005]. They used targets with different colours and asked the users to speak out the colour of the target loudly. They showed that this method allows addressing targets subtending 0.85 degrees in size with 0.3-degree gaps between them. The method does not bring speed benefit and it is not clear whether the better pointing accuracy is worth the extra effort of speaking at least in the case of operating a standard GUI. However, the concept is interesting because it is close to human-human interaction. Normally we are aware where other persons look to but with an accuracy much lower than an eye tracker. When we say, “Please give me the green book” and look on a table we get the green book from the table and not the green book from the shelves. We assume that the other person knows where we are looking to and it is only necessary to specify the object within that scope.

4.3 MAGIC Pointing with a Touch-Sensitive Mouse Device

When watching people working with big screens or dual monitor setups it is easy to observe that they have sometimes problems to find the mouse pointer. In addition, the mouse pointer is very often far away from the target intended to click. In consequence, it is necessary to draw the mouse a long way across the screen. The idea presented here (published in [Drewes, Schmidt 2006] and [Drewes, Schmidt 2009]) was to build a touch-sensitive mouse and position the mouse pointer at the gaze position when touching the mouse.

There is a similar idea in the literature. In 1999 Zhai et al. [Zhai, Morimoto, Ihde 1999] published their work on MAGIC (Mouse And Gaze Input Cascaded) pointing. The basic idea of the MAGIC pointing is positioning the mouse cursor at the gaze position on the first mouse move after some time of inactivity. In contrast to the idea above Zhai et al. proposed the MAGIC pointing to solve the accuracy issue. With MAGIC pointing the gaze positions the mouse pointer close to the target and the mouse is used for fine positioning. In their research Zhai et al. found out that there is the problem of overshooting the target because hand and mouse are already in motion at the moment of positioning. They suggested a compensation method calculated from the distance and the initial motion vector.

Combining both ideas leads to the concept of MAGIC touch which enhances the MAGIC pointing by replacing the detection of the mouse movement with the detection of a touch. Touching the mouse places the mouse pointer at the position of the gaze. This approach has the advantage that it does not need compensation techniques as the mouse does not move in the moment of touch. Another benefit of the concept is that there is no timeout. There is no need to estimate a good value for the timeout and in the case that the user looked at a wrong position in the moment of gaze positioning, she or he can reposition immediately without waiting for the timeout.

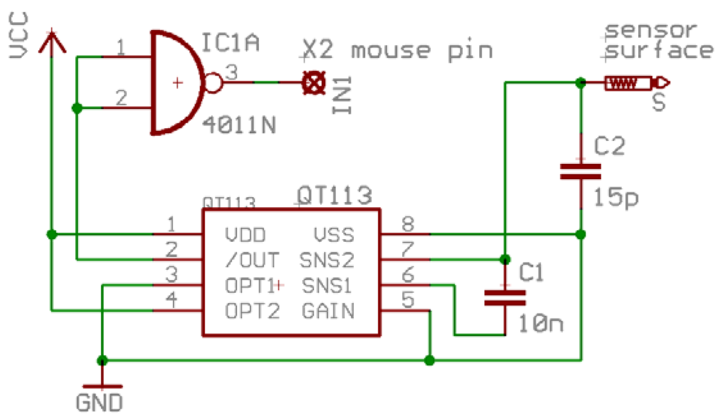
Figure 43 (left): The first version of the touch-sensitive mouse

Figure 44 (right): The second version of the touch-sensitive mouse

The work of Hinckley et al. [Hinckley, Sinclair 1999], where a touch-sensitive key is used to display and hide menus, inspired the initial design of the touch sensitive mouse built for the user studies. For the first prototype the entire mouse was made touch-sensitive (see Figure 43). This first idea turned out to be naïve. Observation during a pilot study showed that users tend to leave their hand on the mouse while spotting for the next target to click. In addition, the users looked at the mouse in the moment of grabbing it. To use the positioning by gaze with the first prototype the whole hand had to be lifted. For this reason the second prototype had only a touch-sensitive left mouse button detecting only the touch of the finger (see Figure 44).

The sensor for detecting touch uses capacitive sensing. The connection to the PC was first realized with an USB-I/O chip (IO-Warrior) that implements a HID (human interface device). While experimenting with the second prototype the touch event occasionally arrived after the click event. It turned out that, in the case the user trusts the system and does not wait for the feedback by a new pointer position, the click happens only 100 milliseconds after the touching. The detection time of 130 ms of the sensor used was too slow. Therefore, the chip was replaced with a faster version (QT 113) with a response time of 30 milliseconds. See [Qprox@] for a product description of capacitive sensor QT 113. Figure 45 depicts the schematic of the circuit. The additional electronics are small enough to fit into a mouse (see Figure 46). As the circuit is very sensitive, it was shielded with a thin layer of copper.

Figure 45 (left): Circuit of the touch-sensitive mouse device
Figure 46 (right): The touch-sensitive mouse with the electronics for the touch sensor



To further speed up the propagation of the touch event, the output of the touch sensor chip was connected to the X2 button of the mouse instead of polling the state of the sensor with the USB I/O-chip. This made it easy to program because the X2 button event is a standard event generated by the interrupt driven mouse driver. The disadvantage of this implementation was that the X2 button has an assigned meaning within the operating system and the modified mouse could only be used with the test software implemented for the experiment.

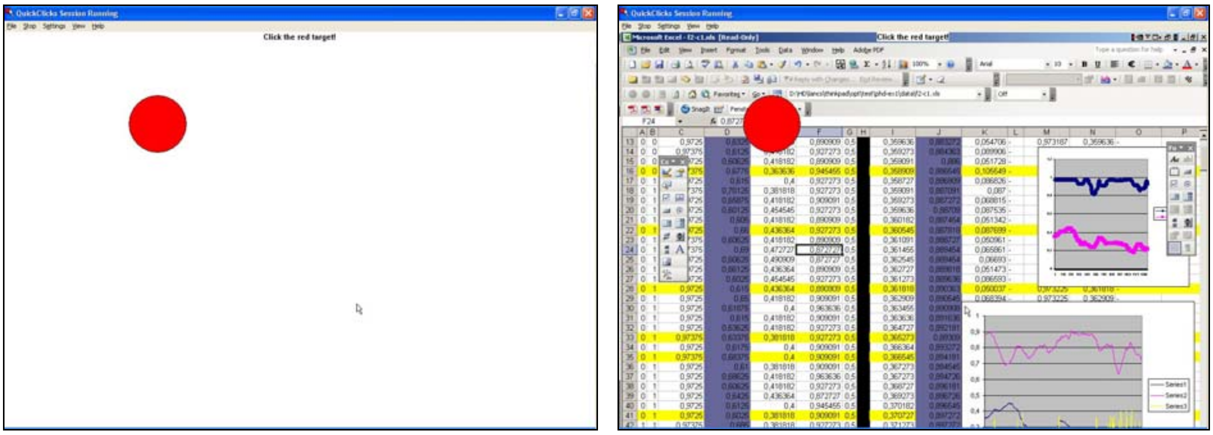
As only the rising edge of the touch signal triggers the X2 event and causes the positioning of the mouse cursor, the mouse behaves like an ordinary mouse while the finger stays on the sensor.

4.4 User Studies with the Touch-Sensitive Mouse Device

4.4.1 First User Study – Testing the Concept

The goal of the first user study was to compare the touch-sensitive mouse and gaze positioning against the classical pointing with a standard mouse device. The program written for the user study sets the target, a big red circle, and the mouse pointer at random positions and records the time until the first mouse click into the target. A pilot study did not show the speed benefit hoped for although subjectively it felt faster with the touch-sensitive mouse. The reason for the good performance of the classical mouse lies in the presentation on an empty background. At the beginning of the task there are only two changes on the display at the positions of the target and the mouse pointer. Both can be spotted with pre-attentive perception. To simulate the common situation, where the user is not aware about the position of the mouse pointer, the program for the user study was extended to provide a second mode – target and mouse pointer are displayed together with a background picture (see Figure 47). The pilot study now showed a speed benefit because the users started to stir the mouse to locate the mouse pointer position.

Figure 47: Displaying target and mouse pointer without and with background.



Ten participants in the age from 23 to 46 years, nine male and one female, all from European countries and regular computer users, participated in the user study. Every person had to do four runs with 50 clicks per tasks. The size of the target was 100 pixels (≈3°) in order to avoid accuracy problems and calibration shift. See 2.6.2 for the geometry of the setup.

Every user got a one-minute explanation and one minute to try the system. The first run was on white background using a classical mouse. In the second run the participants had to use the touch sensitive mouse. After this the users were asked with which input method they think to be faster and whether it is convenient not having to search the mouse pointer. The next two runs were a repetition of the first two runs, but this time with the background picture displayed. Again, the participants were asked both questions.

Normally, a user study should use a random order for the tasks to average possible learning effects and indeed the data show effects of learning. The reason for a fixed order lies in the observation made in the pilot study. Gaze positioning on a blank background felt faster than the classical mouse positioning although the measured times were equal. Because on complex background gaze positioning is faster the participants had to answer their subjective speed impression on a blank background first without bias from the complex background task.

Table 9 summarized the results of the study. The medians are chosen to eliminate outliers due to irritations of the participant during the task. A typical reason for outliers was confusion in the sequence of movements – lift the finger, look, touch the mouse key, click the mouse key. Calculations with the arithmetic means do not change the qualitative results. The mean of the distance to the target centre was about 400 pixels.

Table 9: Medians for total time in milliseconds of all participants

	Median for total times (in milliseconds)			
	blank background		with background	
Participant	mouse positioning	gaze positioning	mouse positioning	gaze positioning
P1	1097.0	751.0	1382.0	826.5
P2	971.5	1007.0	1066.0	806.0
P3	1121.0	716.0	1276.5	791.0
P4	1096.5	1066.5	1352.0	896.5
P5	932.0	871.0	1191.0	696.0
P6	926.5	1256.5	1066.0	1121.0
P7	1111.0	957.0	1217.0	891.0
P8	1211.5	1327.0	1412.0	1327.0
P9	1062.0	1482.0	1266.5	1277.0
P10	976.0	881.0	1101.0	656.0
Mean	**1051**	**1032**	**1233**	**929**
Std. Dev	94	253	128	234

Table 10: T-tests for four different task combinations (p-values)

t-test			
gaze vs. mouse pointing		with background vs. blank background	
blank background	with background	gaze positioning	mouse positioning
0.823686	**0.002036**	**0.020738**	**0.000014**

Table 10 shows the results of paired Student's t-test on the medians from Table 9 for four different combinations, which means the probabilities that the compared data sets are from a distribution with the same mean. There is strong significance that gaze positioning is faster than classical mouse positioning when using a background i.e. in the case the user is not aware of the mouse pointer position. The values also show that classical mouse positioning takes longer on a complex background. The effect of (significant) better performance for gaze positioning on a background compared to no background is most probably an effect of learning, as the order of the runs was not randomized. For a strict statistical evaluation it is necessary to show that the data obey the preconditions (Gaussian distribution) for the validity of the t-test. However, this is not worth the effort as the user study was designed with a poor understanding of the situation. The analysis, presented in 4.5, will give a deeper insight which does not need statistical values as a proof and does not even need a user study. Nevertheless, this user study was an important step on the way of understanding.

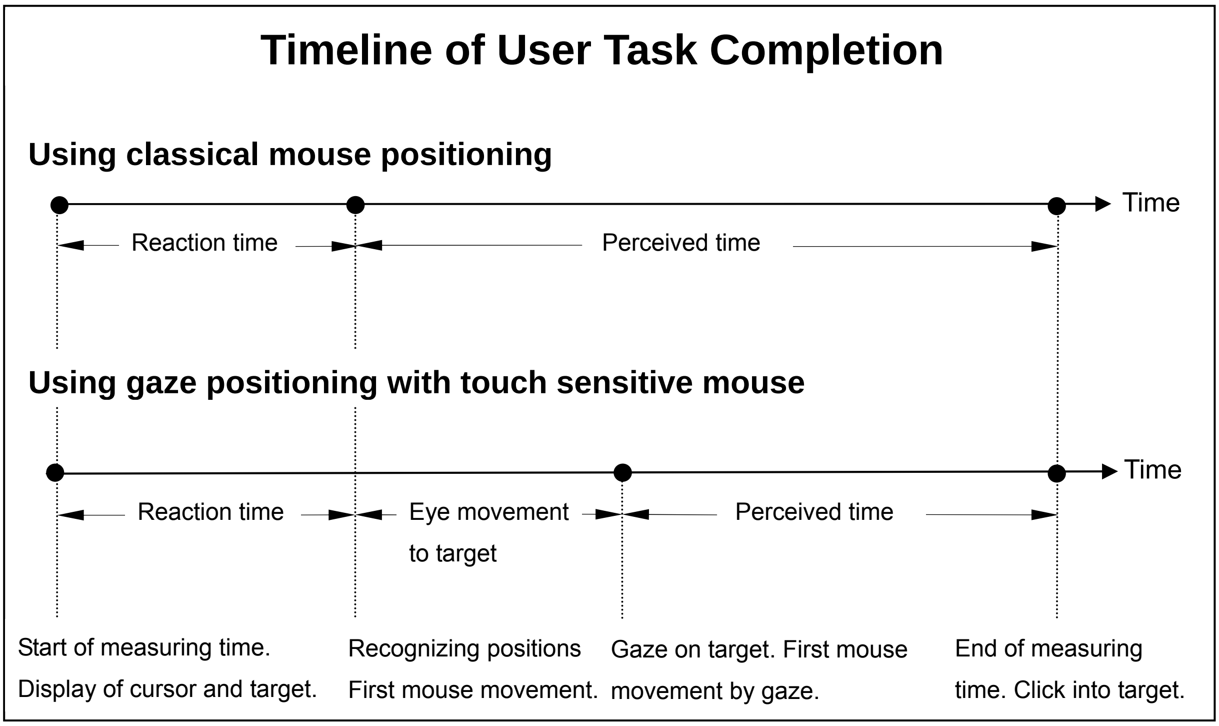
In the interviews after the blank and complex background task all but two users stated to be faster with the touch sensitive device under both conditions (see Table 11). However the measured times showed that this was not objectively true in the case of the blank background (see Table 9). One of the two users who did not perceive gaze positioning as faster insisted on equal performance time. This answer was not valid within the design of the study and finally his or her answer was counted in favour for mouse positioning.

Table 11: Judgments of speed after the tasks with blank and complex background.

Which method is faster?	Mouse Positioning	Gaze Positioning
After blank background	**2**	**8**
After complex background	**0**	**10**

The timeline diagram in Figure 48 gives a potential explanation. It shows typical timings for the condition with a white background. Even though the overall time needed is the same, the time users move the hand or finger is shorter when using gaze. This suggests that the time required moving the eye is not perceived as time needed to perform the task.

Figure 48: Timing analysis that explains the perceived speed-up with a touch mouse.



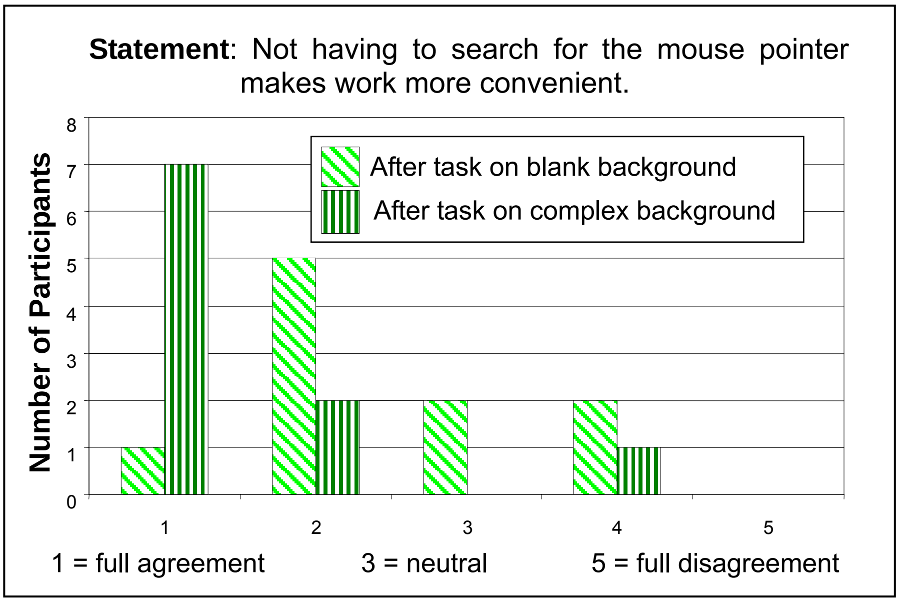
An analysis of the reaction times supports the assumption. Table 12 shows the times from displaying the targets to the first mouse event.

Table 12: Medians of the time of the first mouse move for all participants

	Median of the time of first occurrence of a mouse move (in milliseconds)			
	blank background		complex background	
Participant	mouse positioning	gaze positioning	mouse positioning	gaze positioning
P1	335.5	521.0	400.5	525.5
P2	225.5	361.0	145.5	385.5
P3	290.5	531.0	350.0	666.0
P4	330.5	626.0	360.0	541.0
P5	260.0	516.0	296.0	545.5
P6	285.0	656.0	175.5	631.0
P7	310.5	681.0	410.5	551.0
P8	310.0	766.0	381.0	611.0
P9	260.5	901.0	305.0	881.5
P10	305.5	490.5	225.0	521.0
Mean	**291**	**605**	**305**	**586**
Std. Dev	34	155	94	129

With the classical mouse the first mouse move takes place about 300 milliseconds after displaying the task. As soon as the participant realizes that the task started she or he starts to move the mouse. At this moment the eyes are not on the target; mouse pointer and gaze move simultaneously towards the target. With the touch-sensitive device the first mouse move takes place at about 600 milliseconds after displaying the targets. After realizing the start of the task the participants gaze moves to the target and the first mouse movement is triggered by the touch event. There is a more detailed discussion on hand-eye coordination in 4.4.2.

Figure 49: Judgments of convenience on a Likert scale from 1 to 5 for all users.



After each task the participants were asked whether it is convenient not having to search for the mouse pointer. The participants had to give their answer on a Likert scale from 1 (full agreement) to 5 (full disagreement). Figure 49 presents the results. After the task on complex background the agreement shifted towards full agreement.

From the informal talks with the participants it became clear that there is a very high acceptance for the gaze positioning of the mouse pointer. Most participants would like to use such technology if available. The big advantage of the raw positioning with the eye and fine positioning with the mouse lies in the fact that the approach does not change the functionality of the user interface but only adds a new function. The coordination of the movements – lift finger, look, touch, move mouse, click mouse key – requires some learning for some users. However as the principle is well-known and widely used for digital cameras, where in most cases a half-way press focuses and the full press takes the picture, it should not be a problem after some practice. The only matter of discussion was the method to trigger the positioning. The main concern was that a touch sensor on the left mouse key is inherently tied with normal use of the mouse key and it is not possible to click something without touching the mouse key. Some people suggested using an extra 'gaze positioning key' instead at the mouse device, for example at the thumb position of the right hand on the mouse or on the keyboard at the left hand side. The concern is not really justified as the triggering takes place at the raising edge of the touch signal and people tend to leave the finger on the mouse button while working with the mouse. However, as a 'gaze positioning key' is easy to realize, a commercial product can offer both methods and the users can choose according to their preferences.

It is obvious that the gaze positioning of the mouse pointer saves many mouse movements. The surprising observation in the study was that the people did not move the mouse at all. In the case of missing the target, the participants did not use the mouse to move the pointer to the target. Instead, they lifted the finger and repositioned again with the gaze. This was only possible because the target was big enough. See the third user study for results on small targets.

Lessons learned:

- People keep their hands on the mouse. Touch sensors for the fingers are more appropriate than a touch sensor for the hand. Lifting the finger is easier than lifting the whole hand.
- Triggering the gaze positioning by touching a sensor on the mouse button allows easy repositioning which is an advantage against the original MAGIC pointing where the mouse has to rest for a while.
- Triggering the gaze positioning by touching a sensor on the mouse button avoids the compensation methods against overshooting as needed for the original MAGIC pointing.
- The gaze positioning by touching the sensor gives the opportunity to check the pointer position before clicking the mouse button. It is a way to solve the feedback issue.
- On standard screen sizes and blank background the gaze positioning does not bring a speed benefit but it feels faster for the users.
- On standard screen sizes and complex background the gaze positioning brings a speed benefit because of saving the time to locate the mouse pointer.
- Gaze positioning saves many mouse movements. With big targets people nearly do not move the mouse at all.

4.4.2 Second User Study – Learning and Hand-Eye Coordination

One intention of the second user study was to find out more on the details of hand-eye coordination. For this the software was improved to record every fixation and all mouse events. Another goal was to measure learning effects. One observation from the first user study was that people repositioned the mouse pointer by a second touch when they missed the target. They did not move the mouse to bring the pointer into the target. This inspired testing gaze positioning without mouse but with a key on the keyboard called gaze key. To get closer to a vision of a system without mouse a run for what Ware and Mikaelian [Ware, Mikaelian 1987] called screen key was added. The basic task was the same as in the first user study but this time four pointing device conditions were used – classical mouse, touch-sensitive mouse, gaze key, and screen key – in combination with the two background conditions.

Classical mouse: Pointing with a classical optical mouse was accomplished by moving the mouse and selection by clicking the left mouse button.

Touch sensitive mouse: A touch on the left mouse button positioned the mouse pointer on the position of the gaze. This gave the users the opportunity to inspect whether the mouse pointer is in the correct position and go on with clicking the mouse button. If not on the target the user could move the mouse or touch the button again.

Gaze key: The 'delete' key on the number pad of a standard keyboard was used as gaze key. Pressing the gaze key positioned the mouse pointer at the position of the gaze and triggered a mouse click.

Screen key. The screen key is a two-target task. The pointing task starts with the display of a red and a grey circle both randomly positioned. After looking at the red circle the circle disappears and the grey circle becomes red. Looking at this second circle finished the task. During the whole task the mouse pointer was invisible and a cross-hair cursor was displayed as feedback for the gaze. Although the provision of feedback for the eye is problematic (see 4.1.3), there were two pragmatic reasons to introduce a cursor here. The first reason was to give the participants, who used an eye tracker for the first time in their lives, a confirmation that the system knows the position of their gaze. The second reason was the liberate head fixation with a chin rest only which bears the risk of spoiling the calibration by a small head movement. Displaying a cursor made it possible to detect calibration problems. The thin cross-hair cursor was chosen because it disturbs the perception less than the standard mouse cursor.

Nine volunteers took part, five were female, and four were male. The age range was from 23 – 47. All used traditional computers in their work or at home. The ethnics of the participants were mostly European but there was a Taiwanese person in the group. The ethnic background is mentioned here because the gaze behaviour seems to be cultural dependent [Chua, Boland, Nisbett 2005]. However, the data from this person showed no differences from the data of the others.

The experiment started with collecting basic data such as age and computer literacy. Then all participants got a short explanation to the overall system and an opportunity to try out the four different pointing techniques for about five minutes all together.

All participants had to complete four blocks. The total number of runs within one block was eight (2 backgrounds x 4 pointing devices). In each run the pointing task was repeated 35 times. In total each participant performed 1120 (8 x 35 x 4) pointing actions. One block took about 15 minutes. The first three blocks were during the first session with 5 minutes breaks in between. The final block was at a second session one week later to see whether people forget what they learned in the first three blocks. After both sessions structured interviews that lasted between 5 and 30 minutes were conducted.

The study tested the following hypotheses.

Hypothesis 1a: The task completion time for pointing is independent of the background conditions when using gaze.

Hypothesis 1b: The task completion time for pointing using a traditional mouse depends on the background condition.

Hypothesis 2: The use of a gaze key is faster than the traditional mouse (independent of the background condition).

Hypothesis 3: The gaze key is faster than the traditional mouse in both background conditions (already shown by [Ware, Mikaelian 1987]).

For the evaluation, the median of performance time out of every block of 35 pointing actions was taken and averaged over the four blocks of same pointing technique and background condition. This results in eight values for each person. Table 13 shows these values for each person.

Table 13: Average of median times in milliseconds for four different pointing techniques without and with background for all participants

	Median for total times (in milliseconds)							
	Classical mouse		**Touch mouse**		**Gaze key**		**Screen key**	
	no bg	bg	no bg	bg	no bg	bg	no bg	bg
P1	671.00	706.00	818.50	773.25	525.25	520.50	848.50	1003.75
P2	968.50	1074.50	1169.50	1046.75	738.50	583.25	1356.75	1269.50
P3	1011.25	1206.75	931.00	1086.50	566.00	613.50	1086.75	1179.50
P4	799.00	946.25	1011.75	1179.25	586.00	578.25	979.00	1191.75
P5	726.00	871.00	738.50	736.00	468.25	528.25	974.00	984.25
P6	773.50	909.00	828.50	869.00	493.25	513.25	1109.50	1119.25
P7	756.00	978.75	1008.75	1006.75	693.25	678.75	1016.25	1344.00
P8	1083.75	1209.50	2013.25	1582.25	834.00	866.25	1227.00	991.25
P9	934.25	1131.50	881.75	758.50	553.25	571.00	893.75	914.00
Mean	**858.14**	**1003.69**	**1044.61**	**1004.25**	**606.42**	**605.89**	**1054.61**	**1110.81**
Std dev	144.00	167.35	385.23	268.63	122.55	110.42	160.89	146.38

Table 14 shows the means over values and persons for each input modality and background condition. There is a clear order in performance for the four different pointing techniques tested. The fastest pointing action is the gaze key. The reason why this input method is fast is that positioning and selection of the target happens as a single action. The touch mouse mechanism in contrast consists of two separate actions. The touch event positions the mouse cursor, the user checks for correct position and clicks the mouse button for selection. The time difference between both pointing methods is the time to check whether the pointer is in the right position and the second movement of the finger for the click. If a user gets used to the touch sensitive mouse and trusts in the accuracy of positioning, the same speed as for the gaze key method should be possible.

The second fastest pointing action is the classical mouse on blank background. Modern mouse devices are highly optimized and the participants are well trained on that device. Repetitions of the experiments of Card [Card, English, Burr 1978] or Buxton [MacKenzie, Sellen, Buxton 1991] as exercise in student courses normally produce higher speeds nowadays.

Table 14: Average times in milliseconds for four different pointing techniques on blank and complex background

	Blank background		Complex background	
	Mean	Std. dev	Mean	Std. dev
Classical mouse	**858**	144	**1003**	167
Touch mouse	**1044**	385	**1004**	268
Gaze key	**606**	122	**605**	110
Screen key	**1054**	160	**1110**	146

On complex background the classical mouse has the same speed like the touch sensitive mouse and the screen key. The reason for this lies in the fact that on a blank background the user knows the position of the mouse cursor by pre-attentive perception and therefore she or he moves the mouse immediately towards the target. With complex background the user has to search for the mouse cursor first.

Hypothesis 1

Table 15 shows a t-test comparing the times with and without background. Both hypothesis 1a and 1b can be accepted from the experiment. The data clearly indicates independence of the background condition for all gaze pointing methods, see row 1-3 in Table 15. In contrast, the classical mouse depends on the background condition as indicated by the value in row 4.

Table 15: T-test (p-values) comparing four different pointing techniques on blank and complex background

Blank vs. complex background	t-test
Touch mouse	**0.52**
Gaze key	**0.99**
Screen key	**0.40**
Classical mouse	**0.000055**

Hypothesis 2

From the data gathered hypothesis 2 has to be rejected. The data cannot confirm that the touch mouse is faster than a traditional mouse if a complex background is present. This is in contrast to the results from the first user study. One participant seemed to be stressed and had extreme difficulties with the coordination of the finger on the touch sensor, got upset, and blamed the apparatus not to work correctly. Therefore, this participant needed twice the time than the others. This pushed the value for the average up to 100 ms more and lead to a high standard deviation. Without the data of this participant the hypothesis could be confirmed.

Hypothesis 3

The data confirms hypothesis 3. The gaze key is indeed faster than the traditional mouse in both background conditions. The calculated significance of the t-Test is 0.00016 in the case of a white background and 0.000016 in the case of a complex background.

Learning

In the first user study it appeared that the participants performed better in the second task with the touch sensitive mouse hence there could be some learning. The expectation was that the users are well trained with the mouse and no learning will be seen for the classical mouse. For the touch-sensitive mouse the expectation was a learning curve.

The data from the study did not confirm this expectation. First, there were learning effects for the classical mouse experiment. Because all participants were experienced mouse users, they must be at the very flat part in the learning curve. Consequently, this means the participants did not learn the pointing action but the overall experiment. Therefore it is not possible to provide an insight if and to what extent learning occurs for the gaze pointing. Additionally it is in general problematic to fit the results to a learning curve if there is no model behind [Heathcote, Brown, Mewhort 2000].

It seems there is no learning for the gaze key. The eye is trained already and there is nothing to learn pressing a key. The learning effects for the touch sensitive mouse were not uniform. For the last block after one week the expectations were that people might have forgotten some of the abilities. However, some participants performed much better after a week without practice. It seems that learning motor tasks is a complex process or that other factors like the mood or fitness at the day of the study have effects on the performance.

Hand-eye coordination

A detailed analysis of the mouse and eye movement in the classical mouse experiment provides insight into the hand-eye coordination in pointing tasks. The main reason for this analysis was to show that using gaze positioning does not put extra load to the eye muscles and does not cause more stress on the eyes than working with a traditional mouse.

In general, the mouse pointer and gaze move independently towards the target where they meet. Throughout the experiment it became clear that the gaze is in general earlier at the target than the mouse and hence the gaze is overall faster than the mouse. Nevertheless, the mouse can be temporarily faster than the eyes because of accelerator techniques in the mouse driver (see the slopes in Figure 51 and Figure 53).

Figure 50 depicts a mouse trail and a gaze path. Both paths move independently towards the target. The dotted grey lines connect points of same time on mouse and gaze trail. From this it can be seen that the gaze is already in the target when the mouse starts moving towards the target. As the eye does not rest longer in the same position than for the time of a typical fixation, it seems that reaching the target it moves again.

Figure 51 shows the distance of gaze and mouse pointer to the centre of the target over time. The size of the target is shown as grey area. The reaction time to start is about the same for both operations but the eye moves much quicker towards the target.

Figure 50: Gaze (dashed) and mouse movement (solid) for the classical mouse task without background. The dotted grey lines connect points of same time.

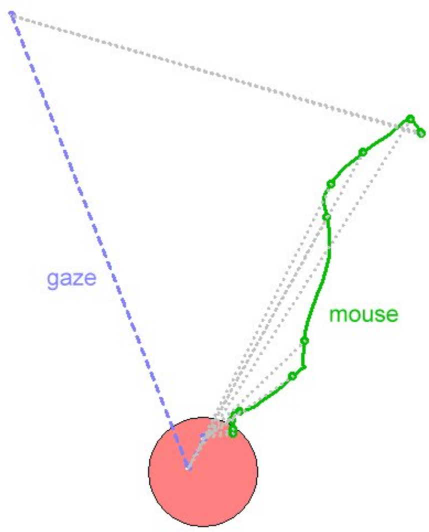


Figure 51: Gaze (dashed) and mouse movement (solid) for the classical mouse task without background as a plot of distance to target over time.

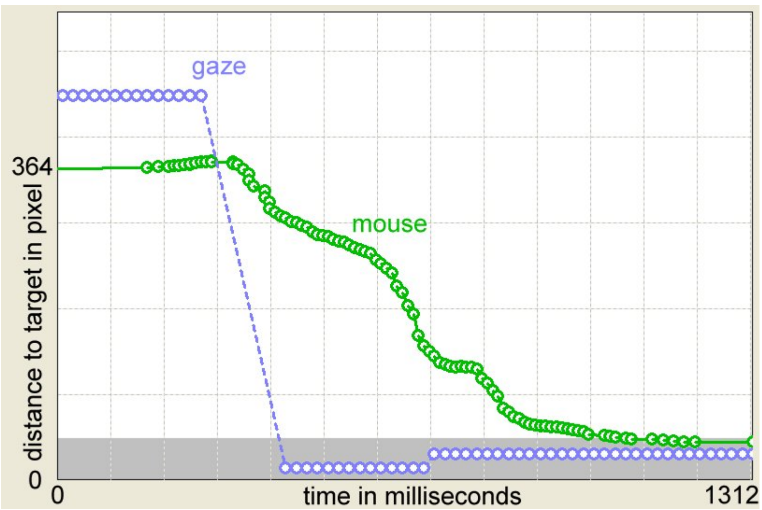


Figure 52 depicts an example of pointing on a complex background. Figure 53 shows the corresponding plot for the distance to the target over time. The eye moves directly towards the target and arrives there very fast. In contrast, the mouse first does not move towards the target but appears to move in a random direction to create a visual stimulus. As soon as the eye detects the position of the mouse pointer, interestingly without moving the gaze to look, the mouse moves towards the target. It shows that peripheral vision of movement is sufficient to locate the position of the mouse pointer. This result allows predicting that people will get problems to find a pointer when it is outside of the peripheral vision area. This may be the case when standing close to a very large display.

Figure 52: Gaze (dashed) and mouse trail (solid) for the classical mouse task with background. In the beginning, the user stirs the mouse to detect its position by movement.

Figure 53: Gaze (dashed) and mouse trail (solid) for the classical mouse task on complex background as a plot of distance to target over time.

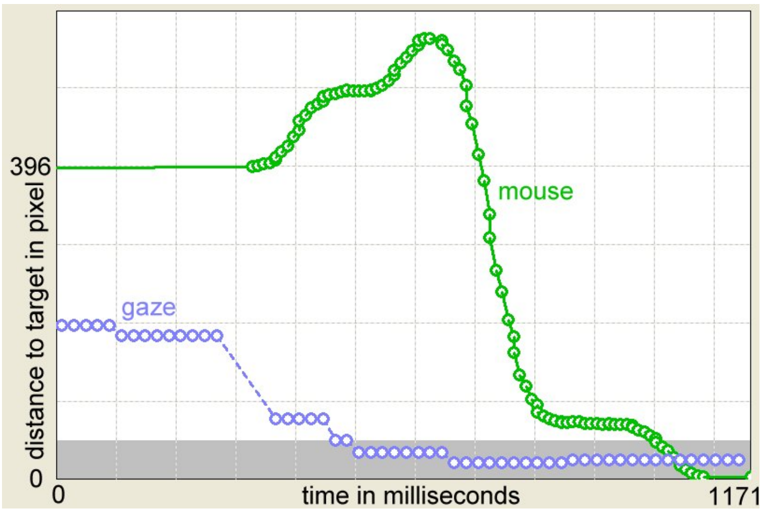
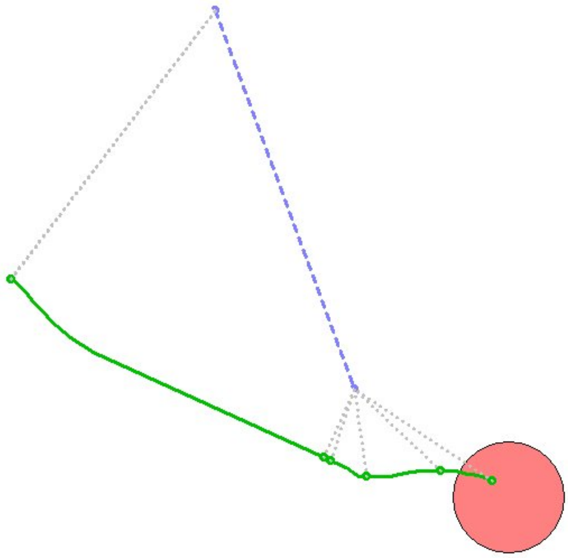


Figure 52 and Figure 53 show that the eye stopped on the way to the target. However, there is no explanation for this behaviour. Such stops happen sometimes but not always. In approximately 20% of the recordings (490 of 2520) the eye gaze did not even hit the target but only moved close to it. The circle experiment (see 3.4.4) showed that, when offering big targets, the eye gaze does not move to the centre of the target but already stops after slightly crossing an edge of the target. The first idea was that not hitting the target could be an effect of calibration drift. Closer inspection revealed that the effect cannot be explained with inaccuracy and calibration errors; the end point of the gaze trail in Figure 54 is definitely too far away from the target and the data of the pointing tasks performed before and after that do not show any irregularities. About 6.7% (170 of 2520) of all interactions showed that the eye gaze moved less than 90% of the distance from the initial gaze position to the target. This leads to the conclusion that it is possible to hit a target without looking at it.

Figure 54: Gaze (dashed) and mouse trail (solid) for the classical mouse task on blank background. Here the eye gaze does not hit the target.



Attempts to reproduce such data intentionally showed that this is indeed possible. The light sensitive receptors on the retina are not distributed uniformly. There is only a small spot, the fovea with a visual field of about 1°, where the density of the receptors is high and makes clear vision possible. In the rest of our visual field the resolution is quite low. However, the target used is big enough and its red colour is striking and can thus easily be detected with peripheral vision. The position of the standard mouse pointer is less obvious but can be easily located because of its movement. Consequently, it is possible to click on the target without directly looking at it. Nevertheless, such cases do not contribute much and finally the distance covered by the gaze in a classical mouse task is about the same as for gaze pointing.

Lessons learned:

- The second user study confirmed most of the results from the first user study and confirms all results when treating the data of one participant as outlier.
- On textured background where it is difficult to find the mouse pointer, people start to stir the mouse to locate the pointer. The detection of the moving pointer uses peripheral vision – the gaze does not move to the mouse pointer and does not follow the pointer.
- The gaze key is the fastest interaction because it does not provide feedback and there is no checking whether the gaze positioning is correct.
- The screen key method as a gaze-only interaction does not bring speed benefit compared to classical mouse or dwell time method.
- Learning the coordination of gaze, hand and finger movements do not follow a simple law. Some people can do it instantly while others struggle with it.
- Both eye gaze and mouse pointer move independently towards the target. Therefore, using gaze positioning does not put additional load to the eye as the eye moves to the target anyway.

4.4.3 Third User Study – Raw and Fine Positioning

The third user study deals with the accuracy problem. The aim of the study was to show that gaze positioning with a touch-sensitive mouse brings benefit for the user also in the case of small targets. Zhai et al. showed this already for MAGIC pointing. The MAGIC touch presented here only differs in the trigger mechanism for the gaze positioning but not in the concept of raw and fine positioning. Therefore, the user study was done for completeness and not with the intention to repeat the proof.

The observations from the studies before showed that the sequence – lift the finger, look, touch the mouse key, move the mouse, click the mouse key – is difficult for some people. To avoid a long period of practice to become familiar with the coordination the third user study was done with the four best participants of the second user study. With the small number of participants it is called user study only for the consistency in the headings.

Observations from the two previous studies showed that the participants did not move the mouse at all and repositioned with the gaze in case of missing the target. This was only possible because the target was big enough hitting it by gaze. This experiment forced the participants to move the mouse because of smaller targets which are not easy to hit by gaze. The circle experiment (see 3.4.5) showed a target of 40 pixels diameter is still possible to hit with the gaze but for a target of 20 pixels diameter this is very difficult. This user study was a repetition of the first one with 50 clicks per task but this time with target sizes of 20 and 40 pixels instead of 100 pixels.

Table 16: Medians for total time in milliseconds of all participants

	Median for total times (in milliseconds)							
	20 pixel target size				40 pixel target size			
	blank background		with background		blank background		with background	
Participant	mouse	gaze	mouse	gaze	mouse	gaze	mouse	gaze
P1	1041.5	1542.0	1111.5	1482.0	886.0	1121.0	1021.5	1066.5
P2	1156.0	1502.0	1542.0	1532.0	1011.5	1186.0	1412.0	1202.0
P3	1001.0	1602.5	1181.5	1602.5	1031.0	1276.5	1117.0	1282.0
P4	936.0	1427.0	1071.5	1497.5	831.5	1042.0	1052.0	1261.5
Mean	**1033.6**	**1518.4**	**1226.6**	**1528.5**	**940.0**	**1156.4**	**1150.6**	**1203.0**
Std. Dev.	92.4	73.6	215.1	53.6	96.7	99.4	178.7	97.1

Table 16 shows the results for both target sizes. There is no speed benefit for the touch-sensitive mouse with gaze positioning if a fine positioning into a small target is necessary. Nevertheless, the distance covered by the mouse is much smaller. The participants used the classical mouse for their daily work and therefore were highly trained on this device, while the touch-sensitive mouse and gaze positioning was a new interface technique for them. It is possible that the performance with the MAGIC touch will increase after some training.

Lessons learned:

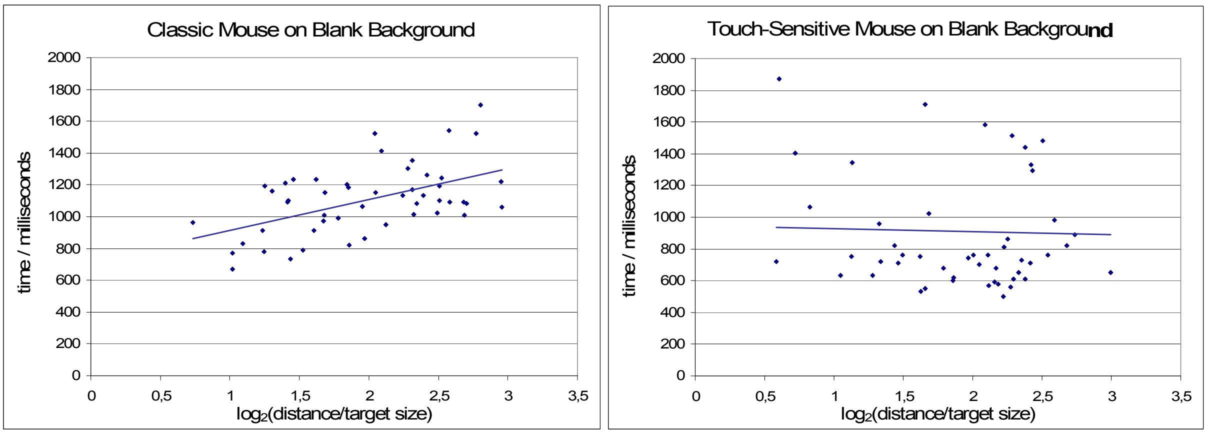
- The MAGIC touch principle works also on small targets where fine positioning is necessary.
- Raw positioning with the gaze and fine positioning with the mouse takes the same or even more time than classical mouse positioning.
- Even when mouse movement is necessary, there is still an enormous saving in distance covered by the mouse.

4.5 A Deeper Understanding of MAGIC Touch

The research of MAGIC touch followed the typical pattern of scientific work. It started with identifying a problem, the users have problems to find the mouse pointer, continued with an idea to solve the problem, the touch-sensitive mouse and gaze positioning, which caused setting up and conduct a user study. The user study provided data which were evaluated with statistical methods. Statistical methods however, require a lot of care to create reliable results. The first demand for a statistical evaluation is a representative and large data set. It is questionable whether the user studies presented here fulfil this demand. The 10 participants in the user study are not a big group and therefore cannot represent all ethnics, all age groups, and all levels of education. Another demand is that the data obey certain assumptions like Gaussian distribution for a legitimate application of statistical tests. Typically, a proper statistical evaluation must show that the data obey the demanded assumption, which was not done for the data presented here. Finally, statistical evaluations always have an uncertainty that the values are a result of chance. The fact that researchers only publish results where they got significance and throw away the other results worsens the situation. Statistical evaluations can prove a statement but do not give an explanation or causality. Therefore, statistical methods are a valuable tool for the case there is no other way to gain knowledge. However, whenever it is possible other methods are preferable. This was the motivation to think a bit deeper about the MAGIC touch principle.

The completion time for the traditional mouse increases with distance from target to pointer (obeying Fitts' law). In contrast, locating the target with the gaze is independent of the mouse pointer position. For the MAGIC touch principle the eye has to find the target and after that the finger has to touch the sensor on the mouse. The location of the mouse pointer is irrelevant and this leads to constant completion time independent of the distance of mouse pointer to target. A monotonically increasing function as it is the case for the classical mouse and a constant function as it is the case for gaze positioning cross each over. Figure 55 shows the plots of the time for the pointing task versus the index of difficulty ($\log_2$(distance/target size)). The experiment was conducted with only one target size and strictly, it is a plot of time over the logarithm of distance. To give a point of comparison it is presented here as typically done for the evaluation of Fitts' law.

Figure 55: Plots of time to click the target (of 100 pixels) against the index of difficulty for classical mouse and for gaze positioning with the touch-sensitive mouse.



In Figure 55 it is easy to see that the time to hit the target depends on the distance for the classical mouse while for the touch-sensitive mouse it does not. The crossover is at about 400 pixels, the value used by chance in the study. For this reason the performance times in the user study for classical mouse and gaze positioning were equal. This means that for longer distances there will be a speed benefit also in the case of a blank background.

This means that there is a fundamental insight on the time relation for the blank background condition. Statistical significance of experimental data as proof is not necessary any more; it is not worth the effort to repeat the user study with randomized task order and more participants. For the classical mouse in complex background condition it is also not necessary to proof the increase of performance time against the blank background condition with a statistical evaluation. It is common sense that stirring the mouse for locating the pointer takes time. The only benefit from a statistical evaluation would be precise values for the crossover distance and the stirring time. However, the situation in the user study is very artificial and not directly transferable to ordinary working situations and a precise value holds only for the situation in the user study. With respect to individual differences and noise the presented values are good enough for the magnitude of the effects.

This late insights also teaches to be careful with conclusions based on statistical significance. The small low-budget eye tracker with a 12-inch display was the reason why the average distance in the user study was 400 pixels. The same user study with an eye tracker with a 17-inch display would have had an average distance of 600 pixels. In consequence, the user study would have created data which show a significant speed benefit for the touch-sensitive mouse even on blank background. This example shows that significance alone without understanding can be misleading. Results based only on statistical values should be taken with care and in general it is better to seek for deeper understanding.

Lessons learned:

- Gaze positioning is independent of the distance of mouse pointer to target while mouse positioning depends on that distance. The crossover is at about 400 pixels (11°).
- Significance in the data of a user study does not mean that we understand the situation.

4.6 Summary on the Results for Eye Gaze as Pointing Device

The findings are:

- Gaze pointing does not put additional load to the eyes because the eyes move to the target anyway.
- Using gaze positioning reduces the need for mouse movement significantly.
- The task completion time for pointing with a touch-gaze system is independent of the background.
- The task completion time for pointing with a touch-gaze system is independent of the position of the mouse pointer.
- For large targets (larger than 3°) and long distances (larger than 11°) the touch-gaze system is faster than a traditional mouse. On complex backgrounds, the touch-gaze system is already faster for medium distances (larger than 8°).

4.7 Conclusions on Eye Gaze as Pointing Device

The use of a gaze pointing technique saves a significant part of mouse movements without incurring additional load to the eye. The use of eye gaze pointing will therefore not fatigue the eye muscles more than with traditional mouse pointing. With the common trend to bigger displays the distances covered by the mouse will become longer. Gaze positioning in general seems to be a desirable input technology because it reduces the necessary mouse movements and less space on the desk is required for the mouse.

The eye gaze has an intrinsic inaccuracy, which is bigger than the size of standard GUI interaction elements. The easiest way to design a gaze-aware user interface is to increase the sizes of the interaction elements, as it is the case for the systems used in the field of accessibility. Together with a gaze key such systems have the possibility to be quicker to operate than with a mouse (as long as the effect of less possible targets does not cause an increase of needed selections). Such specifically designed new interfaces could make the mouse obsolete. Such systems could increase the overall speed as there would also be no need for the homing of the hand between mouse and keyboard.

A touch-sensitive positioning key on the keyboard could be one potential solution to solve the feedback problem. The positioning on touch gives the user the chance to check whether the eye hit the target correctly before confirming the operation with a key press. However, the necessary increase of interaction elements on the screen is about factor 3 in each dimension and causes a decrease of displayable information by factor 9 or roughly 10.

A gaze-aware user interface, which keeps the sizes of the GUI elements, must solve the accuracy problem. From the suggestions mentioned in the related work section the MAGIC pointing with the concept of using the gaze only for raw positioning seems to be promising. In contrast to the other methods like expanding targets [Miniotas, Špakov, MacKenzie 2004] or enlarging parts of the display [Kumar, Paepcke, Winograd 2007], the MAGIC pointing does not need any changes inside the GUI code and it also does not change the appearance of the GUI to the user.

Enhancing the MAGIC pointing with a touch-sensitive mouse saves all compensation techniques and gives more control to the user. Touching a key with the finger is more intentional than doing a first mouse move and it offers the possibility to reposition the mouse pointer immediately by lifting the finger and touching the key again.

Using the gaze as the only input modality has many severe challenges but the combination with a touch sensitive mouse key solves some of them – the Midas Touch problem, the feedback issue and issues related to inaccuracy. The combination of an eye tracker with a touch sensitive mouse key offers a way to utilize the gaze as an additional pointing device without changes in the behaviour of the GUI. There is only a little speed benefit in the case that the user is not aware of the mouse pointer position, but the savings in distance covered by the mouse are enormous. During the interviews there was very positive feedback on gaze positioning. The possibility to position the mouse pointer at the position of the gaze just with a little movement of the fingertip feels pleasant. This leads to the conclusion that the use of gaze pointing is desirable.

5 Gaze Gestures

Gestures are a well-known concept for computer human interaction. The idea behind gestures is the fact that we are used to employ movements of the body, mostly with the hands and the head, to communicate or to support communication. While such intuitive gestures are vague and culture dependent we are also able to perform well-defined and elaborated gestures. One example is handwriting; other examples are the sign language for deaf mutes or the semaphore alphabet.

Most research deals with gestures done with a pen or with a mouse. Some examples are Unistroke [Goldberg, Richardson 1993], Cirrin [Mankof, Abowd 1998], EdgeWrite [Wobbrock, Myers, Kembel 2003] and the mouse gestures for the Firefox web browser [mouse gestures@]. Typical usage for such gestures is text entry with gestures for the letters or directing the computer with gesture commands.

Gestures for computer interaction are not very intuitive, as it requires learning a set of gestures and their semantics. For this reason the use of gestures for computer interaction has been seen as something for specialists or for very specific purposes. However, with the introduction of the iPhone and similar products, which have a touch sensitive surface as the only input modality, the use of gestures performed with the fingers became popular. In addition, the interaction with tabletop computers, which will appear on the mass market soon, will strongly rely on gesture input.

The research presented here focuses on gestures performed with the eyes.

5.1 Related Work on Gaze Gestures

We use eye movements for our human-human communication which could be called eye gestures. Typical examples are to wink, to blink with the eye, or to roll the eyes. Such eye movements can include movements of the eyelid and eyebrows and may be seen as part of a facial expression. See [Ravyse, Sahli, Reinders, Cornelis 2000], [Ravyse, Reinders, Cornelis, Sahli 2000], [Ying-li, Kanade, Cohn 2000] and [Morimoto, Koons, Amir, Flickner 1999] for research on this topic. This work restricts itself to the eye movements reported from a commercial eye tracker which represent the direction of the gaze. For this reason the gestures performed with the eye are called gaze gestures and not eye gestures.

There is only a small body of research on concepts like gestures done with the eyes. Qvarfordt and Zhai built a dialog system, which uses eye-gaze patterns to converse with the user [Qvarfordt, Zhai 2005]. They studied gaze patterns in human-human dialogs and used the results to mediate a human-computer dialog. The users did not learn gaze gestures to operate the system and were not even aware that they were performing gestures. This approach is in accordance with the sceptical opinion towards motor control tasks for the eyes in [Zhai, Morimoto, Ihde 1999].

Isokoski did the first close research to gaze gestures as introduced here. He suggested the use of off-screen targets for text input [Isokoski 2000]. The eye gaze has to visit the off-screen targets in a certain order to enter characters. The resulting eye movements are gaze gestures. However, off-screen targets force the gesture to be performed in a fixed location and with a fixed size. In contrast, the gaze gestures presented here are scalable and can be performed in any location.

Milekic used the term gaze gesture already in 2003 [Milekic 2003]. Milekic outlined a conceptual framework for the development of a gaze-based interface for use in a museum context but being from a Department of Art Education & Art Therapy his approach is not strictly scientific – there is no algorithm given and no user study was done.

In 2007, the same year of publication of the gaze gestures [Drewes, Schmidt 2007] described below, Wobbrock et al. published the same idea to use EdgeWrite gestures for gaze entry of letters [Wobbrock, Rubinstein, Sawyer, Duchowski 2007]. The authors claim that an experienced EdgeWrite user can write 7.99 WPM (words per minute). The results from the user studies presented in this work indicate that the entry of letters with gestures is slower than eye typing using the dwell time method.

In 2008 while writing down this thesis three further publications on gaze gestures appeared [Porta, Turina 2008], [Bee, Andre 2008] and [Wobbrock, Rubinstein, Sawyer, Duchowski 2008]. All three publications focus on gaze gestures for text input.

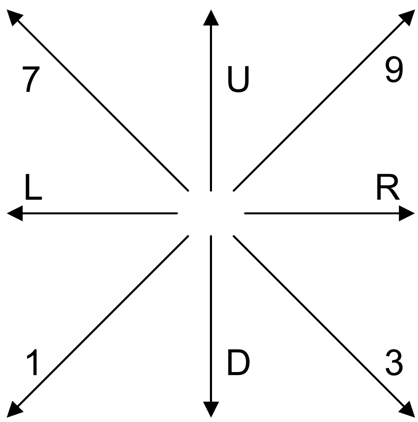
Also in 2008, Bulling et al. published papers on gaze gestures with eye tracking by electro-occulography [Bulling, Roggen, Tröster 2008a], [Bulling, Roggen, Tröster 2008b]. They refer to the gaze gestures and algorithm from this work and confirm the results presented here. Their research shows that the concept of gaze gestures does not depend on the eye-tracking technology used.

5.2 The Concept of Gaze Gestures

5.2.1 The Firefox/Opera Mouse Gestures

The popular and freely available mouse gesture plug-in for the Firefox web browser [mouse gestures@] gave the inspiration to implement a similar gaze gesture algorithm. The mouse gesture plug-in traces the mouse movements when a gesture key, normally the right mouse key, is pressed and translates the movements into a string of characters or tokens representing strokes in eight directions as depicted in Figure 56.

Figure 56: The naming of the eight directions used for the mouse gestures.



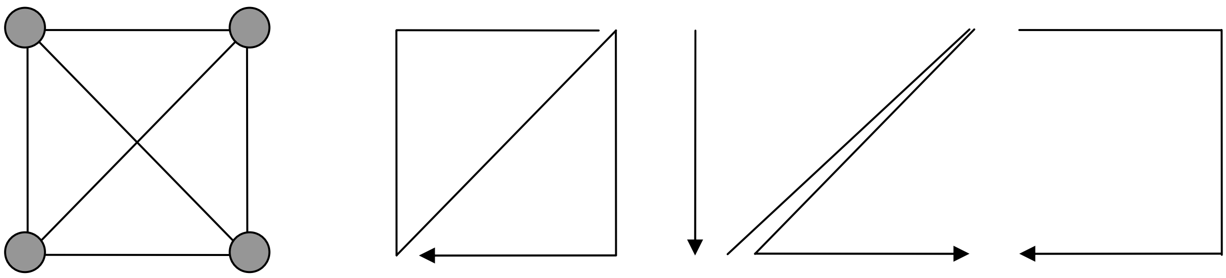
The mouse gesture recognition algorithm of the Firefox browser receives the coordinates of the mouse device. Whenever the mouse pointer moved more than a threshold distance away from the start position the algorithm outputs a character for the direction of the movement. The current mouse pointer position becomes the new start position and the detection of a new stroke starts. The algorithm uses eight directions; U, R, D, and L for up, right, down and left respectively and 1, 3, 7, and 9 for the diagonal direction according to the standard layout of the number pad on the keyboard. The notation of the direction follows the notation introduced by the mouse gesture algorithm.

The definition of a gesture is a string consisting of the eight characters for the eight directions. As the mouse gesture recognition does not produce a new character if the direction did not change, the string must not contain pairs of the same character.

5.2.2 The EdgeWrite Gestures

Beside the mouse gestures the EdgeWrite gestures [Wobbrock, Myers, Kembel 2003], [Wobbrock, Myers 2006] were most inspiring for this work. The EdgeWrite gestures use the order in which four points, the corners of a square, are reached (see Figure 57). It is easy to see that all EdgeWrite gestures can be expressed with the direction characters of the mouse gestures. Consequently, the EdgeWrite gestures are a subset of the mouse gestures. Nevertheless, they have the capability to define a large alphabet as shown by Wobbrock et al. who assigned at least one gesture to all letters and digits in the Latin alphabet. As a display has four corners to look at, the Edgewrite gestures are easy to adapt to the gaze.

Figure 57: The four corners and the six connecting lines used for the EdgeWrite gestures and examples for EdgeWrite gestures (digits 0, 1, 2 and 3).



5.3 The Gaze Gesture Recognition Algorithm

A gesture consists of a sequence of continuous elements, typically strokes, which are performed in a sequential time order. To detect a gesture it is necessary to know when the gesture starts and when it ends. In the case of pen input the gestures starts at the moment the pen touches a surface and ends when lifting the pen. For the mouse gestures the situation is different, as it does not help to lift the mouse pointer. For this reason the mouse gestures use a gesture key, the right mouse button in the standard configuration, to indicate the input of a gesture. This makes sure that the algorithm does not detect gestures during normal mouse input and it allows defining an alphabet where one gesture pattern is a part of another gesture pattern.

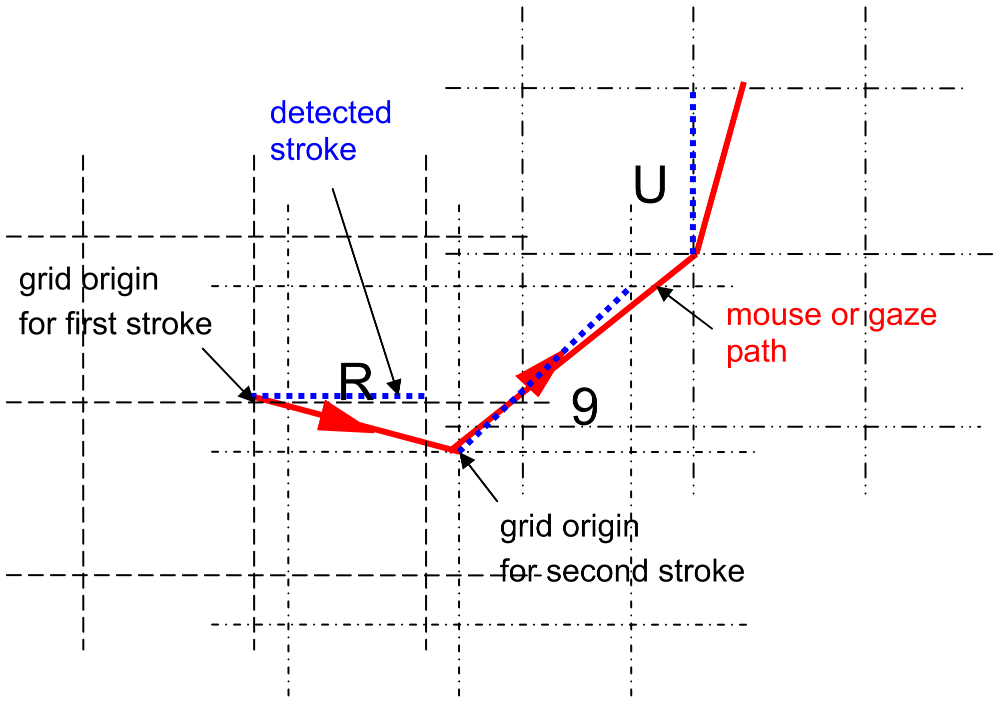
The situation for gaze gestures is similar to the mouse gestures because we cannot lift the gaze like a pen. However, gaze gestures should avoid the need of a gesture key. With the need of a gesture key the gaze gestures lose the properties of being a remote control and a hygienic (touch free) interface which is their main benefit.

The absence of a gesture key requires continuous gesture recognition and creates the problem that natural eye movements could produce a valid gaze gesture without intention. The problem of separating gaze gestures from natural eye movements is very similar to the challenge of recognizing spoken commands for computer input where a normal conversation should not trigger commands.

The mouse gesture detection algorithm works in a quite simple way. The algorithm receives x-y positions and calculates the difference to the start position. As long as an integer division with grid size *s* produces zero for both components the algorithm continues to receive further x-y positions. As soon as the integer division has a result different from zero for at least one component, the current position becomes the new start position and the algorithm produces a character for the direction of the stroke as output, but only if this character is different from the last output. See Figure 58 for an illustration. The algorithm detects a gesture if the output of the last characters matches a given gesture string.

To achieve a good separation from natural eye movements the demand is a performance of the gesture without break. This means that a time aspect has to be introduced into the detection algorithm. Whenever the algorithm did not detect a stroke for time *t* it outputs a colon as the ninth possible character. Gazing longer than time *t* at the same position resets the gesture recognition. The output of a colon also sets a new start position. This becomes important when working with big grid sizes i.e. grid sizes bigger than half of the display size.

Figure 58: The figure shows how a mouse or gaze path translates into the string R9U. The end of a detected stroke is the origin for the grid to detect the next stroke.



5.4 User Studies and Experiments with Gaze Gestures

To test the applicability of gaze gestures four user studies and some experiments were conducted. The goal of the first user study was to find out whether people are able to perform gaze gestures intentionally. The experiments were done to find possible applications for the gaze gestures. The second user study focused on the optimization of the parameters grid size *s* and timeout *t*. The third user study tested gaze gestures on small displays of mobile devices and the fourth user study used gaze gestures for secure PIN entry at ATM cash machines.

5.4.1 First User Study – Testing the Concept

Nine people, six male and three female, in the age from 23 to 47 years, participated in the user study. All persons had a European cultural background and academic education. They were familiar with traditional computer input and used computers regularly in their work but they had never used an eye tracker before.

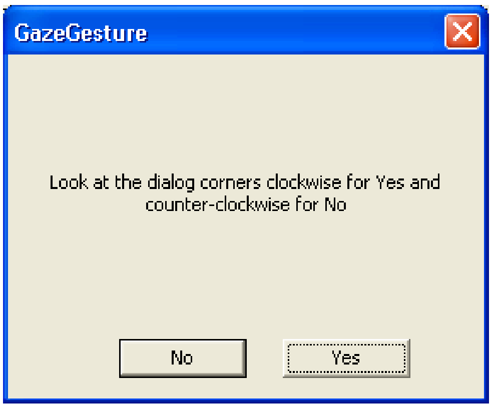
Design of the User Study

The software (see Figure 61) for the user study was purpose-written to allow parameter adjustments on the grid size and timeout. For this user study the timeout was 1000 milliseconds and the grid size was 80 pixels. See 2.6.2 for the geometry of the setup and the conversion of pixels to degrees.

Prior to the study the participant got a brief introduction to the system. The user study consisted of three different tasks.

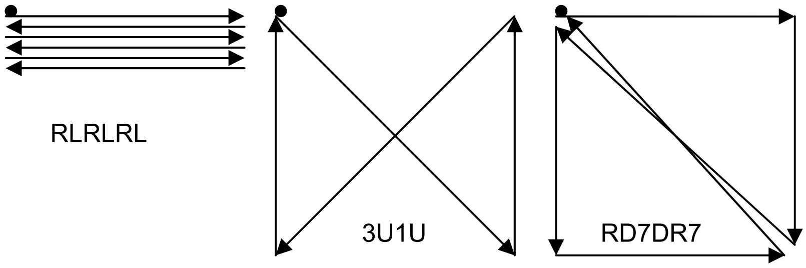
The first task was to close a dialog (depicted in Figure 59) by using gaze gestures instead of the mouse. The participants were instructed to perform the action by visiting the corners of the dialog with their gaze clockwise for YES and counter-clockwise for NO.

Figure 59: The first task in the user study was to close a dialog with a gaze gesture.



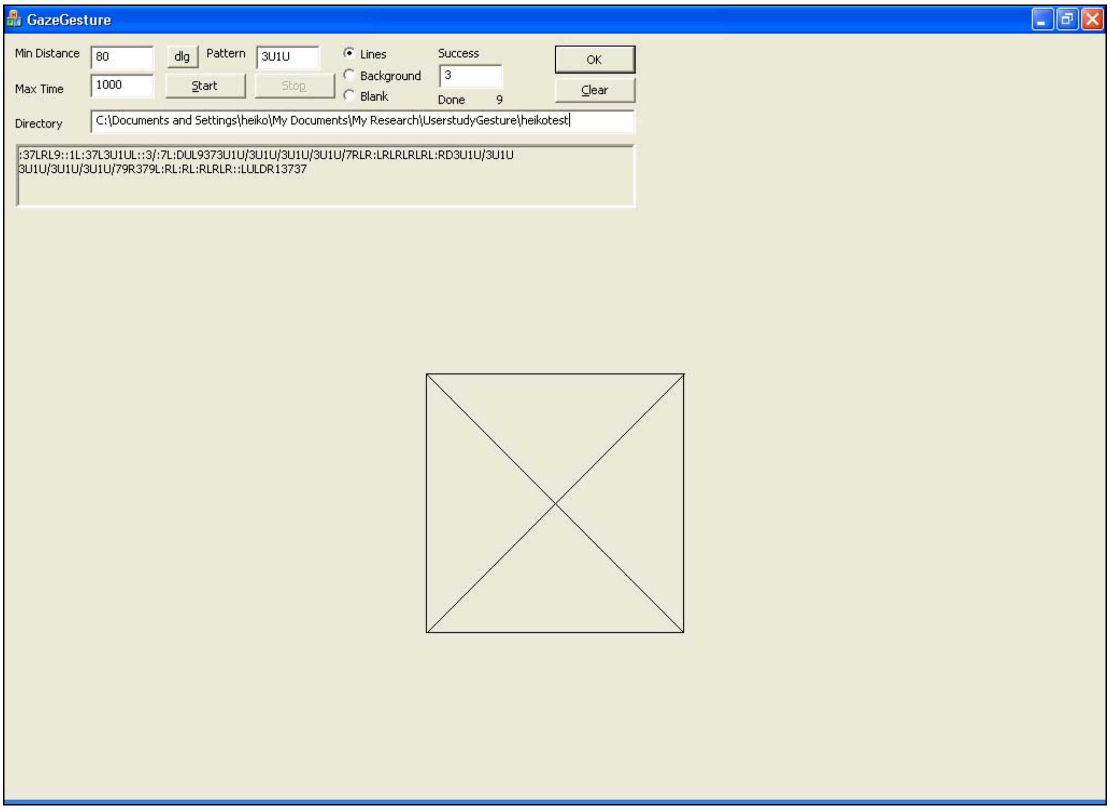
In the second task the participants had to do three different gaze gestures of increasing complexity on three different backgrounds. To make sure that the user was able to do the requested gesture each gesture had to be repeated three times. This resulted in 27 gestures per candidate. The gestures used were RLRLRL, 3U1U and RD7DR7, see Figure 60 for an illustration. For each performed gesture the required time was recorded.

Figure 60: The three gaze gestures of increasing complexity investigated in the first user study.



The first background showed an outlined square with diagonal lines as shown in Figure 61. The helping lines were given to guide the gaze gestures. The second background was a screenshot of a desktop with an open spreadsheet document. This enabled the test users to choose positions for fixations. The third background was just blank.

Figure 61: Helping lines to do gaze gestures. In the second and third task these lines were not displayed.



The third task was to surf the internet for three minutes. The gesture recognition software logged the resulting gesture string. The reason for this task was to find out whether the gestures occur during normal work.

Results of the User Study

All users were immediately able to close the dialogs by gaze with YES and NO using gaze gestures. The average time to perform the gesture was about 1900 milliseconds (see Table 17) with a standard deviation of about 600 milliseconds. All participants reported this as an easy task.

Table 17: Mean times to perform the gesture for closing the dialog in the first task

Time in milliseconds to perform gesture (Average over all subjects)	
Gesture	**Gesture time**
YES (clockwise all corners)	1905
NO (counter-clockwise all corners)	1818

In the second task, where the participants had to perform the three different gestures, all users were able to perform all gestures on the helpline and text background, some of them with ease. The time to complete a gesture varied very much. Table 18 shows the task performance times for performing three times the 3U1U gesture.

Table 18: Total time in milliseconds to perform three times the 3U1U gesture

	Total time to perform three times the 3U1U gesture (in milliseconds)		
Participant	Helping Lines	Text Background	Blank Background
P1	26808	30795	23915
P2	33528	28400	25407
P3	5899	35611	23513
P4	160370	38506	74567
P5	25106	33177	97240
P6	10355	9353	15022
P7	12789	60708	71633
P8	26849	32477	10926
P9	23724	114074	56722
Mean	**36159**	**42567**	**44327**
Std. Dev.	47452	29874	31216

For the blank background all users could accomplish the gestures RLRLRL and 3U1U. Five of nine users were even able to perform the most difficult task – RD7DR7 on a blank background. In some of these cases, it initially took quite long to get the gesture done but after the first success, it took not much time to repeat the gesture again.

Table 19 shows the completion time for the different gestures on different backgrounds this means the time from the first stroke to the last stroke in a successfully performed gesture. Neither the background nor the complexity of the gesture has a significant impact on the completion time. The time for the gesture depends only on the number of strokes. The average time required for a stroke was 557 milliseconds. A typical saccade needs 100 to 200 milliseconds (see 3.3.3) and an intentional fixation lasts about 300 milliseconds. This means the time for a stroke is within a plausible range.

Table 19: Average gestures time and standard deviation in milliseconds to perform the three different gestures on three different backgrounds. The data are from nine participants, except RD7DR7 on blank background where only five participants were able to perform the gesture.

Gesture	Helping Lines	Text Background	Blank Background
RLRLRL	3113 (±627)	3089 (±728)	3288 (±810)
3U1U	2222 (±356)	2311 (±443)	2429 (±307)
RD7DR7	3163 (±490)	3563 (±651)	3569 (±520)

The third task recorded the characters produced by the gaze gesture recognition algorithm while surfing in the Internet. The total time for the nine users was 1700 seconds or 28 minutes, resulting in 2737 characters. This results in 1.6 characters per second or about 600 milliseconds for a stroke. Table 20 shows the statistics for the occurrence of characters. The high occurrence of L- and R-strokes is a result of reading horizontal lines of text.

Table 20: Statistics for detected strokes within half an hour of web surfing

Stroke	Occurrences	Percentage	Stroke	Occurrences	Percentage
:	388	14.1%			
1	136	5.0%	D	178	6.5%
3	136	5.0%	U	229	8.3%
7	138	5.0%	L	685	25.0%
9	115	4.2%	R	732	26.7%

This string of characters produced by the gaze gesture recognition was searched for the occurrences of gestures of the first and second task. Table 21 shows the result. The RLRLRL gesture occurs very often because this is the natural eye movement during reading and consequently should not be used for commands in general. The gestures used in the first task (closing the dialog) occurred occasionally and could be used as a gesture if only used within a short lasting context like closing a dialog. The 3U1U and the RD7DR7 gesture were not recognized within half hour of surfing and therefore are promising candidates for gesture commands.

Table 21: Occurrence of the gestures from task 1 and 2 within half an hour of web surfing

Gesture		Gesture		Gesture	
RDLU	0	DRUL	2	RLRLRL	69
DLUR	2	RULD	3	3U1U	0
LURD	1	ULDR	0	RD7DR7	0
URDL	1	LDRU	1		

Lessons learned:

- People are able to perform gaze gestures.
- Gaze gestures are separable from natural eye movements if the gesture is complex enough.
- A gaze gesture takes about 500 milliseconds per stroke.

5.4.2 Experiments – When to Use Gaze Gestures

After the positive results from the first user study the next step was to look for fields of application for this novel type of interaction. The EdgeWrite gestures provide a full alphabet, but the gaze gestures are not adequate for text input. As seen in the user study, a (four-stroke) gesture needs about 2 seconds to enter. Even the standard dwell time method is faster. Typing with the fingers is definitely a more efficient way of inputting text.

A useful application of gaze gestures is the field of accessibility. Because of the robustness of gaze gestures against accuracy problems and immunity against calibration shift a gaze gesture is the perfect way to invoke a recalibration process for the disabled users of eye-tracker systems. It may also be possible to use the gestures for general macro functions within accessibility systems. For example, a gaze gesture could be used to save a document and close the application or to paste content from the clipboard.

One idea was to offer the macro functionality as an extra input modality. This was the reason to implement a software prototype, which is able to recognize a list of gestures and trigger a corresponding command like opening, saving or closing a document (using the WM_APPCOMMAND message of the Windows operating system). When observing people working with documents and applications it is easy to notice that many users use the mouse to select the save option from the menu and return the hand to the keyboard for further text entry. With the use of gaze gestures it is possible to leave the hands on the keyboard – saving the lengthy time for homing and selection – and invoke the save operation with the eyes.

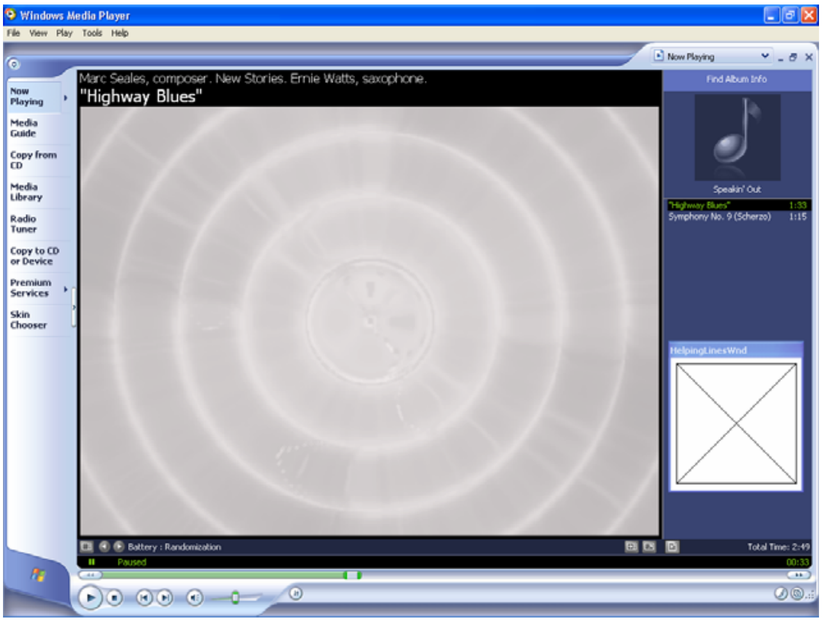
The idea was tested on colleagues who were not involved in this research. They were asked to type something and save the document with the gaze gesture. They were instantly able to perform the operation asked for. The participants reported some fascination with the possibility of directing the computer without touching. However, the participants suggested pressing the short cut 'ctrl-s' to save their document would be quicker. Despite this, they returned to their office and saved the next document by using the mouse. Of course, a key press could also invoke every command invoked by a gaze gesture. It is not worth the effort of an eye-tracking system to do something with the gaze which could be done easily by a key press.

Consequently, gaze gestures are only useful in the case that keyboard and pointing device are out of reach. Mobile interaction is a typical situation where a classical input device is out of reach or inaccessible because the hands are needed for other tasks. Wearable eye trackers allow walking around and in such situation gaze gestures are a possible way to direct computers [Bulling, Roggen, Tröster 2008a].

Another typical situation with the device out of reach is the use of media devices, especially media centre computers; such devices normally come along with a remote control. Gaze gestures are well suited for remote control. The accuracy of an eye tracker is given in visual angle. By principle, this means that the spatial accuracy in millimetres or pixels on the screen gets worse with growing distance. However, gaze gestures on a big grid are insensitive to accuracy problems and seem to be well suited to the situation. In contrast to the research of Vertegaal et al. [Vertegaal, Mamuji, Sohn, Cheng 2005], who used one remote control and eye trackers on several devices to find out for which device the remote control gives commands, the gaze gestures approach does not even need the remote control.

To examine the possible substitution of the remote control with a gaze interface, the software was extended with additional commands for media control such as play, pause, stop, previous track, next track, media channel up and down and volume control (see Figure 62). The system was tested by placing candidates in a distance of one meter away from the display. The maximum distance the ERICA eye-tracker optics is able to focus is one meter. This distance is less than the typical distance to a media device but it is longer than the arms of the candidates so they could not reach the keyboard. Bigger distances are not a general problem, as this requires only minor changes in the optics for a longer focus length. A zoom lens or a higher resolution camera respectively can compensate the effect that the eye appears smaller in the camera picture with bigger distances.

Figure 62: Screenshot of the media player and a window with helping lines to perform the gestures. It turned out that it is more convenient to use the edges of the main display to enter the gesture so that the helping lines are not necessary and even counter-productive.



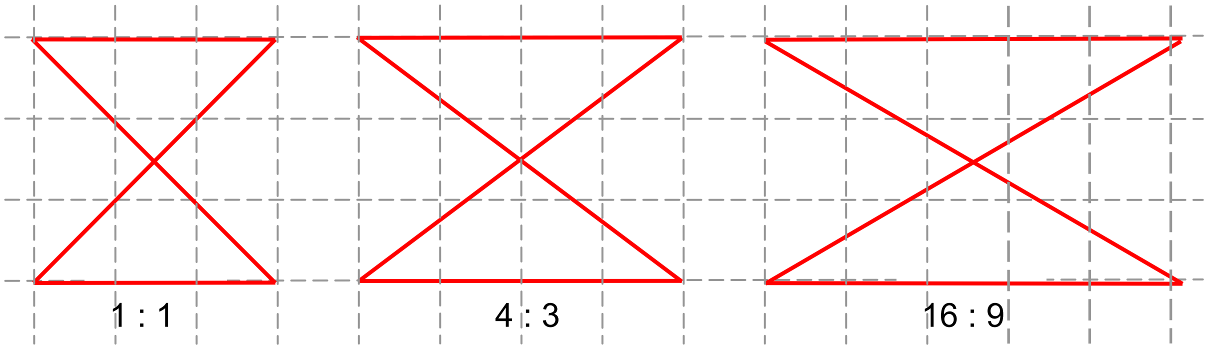
The observations during the experiments were encouraging. The first observation was that people did not need the helping lines offered. The corners of the display window or the screen provide a natural orientation to perform the gestures. The participants had no difficulties to perform the gestures.

The next observation was that people experienced it easier to perform large-scale gestures than small-scale gestures. The data on the statistics of eye movements (see 3.3.3) show that the time needed for a saccade does not increase much if the saccade length gets bigger. All saccades above 5° visual angle last about 120 milliseconds (see Figure 24). The size used for the gesture does not have big influence on the time to perform the gesture.

It also turned out that the grid size of the gesture algorithm is not critical. The prototype detected large scaled gestures reliably even with the grid size settings for small scaled gestures. People seem to perform horizontal and vertical eye movements with high precision. This is a consequence of the anatomy of the eyes – there are two independent muscle pairs for movements in x- and y-direction (see Figure 14).

Another observation was the insensitivity of the gesture recognition to the aspect ratio. The gestures do not have to be in square. An aspect ratio of 4:3 and 16:9 for the corners also work well. See Figure 63 for an illustration.

Figure 63: The gesture algorithm is independent of the aspect ratio.



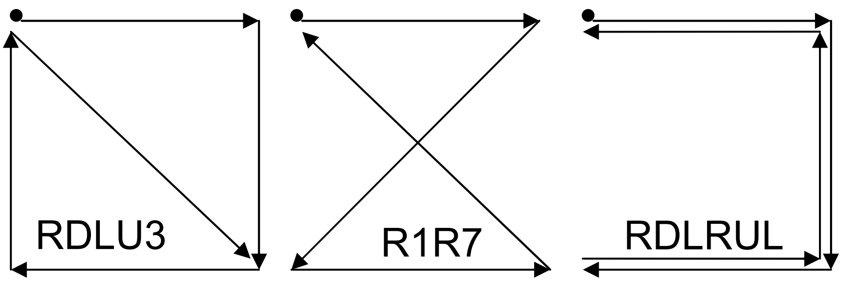
Lessons learned:

- Gaze gestures can provide additional functionalities to a dwell time system used by the disabled.
- Gaze gestures do not make sense for regular users if a keyboard is in reach.
- The corners of the display give good assistance to perform gestures.
- Gaze gestures work well with different aspect ratios for the corners of the display.

5.4.3 Second User Study – Optimizing the Parameters

The second user study was mainly a repetition of the first user study. The main motivation was to find out how to optimize the parameters for the gesture recognition for reliable separation from natural eye movements. For this reason, the grid size was increased from 80 to 250 pixels and the timeout was reduced from 1000 to 700 milliseconds. The first task was the same as in the first user study. The gestures for the second task, performing given gestures, were changed against the first study just to get experiences on some other gestures. Figure 64 shows the three gaze gestures used for the second task.

Figure 64: The three gaze gestures used in the second study.



For the third task the participants were asked to watch a video for three minutes additionally to the three minutes of web surfing. In contrast to the first study the raw eye movement data were recorded and not just the gesture strings. This gave the possibility to calculate the gesture string later using different values for the parameters *s* and *t*.

Eight people, seven male and one female, in the age from 24 to 50 participated in the study. None of them participated in the first user study.

Table 22 shows the average time to get the gestures done (compare with Table 18). Six of eight participants were able to perform all gestures in all background conditions. The other two participants successfully performed six or seven (of nine) gestures respectively. Surprisingly, one person was not able to perform the gesture with helping lines displayed but was able to perform the gesture later on blank background. It turned out that the helping lines are counter-productive. Trying to move the gaze along a line is a difficult task and against the saccadic movement of the eyes.

Table 22: Average time to get gesture done in milliseconds (1/9 of the total time to perform three times the gesture in three background conditions)

Participant	RDLU3	R1R7	RDLRUL
P1	15328	4781	11528
P2	33528	9382	8252
P3	10441	3601	4035
P4	23542	8499	8383
P5	-	22122	-
P6	-	7647	21431
P7	14126	2884	14144
P8	5179	2188	3234
Mean	**17024**	**7638**	**10144**
Std. Dev.	10098	6440	6286

The gesture that caused the most problems was RDLU3. Two participants were not able to perform this gesture three times in three background conditions. The others needed more time than for the other gestures. With the high variance in the data it is not possible to prove this, but it seems that people prefer closed gestures, especially for repetitive input.

One person had difficulties with the reduced timeout. The analysis of the raw eye movement data revealed that this person could have performed the gestures instantly with a timeout of 1000 milliseconds but needed minutes to get it done with the 700 milliseconds timeout.

The evaluation of the eye movement recordings during surfing the Internet and watching a video will follow in section 5.6.

Lessons learned:

- The results of the first user study are reproducible.
- A timeout of 700 milliseconds may be too short for untrained users
- Do not instruct the people to follow a line with the gaze as helping lines do not help. Instruct them to visit the corners of the display with the gaze in a certain order.
- People seem to prefer closed gestures.

5.4.4 Mobile Phone User Study

Most mobile phones sold today have a camera already built-in. Even more, many modern phones have a secondary camera on the front side as well which is included for video conferencing. Eye-tracking technology for interaction with mobile phones is not yet available. One reason is the lack of processing power to handle video streams on these devices in real-time. Looking retrospectively at the development of mobile devices, it is noticeable that their processing power, as well as the quality of their components, increases steadily. Thus, it is only a matter of time until eye-tracking technology can be offered within these devices by software at virtually no extra costs.

As small mobile devices do not have elaborated input facilities, like a big keyboard for two-handed typing, and they mostly do not have any pointing device like a mouse or a touchpad, eye gaze could be an additional input modality.

One problem of eye tracking for mobile phones is outdoor use because of varying light conditions and saturation effects by bright sun light, which makes it difficult to detect reliably the pupil and the glint within a camera picture. Approaches with differential pictures and infrared illumination synchronized with the frame rate of the video camera or the use of polarized light are promising [Morimoto, Koons, Amir, Flickner 1999]. Therewith, eye tracking for at least most light conditions, except the extremes, should be achievable.

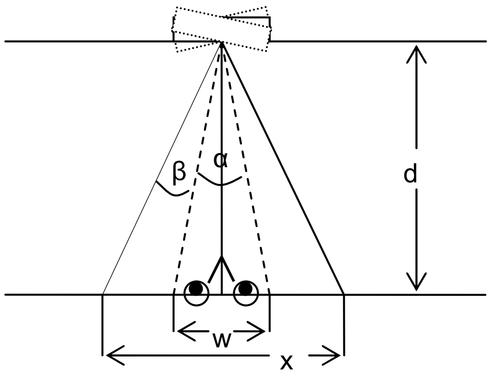
Eye trackers which allow free movement of the head also depend on head tracking. Commercially available eye trackers for free head movements are desktop systems where the camera stays in a stable position. For a mobile phone the situation is different. The camera is built into the device, which is held by the user, and naturally the hand moves constantly. Especially angular movements cause very quick changes in the camera picture.

To find the camera resolution required to achieve the same accuracy as a standard eye tracker, some experiments were done on the angle at which a mobile phone is held during standard operation. A volunteer performed some standard tasks, such as searching an entry in the phone list and writing an SMS on his mobile phone, for measuring the angles at which he held the mobile phone. The observation estimated the tilt angle to be about 20° and a roll angle of about 10°.

Figure 65: Measurement of the angles holding a mobile phone

The camera of the ERICA eye tracker has a video stream resolution of 640 × 480 pixels and projects an area of about 6 cm × 4.5 cm. This means approximately 100 pixels for 1 cm. The following calculation demands that the camera resolution of the handheld device has the same resolution. With a face width *w* of 15 cm, a typical distance *d* of 40 cm and a roll angle β of 10° the necessary camera resolution in horizontal direction results to approximately 3000 pixels (see Figure 66). Already today such camera resolutions are available in mobile phones, at least for still images.

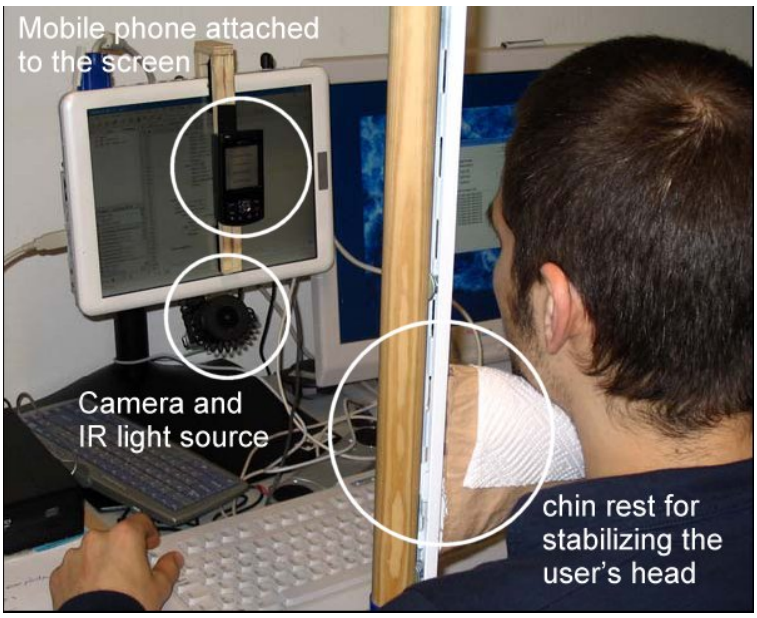
Figure 66: Calculation of the resolution needed for the eye tracker camera of a mobile device



$$\tan\frac{\alpha}{2} = \frac{w}{2d} = \frac{15}{80} \Rightarrow \frac{\alpha}{2} \approx 10°$$

$$\tan\left(\frac{\alpha}{2} + \beta\right) = \frac{x}{2d} \Rightarrow x = 2d \cdot \tan\left(\frac{\alpha}{2} + \beta\right) \approx 2 \cdot 40 \cdot 0.36cm \approx 3000\,px$$

Figure 67: The mobile phone mounted in front of the eye tracker system



To conduct a user study a mobile phone (Nokia N80) was fixed in front of the eye tracker (see Figure 67). The eye tracker delivered the gaze data to the mobile phone using a WLAN connection. A Java (JavaME) program running on the mobile phone processed the gaze data and triggered actions on the mobile phone. The complete study [Drewes, De Luca, Schmidt 2007] had two tasks – selecting a name from the phone book list with the dwell time method and starting and closing the internet browser application with gaze gestures. This text reports only on the task with the gaze gestures to stay in the context of this chapter.

Eight participants, aged from 23 to 50 years, two female and six male, took part in the study. In the gesture task the participants were asked to perform the R1R7 gesture to start and the D9D7 gesture to close the internet browser of the mobile phone. The participants were instructed to use the corners of the mobile phone display as fixation points for the gesture. The grid size for the gesture detection was 60% of the display width which translates to approximately 80 pixels on the eye tracker or a visual angle of 2.2°. Because of the second user study, where one person had problems with the short timeout, the time parameter was increased to 1500 milliseconds.

All participants were able to perform the gestures to open and close the internet browser this way. Table 23 shows the times of all participants needed to perform the gestures (from first stroke to last stroke). Again (see Table 19), the data show an average time per stroke of 400 to 500 milliseconds.

Table 23: Performance times in milliseconds of eight participants for both gaze gestures

Participant	R1R7 milliseconds	D9D7 milliseconds
P1	1142	1853
P2	1783	1392
P3	2474	1532
P4	1332	1572
P5	1572	1142
P6	3435	1932
P7	1953	2013
P8	2133	1522
Mean	**1978**	**1620**
Std. dev.	728	294

The discussion whether a gesture command occurs while normal operation of the mobile phone and therefore may accidentally invoke a command follows in section 5.6.

Lessons learned:

- There is no general problem to perform gaze gestures on small displays.

5.4.5 PIN-Entry User Study

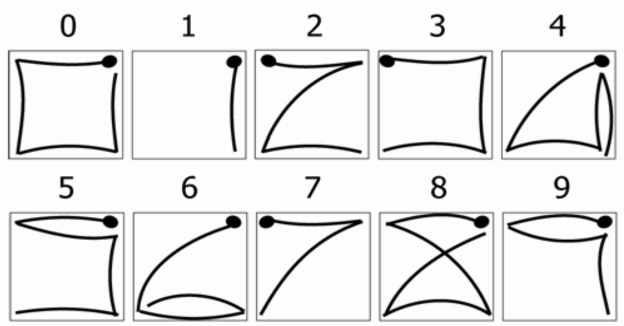
Personal identification numbers (PINs) are one of the most common ways of electronic authentication these days and they are used in a wide variety of applications. PINs have to stay a personal secret as revealing the PIN can result in a personal loss. In the case of drawing cash from an ATM, which mostly happens in public space, there is a high danger that somebody observes the entry of the PIN. To avoid such shoulder surfing a less observable PIN entry method is desirable. Assuming that observing and interpreting eye movements is much more difficult than observing the finger movements on a number pad leads to the idea to use eye gaze for a more secure PIN entry. With this motivation Kumar et al. did a user study with the title “Reducing Shoulder-surfing by Using Gaze-based Password Entry” [Kumar, Garfinkel, Boneh, Winograd 2007]. The study presented here used the same gaze-based input techniques, but additionally researched the use of gaze gestures as a further input technique. The study was done within the project thesis of Roman Weiss and published in [De Luca, Weiss, Drewes 2007].

The PIN-entry user study used three different gaze-based techniques for PIN entry. The first and second method used gaze pointing to enter the PIN on a number pad displayed on the screen (see Figure 68). The first method used a dwell time of 800 milliseconds and the second method used a button, which had to be pressed when looking at the correct number on the number pad display. The second method was introduced as hardware key or gaze key, but called look & shoot method in the context of the user study as this name is self-explaining and got high acceptance by the participants. The prototype displays an ATM number pad of an overall size of 730 × 450 pixels. Each button has a size of 180 × 90 pixels which is about 5° by 3° visual angle and hence clearly above the typical eye tracker accuracy of ± 0.5°. To retain the security benefit there was no feedback except asterisks to indicate successful entry of a digit.

Figure 68: The user study setting (left) and the layout of the number pad for PIN entry

The third method used gaze gestures to enter the digits. As there is no need for remote entry of PINs it is not an obstacle to use a gesture key. With a gesture key it is not necessary to separate natural eye movements from gestures and this allows more freedom in the design of a gesture alphabet and consequently allows a more intuitive alphabet. The alphabet used (see Figure 69) is the one introduced by Wobbrock et al. [Wobbrock, Myers, Kembel 2003]. Without a gesture key the digit 9 and 5 would not be distinguishable.

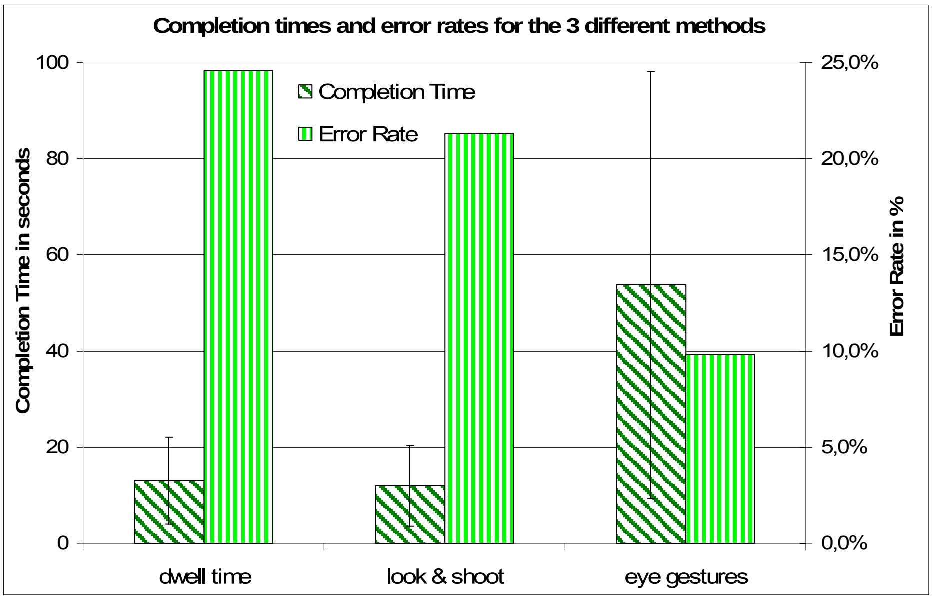
Figure 69: The digit gestures used for the prototype



In the user study, 21 volunteers completed the three different PIN entry tasks and afterwards answered a questionnaire. Seven of the participants were female and all participants were aged between 22 and 37 years. Five of them had already used an eye tracker before, but not on a regular basis.

Figure 70 shows the completion time and error rate for the three different entry methods. The evaluation of the data using analysis of variance (ANOVA) showed no significant advantage regarding execution times for the look & shoot or dwell time method. Using the look & shoot method a four digit PIN entry took the subjects 12 seconds in average whereas a PIN entered using dwell time took 13 seconds. The error probability also showed no significant difference. Using dwell time, 15 of the entered 63 PINs were faulty (23.8%), using look & shoot 13 entered PINs contained errors (20.6%).

Figure 70: Completion times and error rates for three different methods of eye gaze interaction



The results of the user study show that eye gaze interaction is a suitable method for PIN entry and partially confirm the results found by Kumar et al. [Kumar, Garfinkel, Boneh, Winograd 2007] regarding the dwell time and look & shoot techniques. Entering PIN numbers with the gaze gesture method took much longer than using the 'classic' methods (an average of 54 seconds per PIN entry was measured) but was also much more robust against errors than the methods described above. Only six of the entered PINs using gestures were erroneous (9.5%). Using a binomial test shows a significant enhancement of the error rate ($p < 0.008$).

The gaze gesture method is less intuitive than the classic methods as some subjects initially had problems to produce recognizable gestures. Furthermore, the gesture alphabet was unknown to all participants. This explains the big difference in time for completing the PIN entry task. The participants spent much time for looking at the sheet with the gestures for the single digits. As already shown in the previous user studies, a stroke within a gaze gesture needs about 400 to 500 milliseconds (a little bit more than 100 milliseconds for the saccade and around 300 milliseconds for the fixation) and entering a digit with four strokes takes about 2 seconds. A four-digit PIN with one second break between the inputs of the digits will last around 10 seconds. Indeed, there were participants in the study who entered the PIN correctly within 14 seconds. It needs a further study to find out whether all users can achieve this time once they are trained for gaze gesture input.

In addition to the absence of a calibration process, the big advantage of the gaze gesture method is its robustness against input errors. Due to the abandonment of feedback for enhanced security, each wrong gaze leads to an incorrect PIN entry when using the dwell time or look & shoot method. This leads to high error rates for these methods. When using the gestures, a wrong gaze leads most probably to an unrecognizable gesture and not to an entry of a wrong digit. For gaze gestures the errors occur one level below the digit entry, i.e. at the gesture recognition level.

The main reason why a gesture performed by a user is not recognized by the system is a lack of exactness in the hand-eye coordination. As a button has to be pressed and held while performing the gesture, often an additional stroke was detected directly before or after the proper gesture. The reaction time for the finger, typically 300 ms is long compared to the time of a saccade, typically 100 ms. These unintended upstrokes or tails could be filtered out by the algorithm and improve the recognition rate.

Figure 71: Results of the questionnaire on a Likert scale from 1 to 5

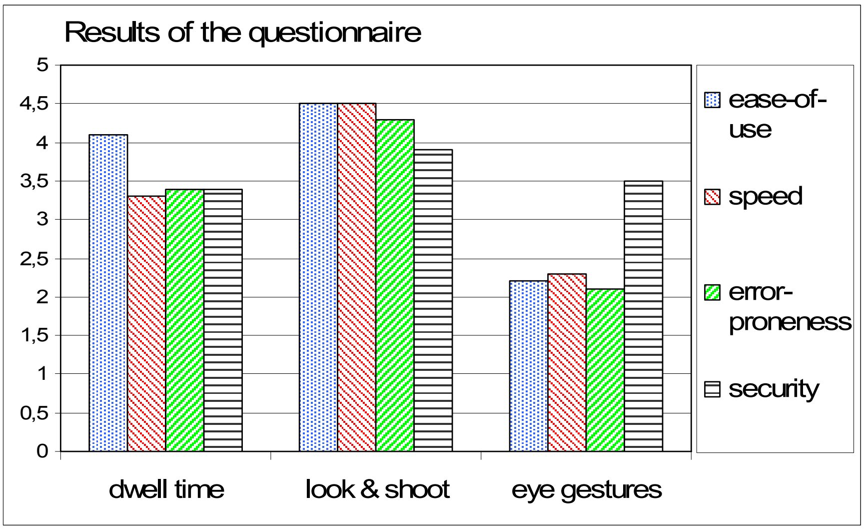


Figure 71 presents the results of the questionnaire. The users prefer the look & shoot method – it is easier and quicker but also more error-prone than the gaze gestures.

Lessons learned:

- The use of gaze gestures protects from accidental input of wrong digits.

5.5 The Gaze Gesture Alphabet

After showing that people are able to perform gaze gestures the question is which gestures to use. The gaze gestures should be easy to perform, easy to memorize, and separable from natural eye movements without the need of a gesture key. In this context the PIN entry presented in the last section is an exception as it uses a gesture key and provides its own special purpose alphabet which resembles digits. In general, gaze gestures are not suited well for text entry because eye-typing using the dwell time method is faster and more intuitive. The gaze gesture alphabet presented here shall serve as a set of commands for remote control. The feasibility of this alphabet is not proven, but it is based on observations from the user studies and reasonable assumptions. Therefore, this alphabet can serve as a starting point for further research. The focus of this work is to find a reasonable alphabet which is separable from natural eye movements.

The notation with the eight directions originating from the mouse gestures gives freedom to define a big number of possible gestures. The possible use of the four corners of the display as helping points proposes to restrict the gestures to the gestures of Edgewrite. A ‘stairway-shaped’ gesture like RDRDRD is difficult to perform without landmarks in the visual field for orientation while visiting the four corners of the display with the gaze seems to be easy. The restriction to the Edgewrite gestures make sure that the gaze has not to leave the display when performing the gesture.

The inventors of the EdgeWrite gestures defined a huge gesture alphabet with multiple gestures for each letter and digit of the Latin alphabet. In the context of gaze gestures it is not possible to use this alphabet. The reason lies in the absence of an extra modality (e.g. gesture key or touch signal of a pen) which could tell the start and end of the gesture. The gesture alphabet of EdgeWrite has gestures which are a part of another gesture, for example the ‘1’ is a single stroke and it is a part of the ‘3’ (see Figure 57). In order to avoid ambiguities in the recognition, any gesture of a gaze gesture alphabet must not be a part of another gesture in the alphabet. An easy way to achieve this is to use gestures with the same number of strokes. Such an alphabet loses the similarity of the gestures to digits and letters. However, as the gestures are used for commands and not for text entry, it does not matter.

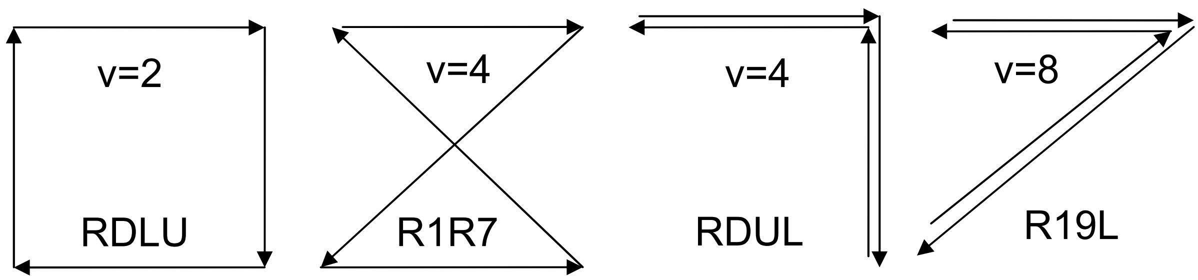
From observations during the second user study (see 5.4.3) and the comments given by participants it seems that users prefer gestures which end at the position where it started. This type of gestures makes it easy to repeat the gesture in the case it was not detected or in the case where repetitive input is wanted, e.g. increasing the volume or switching to the next channel in a media application.

The gestures of the gaze gesture alphabet should also be separable from natural eye movements. Four stroke gestures seem to be a good choice for gaze gestures and therefore worth being tested. Gestures with fewer strokes bear the danger of occurrence within the natural eye movements. Gestures with more than four strokes are complex and hard to memorize according to the opinion of some participants in the user study. As people are able to memorize the alphabet for reading and writing they should be able to learn even a complex gaze gesture alphabet. To memorize the letter A we do not learn an abstract stroke definition but a shape and a motion sequence for writing. Similarly, it is not necessary to memorize the notation for the gaze gestures. RDLU is the definition of a gesture suitable for the detection algorithm but we memorize it as a square performed clockwise. It is also possible to construct and memorize complex gestures by concatenation of simple ones. It is possible to memorize the RD7DR7 gesture as the upper right triangle followed by the lower left triangle. However, as long as there is no reason for complex gestures, they should be as simple as possible. One reason for the need of more complex gestures is the fact that the number of four-stroke gestures is limited to 18 gestures by the demands given above and the following further demand.

A gesture that ends at the starting point should not get a different meaning when the same pattern starts at another corner. If the RDLU gesture has an assigned meaning, the gestures DLUR, LURD, and URDL should have the same meaning or no meaning. Otherwise, when performing the RDLU gesture repetitively and one stroke is not detected correctly, the gesture results in an unwanted action. For an example with two-stroke gestures, let us assume that the gesture RL means 'volume up' and LR means 'volume down'. Then the gesture to turn the volume down three steps is LRLRLR. In the case that the first stroke was not recognized correctly, because the gesture was not performed well or because of a temporary malfunction of the eye tracker (changing light conditions, hand obstructed the camera view), the detected gesture could be L:RLRLR, 7RLRLR or RLRLR. These gestures would turn the volume up two steps – the opposite of the user's intention. One possibility to avoid such problems with concatenated gestures is demanding a rest in the eye movements between two gestures, i.e. the output of a colon from the detection algorithm. However, this slows down the execution time of concatenated gestures.

Gestures which need only two points, typically RLRL, occur too often in natural movement. With a systematic construction of possible gaze gestures which use at least three points, it is possible to identify four types of four-stroke gestures as given in Figure 72. The total number of different gesture commands by mirroring or rotation is 18 (2 + 4 + 4 + 8). This is about the typical amount of buttons on a remote control for a TV set. Allowing to start the gesture at any corner leads to 72 different gesture strings.

Figure 72: Four types of four stroke gaze gestures and their number of variations *v* by rotation and mirroring



5.6 Separation of Gaze Gestures from Natural Eye Movements

To use the gaze gestures as a remote control without an extra key the algorithm has to filter out gesture commands from natural eye movements. It is crucial for the application of gaze gestures that unintended gaze commands do not occur within natural eye movements. For this reason, natural eye movements in three different situations – surfing the internet, watching a movie (see 5.4.3), and entering an SMS on a mobile phone – were used for analysis.

The gesture recognition algorithm has two parameters, grid size *s* and timeout *t* which allow adjustment. Calculations done with the recorded eye movements show that the resulting gesture string does not depend much on the timeout parameter. The eye does not rest long in the same position. An intentional fixation lasts about 300 milliseconds and a saccade about 100 milliseconds. As long as the timeout parameter is above the sum of these values, the resulting gesture string does not change much when changing the timeout value (see Table 24). Consequently, the timeout parameter is not very good to improve the separation of gestures from natural movements. As one participant of the second user study had difficulties with a timeout of 700 milliseconds, a value of 1000 milliseconds or even above should be used.

In contrast to the timeout parameter, the change of the grid size has big influence on the resulting gesture string. Most saccades in natural eye movements are below 50 pixels (≈ 1.5° visual angle) which is easy to see in Figure 23 and explainable by the size of the fovea. Bigger saccades occur occasionally only. In this context the timeout parameter shows its importance. Although the timeout value is not critical, it is necessary to filter single bigger saccades. When using a big grid size the timeout parameter produces colons in the case the eye movements were small and did not leave the area of a grid cell.

Table 24: Translating the same eye movements (surfing the internet) into a gesture string using different values for the grid size *s* and timeout *t*

Parameters	Resulting Gesture String
s=80 *t*=1000	:3LUD::7R1L9:73LR:73LR:7379RL:U:D:U3:LU::RL::R13U::LR:R:73:73D:73:LRLRLR7 373DU7LD:RUL13L:R:RL:RL:LRL:R7L3L9R1UR3DR::7:URLRLRLRDLR7U:R3RL:LR
s=80 *t*=700	:3LU:D::7R:1L9:7:3LR::73L:R::737:9:RL:U:D:U3:LU::RL::R13U::LR:R:73:73D:73:LR:L RLR7373DU7LD:RUL1:3L:R:RL:RL:LRL::R:7L:3:L9R:1UR3DR::7:URLRL:RLRDLR7U
s=250 *t*=1000	:3L:::7R:1U:U3::7RL:UDU:L:::::::RD:R::::73::73::::L::L:::R::L:R:L:::R7:R:RL:DR:L:::::U: R:R::LR:L:RL:UD:R:::L::LR:L:3:L:::::::D:RL:RLRL::DRL::RLRLRLRLR:LRLRL:RLR:7:::
s=250 *t*=700	:3L::::7R:1:U:U:3:::7RL:::UDU::L::::::::RD::R::::::73::73:::::L:::L:::::R::L:R::L::::R7::R::R:L :D:R:L::::::::U:R:R:::LR:::L::RL::UD:R:::::L::LR:L::3::L:::::::::D::RL::RL:RL:::D:RL::RL:RL
s=400 *t*=1000	:3:::L:D::U3:::::U::L:::::::RD:::::73::73::::::L::::RL::::::U::RL:R:::::::7:R::::::L:R:7:R::L::LR:: :::L::R:::::RL::R::L:RL::::::RL:R::L:::RL:RL:RL::RL:RL:R::L:::RL:

Table 25 shows the occurrence of the gestures demanded in the user studies when using different values for the timeout and the grid size. Table 26 shows the occurrences of any of the 72 gestures suggested for an alphabet.

Table 25: The occurrence of gestures used in the studies while surfing the internet and watching a video. The table shows the results from using different parameters for the recognition.

	Occurrence of Gestures while **surfing the internet** (1/2 h)				Occurrence of Gestures while **watching video** (1/2 h)			
Gesture	s=80 t=1000	s=80 t=700	s=250 t=1000	s=250 t=700	s=80 t=1000	s=80 t=700	s=250 t=1000	s=250 t=700
RDLU	3	2	0	0	0	0	0	0
DLUR	4	1	0	0	1	1	0	0
LURD	1	0	1	0	0	0	0	0
URDL	3	2	0	0	1	1	0	0
DRUL	6	4	0	0	1	0	0	0
RULD	3	1	0	0	0	0	0	0
ULDR	2	0	0	0	0	0	0	0
LDRU	1	1	0	0	0	0	0	0
RLRLRL	41	22	5	0	7	3	5	3
3U1U	0	0	0	0	0	0	0	0
RD7DR7	0	0	0	0	0	0	0	0
RDLU3	0	0	0	0	0	0	0	0
R1R7	2	1	0	0	0	0	0	0
RDLRUL	1	1	0	0	0	0	0	0
# char	3447	4022	2651	3391	3182	3690	2326	2873
char / s	1.62	1.89	1.25	1.59	1.59	1.85	1.16	1.44

Table 26: Occurrences of the 72 closed gesture strings using different parameters *s* and *t* for the recognition

s pixels	t milliseconds	# Occurrences Watching a video	# Occurrences Surfing the internet
80	1000	82	136
80	700	47	77
250	1000	15	21
250	700	7	10
400	1000	0	4
500	1000	0	1
500	700	0	1
600	1000	0	0

The answer to how to adjust the parameters so that they eye gestures can be distinguished from natural eye movements is surprisingly simple. When surfing in the Internet or watching a video, the gaze stays within the borders of the display. Saccades with a length close to the dimension of the display occur rather seldom and a series of four or even six saccades of this size almost never happens. Choosing a grid size close to the smaller dimension (length or width) of the display will ensure that natural eye movements are not interpreted as gestures.

The result of the studies recommend the use of four-stroke gestures with a grid size of 80 % of the smaller dimension of the display and a timeout of 1000 ms for normal-sized displays (desktop and stationary media systems).

With small displays, gaze gesture interaction is different. A dataset that was created during the work for [Holleis, Otto, Hussmann, Schmidt 2007], in which Holleis et al. were creating a keystroke level model for advanced mobile phone interaction, was used to find out which gaze gestures appear in standard mobile phone interactions. For their research, they recorded the gaze coordinates of 11 persons when writing a SMS text message. Each participant performed these tasks with his or her own mobile phone. The data set represents 28 minutes of recorded eye movements. Table 27 show translations of the same eye movement recording to gestures using different parameters. With an average size of 120 pixels for the mobile phones display the parameter values 60, 80 and 100 for s relate to 50%, 66%, and 80% of the display's size.

Table 27: Translating eye movements (entering a SMS) into a gesture string using different parameters

Params	Resulting Gesture String
s=60 t=1500	3DU:DUDU:3DLUDU3ULD73DLU31UD1U:DR73RU:UDUDRLUDU:1UDUD7DU19:DRU: DU::DU:U3D7:DR7:3UDRUDUDU:DUD1U:DRU:DU:1RLU:DRL:U31UDU::D9:UDUD9DU
s=80 t=1500	3DU:DUDU:DUDU3UD731UR1UDU:D73:U:DUDRLUDU:DUDUDUDUD9:DRU:D:DU::D U:U3D7:D:U:DUDRUDUDU:DUDU:DRU:DU:D:U:DRL:U31UDU::D9:UDUDUDUDU3L:9L
s=100 t=1000	3DU:DUDU::DU:DUD:U:DU31U:R1UDU::D:7D::U:D:UDR:UDU::DUDUDUDUDU:D:RU:D :U:::DU::U:DU::D:U::DU:D:U:DU:DU::DUDU::D:U:DU::D:U::DRL:URLUDU:::DU:DUDU:D

When comparing the strings of Table 27 with the strings in Table 24 (surfing the Internet), it is obvious that entering a SMS produces many up and down movements of the eye, while surfing the Internet produces many left and right movements. Surfing the Internet requires reading horizontal lines of text while entering an SMS causes the gaze to switch from the display to the number pad below and back.

Analyzing the strings produced from the gesture recognition for entering a SMS shows that the 72 closed four-stroke gestures occur quite often. For this reason they should not be used as commands. Table 28 summarizes the results of the analysis.

Table 28: Occurrences of the 72 closed gestures when entering a SMS using different parameters *s* and *t* for the recognition

s pixels	t milliseconds	# Occurrences Entering a SMS
60	1500	164
80	1500	97
100	1000	40

The six-stroke gestures RD7DR7 and RDLRUL and the five-stroke gesture RDLU3 did not occur for any of the tested parameter combinations. It may be possible to assign actions to gestures that occasionally occur in natural eye movements as long as the action is only applicable for short periods or in specific application contexts. Otherwise, it seems that it needs six-stroke gestures to separate gaze gestures performed on small displays from natural eye movements. As some of the participants in the user studies complained about the complexity of six-stroke gestures, the acceptance of such gestures is questionable.

Lessons learned:

- Short timeouts do not help much for the separation of gestures from natural eye movements but make it difficult for some users to perform the gesture.
- Nevertheless, the timeout parameter is necessary to interrupt the gesture recognition after a sequence of small saccades.
- On big displays (desktop, laptop) the gaze stays within the display. Using big grid sizes for the recognition algorithm allows separation of four-stroke gaze gestures from natural eye movements.
- On small displays (mobile phones) the gaze leaves the display. Reliable separation of gaze gestures from natural movements requires more complex gaze gestures, i.e. six-stroke gestures.

5.7 Summary of the Results for Gaze Gestures

The user studies showed that people are able to perform gaze gestures. Helping lines are not helpful and should be replaced by helping points which displays naturally provide with the four corners. Gaze gesture detection works over some distance and can serve as a remote control. Gaze gestures can easily be separated from natural eye movements if they are performed on a big scale. Performing gaze gestures on a big scale is easy on large displays where the corners provide helping points. The situation is different for small displays like those of mobile phones, especially when buttons are arranged around the display. As it does not feel natural to perform a big gaze gesture with a small handheld device, the way to separate the gaze gestures from natural eye movements is the use of more complex gestures, for example with six strokes. The other solution for small displays is the use of gaze gestures only within a context. If the gesture detection is only active when the device expects an answer, an accidentally performed gaze gesture will not disturb general interaction.

One stroke of a gaze gesture needs about 500 milliseconds. With at least four strokes for the gesture, it is obvious that gaze gestures are not well suited for text entry as they are slower than other gaze-based methods. The argumentation that such form of text entry can make more fun [Wobbrock, Rubinstein, Sawyer, Duchowski 2007] do not correspond to the observations made during the user studies.

5.8 Conclusions for Gaze Gestures

Gaze gestures are an alternative and interesting approach for eye gaze interaction. The absence of a calibration procedure makes it possible to build eye-gaze interfaces for instant use by different people.

Gestures in general are not very intuitive and the user has to acquire skills i.e. learn an alphabet and its meaning, before she or he can use it. It does not look like it is very difficult to learn the motor skills to perform gaze gestures. The number of participants in the study was not big enough to prove that everybody is able to perform gaze gestures but it clearly shows that there are people who can do it instantly and with ease. The eyes have one pair of muscles to control the x-direction and another for the y-direction. These suit well for the horizontal, vertical, and diagonal eye movements needed for the gaze gestures. In contrast, gestures for the hands expect straight lines while the natural movements are curved lines. The reason why it took some people much effort to get a gesture done, is partly because of a bad explanation using the term 'line' and partly because of nervousness of the participants. To look at the corners of a rectangle (corners of the display) in a certain order is not very difficult.

The analysis of natural eye movements recorded from different tasks and people indicates that gestures are reliably separable from natural eye movements when using reasonable parameters for grid size and timeout. Consequently, there is no need for an additional input modality like a gesture key. The corners of the display provide a perfect orientation for the gaze to perform a gesture. The only critical situation is a small display with buttons on the sides and below or above the display.

A main problem of eye tracking is the low accuracy. As the gaze gestures are movements on a big scale this problem dissolves. The accuracy is an angular value and for hitting a target with the gaze the situation gets worse on bigger distances. This is not the case for the gaze gestures because they are performed on an angular basis. The maximum distance of gaze gesture detection is a question of camera resolution and zoom factor of the lens. For this reason gaze gestures can serve as a remote control.

One field for application of gaze gestures is in highly hygienic environments (nothing to be touched) where an interface for instant use by different people (no calibration) is needed. Gaze gestures provide more complex control than possible with an eye contact sensor or by other touch free techniques such as a capacity sensor or a photoelectric barrier.

The question whether gaze gestures could serve as a remote control for the TV set and become an input technology for the masses and not only for special purposes is interesting and still open. The big advantage of remote control by gaze gestures is the absence of a control device – nothing to look for and no batteries to recharge.

To answer the open questions it would be nice to have a small and cheap prototype of an eye-gaze gesture recognizer. This would make it possible to study the gaze gestures outside the lab in real situations.

Finally, the gaze gesture algorithm has the ability to translate eye movements into character strings. This could be useful for recognizing the user's activity. The raw movement data allow the calculation of mean fixation time, mean saccade length or saccades per second, but make it hard to derive other meaningful numeric values. The representation of eye movements in form of a character string allows the application of string pattern matching algorithms. However recognizing the user's activity from eye movements is the topic of the next chapter.

6 Eye Gaze as Context Information

The two previous chapters treated eye-gaze input as an intentional action of the user. A further possibility is to utilize the user's eye gaze for implicit human computer interaction called iHCI by Schmidt [Schmidt 2002, p. 189ff]. The computer uses the information from the eye tracker to analyze the situation and activity of the user and the computer adapts its behaviour according to the current user's state. The idea to take the situation of the user into account dates back to 1997 [Hull, Neaves, Bedford-Roberts 1997]. Since that time the consideration of the user's environment and situation is a topic of HCI research called context awareness.

This chapter gives an overview on various approaches to use eye gaze as context information. As this is a wide field the research work presented here focuses on two topics. The first topic reports on the development of a usability-testing tool which records interaction activity together with eye movements. Such a tool allows analysis how eye gaze and manual interaction depend on each other. It also provides a technological basis to realize applications that use eye gaze as context information especially in the context of web browsing. The second topic deals with the implementation of reading detection algorithms and a possible application within e-learning environments.

6.1 Eye Gaze and Context Awareness

The most obvious way to utilize the eye gaze as context information is the interpretation of the eye gaze as an indicator of attention. In most situations we look at the object that we pay attention to. This may sound trivial but attention is very powerful context information which can provide real benefit for a user. A display of an electronic device like a desktop computer or a mobile phone provides information for the user. However, there is no need to display the information if the user does not look at it. Nowadays systems which are normally not aware of the user's attention may display an important message while the user is not present, and when the user is back the information may be overwritten by the next message. An attention-aware system, realized with an eye tracker, may give the user an overview on what had happened when the user's attention is back at the system.

Logging the user's attention also provides reliable statistics on how much time she or he spent on a particular document. The time the document was displayed is not reliable as the user can open the document and leave the place to fetch a coffee. The statistics on how much time spent on which document can be very useful. In a working environment, where somebody works on different projects, it is necessary to calculate the costs of each project, which finally means to know how much time was spent on each project. Such statistics can also be useful for electronic learning (see 6.5).

Although the accuracy of eye trackers is low compared to a mouse device, it is good enough to identify the user's attention within the screen. As explained already (see 3.2) an eye tracker can deliver the position of the gaze with the accuracy of the size of a thumbnail, which is about the size of an icon on the screen. This means the system knows which object on the screen the user is looking at. The system can utilize this knowledge to assist the user. For example, the system can provide additional information or options for the object the user looks at. In a simple form, this would be something like the tool tips or balloon help, which nowadays appear when the mouse pointer is on the object. However, it could also work the other way round for an annotated schematics where the number of annotations is so big that there is not enough space to display them all. In this case, the system could provide only the annotations for the part of the schematics where the gaze is on. Another example would be a multimedia encyclopaedia with animations. The explanation of a motor provides text and a picture and when the gaze moves onto the picture of the motor, an animation starts. When the gaze moves back to the text the animation automatically stops for not to disturb the reading process.

The applications of eye gaze as context information mentioned so far only used the attention in general or to a particular object. However, it is also possible to guess about user's activity from the data delivered from the eye tracker. The analysis of the data could tell whether the user gazes at a particular object or is searching for something on the screen or is reading a text. As reading is a very important activity it will be discussed in detail later in this chapter (see 6.4). An application using the information on the user's activity can be a smart system, which does not bother the user with unimportant information like the availability of a software update while the user is reading. The other way round, in the case of critical information in the message, the system can display messages that do not disappear until read.

Beyond the activity of the user is the state of mind and the physical condition. Eye trackers can measure how quick the eyes move or how fast the size of the pupil adapts to changing light conditions. Such information allows estimating the physical fitness of the user. Typical applications in this field are car assistance systems that warn the driver before she or he falls asleep.

6.2 Related Work for Eye Gaze as Context Information

Eye gaze as context in the sense of attention always means attention to an object and finally can be seen as pointing. It is definitely pointing whenever the focus lies on high accuracy of eye-gaze direction. However, there is no sharp border. Fono and Vertegaal presented a system where the gaze controls the focus selection of windows within a graphical user interface [Fono, Vertegaal 2005]. The reason to mention their work in this section and not in the related work on eye gaze as pointing is that windows normally have sizes above the accuracy of eye trackers and accuracy is not the main issue. Fono and Vertegaal showed that there is a big speed advantage for activation of windows by gaze compared to manual activation. The researchers around Vertegaal published several papers on the use of context information retrieved from eye tracking. In [Vertegaal, Mamuji, Sohn, Cheng 2005] they introduced Media EyePliances, where a single remote control can control several media devices. The selection of the device to control happens by looking at it. For this they augmented the devices with a simple form of an eye tracker, called Eye Contact Sensor ECS (see Figure 11). The ECS uses the corneal reflection method and is calibration-free. To achieve this, a set of infrared LEDs is mounted on-axis around an infrared camera. When the camera delivers a picture with a glint in the pupil centre it means that the onlooker is looking directly towards the camera. Therefore, the ECS does not deliver coordinates, but signals eye contact only. In a further paper from the same year [Dickie, Vertegaal, Sohn, Cheng 2005] Vertegaal et al. used the ECS in combination with mobile devices. It is typical for mobile devices that they do not get full attention all the time, as the user of the mobile device has to pay some attention to her or his surroundings. With two applications, seeTXT and seeTV, they demonstrate how to use the eye gaze to detect attention and how to use this context information to control the device. The seeTV application is a video player, which automatically pauses when the user is not looking at it. The seeTXT is a reading application, which advances text only when the user is looking.

Other research papers of Vertegaal et al. focus on the social aspects of eye gaze. In [Vertegaal, Dickie, Sohn, Flickner 2002] they present an attentive cell phone, which can detect whether the user is in a face-to-face communication with somebody else and use this information to employ social rules for interruption. The attentive cell phone uses speech detection and an ECS worn by the user. In [Shell, Vertegaal, Cheng, Skaburskis, Sohn, Stewart, Aoudeh, Dickie 2004] Vertegaal et al. go a step further and use attention to open sociable windows of interaction. They integrated the ECS into glasses to have a new wearable input device, which they call ECSGlasses. The ECSGlasses are an interesting hardware for discussing attentive user interfaces.

Most activities of humans involve eye movements related to the activity. Land [Land 2006] describes eye movements for daily life activities such as reading, typing, looking at pictures, drawing, driving, playing table tennis, and making tea. Retrieving context information means to go the other way round and conclude from eye movements to the activity. Using such an approach Iqbal and Bailey measured eye-gaze patterns to identify user tasks [Iqbal, Bailey 2004]. Their aim was to develop an attention manager to mitigate disruptive effects in the user's task sequence by identifying the mental workload. The research showed that each task – reading, searching, object manipulation, and mathematical reasoning – has a unique signature of eye movement. The approach is interesting but the general problem is that the analysis of the eye movements can tell what the user's task was in the past but not predict what the user is going to do. The period providing the data needed for the analysis causes latency. It is also not clear how reliable the task identification works.

Merten and Conati go even further with their paper "Eye-tracking to model and adapt to user meta-cognition in intelligent learning environments" [Merten, Conati 2006]. It is questionable how well such a model will work but the title of the paper shows how far the hopes go on analysis of eye movements. Other researchers have similar ambitious ideas and try to use the gaze for solving ambiguities in a speech interface [Zhang, Imamiya, Go, Mao 2004].

There are better chances for reliable activity detection by focusing on special activities like reading which is easier to define. However, the reliability for reading detection presented by [Campbell, Maglio 2001] and [Foo, Ranganath, Venkatesh 2003] is still below 90%. The reading detection algorithms presented in this thesis and their application in e-learning environments are are published in [Drewes, Atterer, Schmidt 2007].

Of course, there are further ideas on what to do with the context information from the eye movements. For example, Cohen et al. used the eye movements when looking at a picture for cropping [Santella, Agrawala, DeCarlo, Salesin, Cohen 2006]. Although the results are not really convincing for practical use they are another example of what might be possible with context information from the eyes.

6.3 A Usability Tool to Record Eye Movements

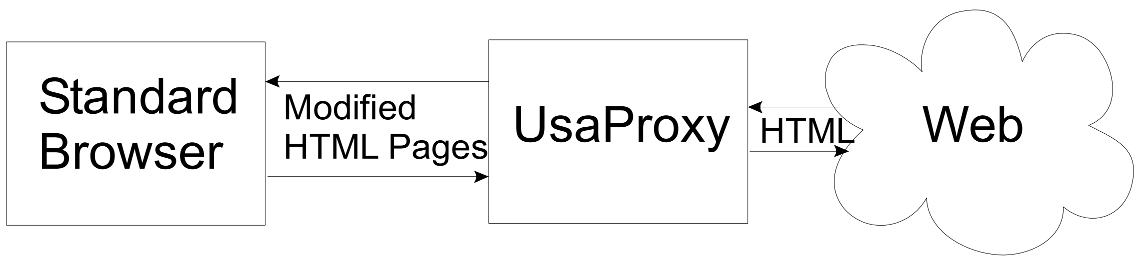
Nowadays, eye trackers are a standard tool for usability research. To understand interaction it is very useful to know where the user looks at. Gaze data can reveal problems of a user interface. For example, if a user looks a very long time at the same interaction element it is a hint that the user does not understand the meaning of it. Gaze data can also show that a user did not notice an interaction element because the gaze never hit it.

In the framework of this thesis a concrete experiment on using gaze context information for a usability tool was carried out. A specific opportunity for such research offered itself through collaboration with a colleague from the same lab as the author of this thesis. This colleague, Richard Atterer, developed a software tool to record all keyboard and mouse activities when a user is interacting with a Web application [Atterer, Wnuk, Schmidt 2006]. As an extension of this tool the idea was close to extend the software to record eye-gaze data also. Beymer and Russell did a similar project with the name WebGazeAnalyzer [Beymer, Russell 2005].

6.3.1 Explanation of UsaProxy

The tool, which was chosen for integration of eye-gaze context, is called UsaProxy. UsaProxy consists of a proxy server written in Java and a JavaScript that the proxy includes in requested pages. The usage of UsaProxy requires that the web browser requests the proxy instead of the HTML server using http (hyper text transfer protocol). See Figure 73 for an illustration. There are two ways to achieve this. One possibility is a server-sided installation, which means that the provider of a web server configures UsaProxy with the IP address and port of the web server to accept requests, which normally would go to the web server. It is the task of UsaProxy to transfer the requests to the web server, which now needs a different IP address or port. With this type of configuration it is possible to trace all interaction activities of all clients connected to the web server. The clients do not notice any difference unless they analyse the HTML code or the net traffic. The other possibility is a client-sided installation, which requires a reconfiguration of the client to send its requests to UsaProxy instead to the standard gateway. In this case UsaProxy may run anywhere in the network but typically it runs on the client. With this type of configuration it is possible to trace all interaction activities of one client to all accessed servers.

Figure 73: UsaProxy in its standard configuration



Beside the transfer of http requests and responses it is the task of UsaProxy to embed a Javascript into the delivered HTML pages and to filter out the reports on interaction activities sent by the Javascript and write them into a log file. The JavaScript reports all events like key presses, mouse moves, mouse clicks, and scrolling to the proxy.

If the HTML-Code received from the web looks like:

```
<html><head><meta HTTP-EQUIV="content-type" CONTENT="text/html;
charset=UTF-8">
<title>Google Advanced Search</title>
```

then UsaProxy modifies the code to:

```
<html><head><script id='proxyScript_UsaProxy'
src='http://141.84.8.64/usaproxylolo/file/proxyscript.js
?sd=2712&amp;ts=2007-06-29,11:28:53.630&amp;
id=' type='text/javascript'></script>
<meta HTTP-EQUIV="content-type" CONTENT="text/html; charset=UTF-8">
<title>Google Advanced Search</title>
```

The modification of UsaProxy consists only of the insertion of a script tag which calls the JavaScript with a timestamp and a server descriptor as parameters. The JavaScript registers event handlers with the web browser and reports all JavaScript events in form of an http-request to UsaProxy. UsaProxy writes the received reports into the log file which looks like this:

```
127.0.0.1 2007-06-25,12:11:47 sd=2712 sid=XdVGZdqwp7T2
   event=mousemove offset=239,132 dom=abbaaac
127.0.0.1 2007-06-25,12:11:48 sd=2712 sid=XdVGZdqwp7T2
   event=mouseover name=as_q dom=abbbaaaaaaaaaabaaaba
127.0.0.1 2007-06-25,12:11:49 sd=2712 sid=XdVGZdqwp7T2
   event=mousedown offset=563,216 name=as_q dom=abbbaaaaaaaaaabaaaba
127.0.0.1 2007-06-25,12:11:49 sd=2712 sid=XdVGZdqwp7T2
   event=select dom=abbbaaaaaaaaaabaaaba text=medieninformatik
127.0.0.1 2007-06-25,12:11:50 sd=2712 sid=XdVGZdqwp7T2
   event=keypress key=m
127.0.0.1 2007-06-25,12:11:57 sd=2712 sid=XdVGZdqwp7T2
   event=focus id=ss dom=abbbaaaaahaaaabac
127.0.0.1 2007-06-25,12:11:57 sd=2712 sid=XdVGZdqwp7T2
   event=change type=radio id=ss dom=abbbaaaaahaaaabac
```

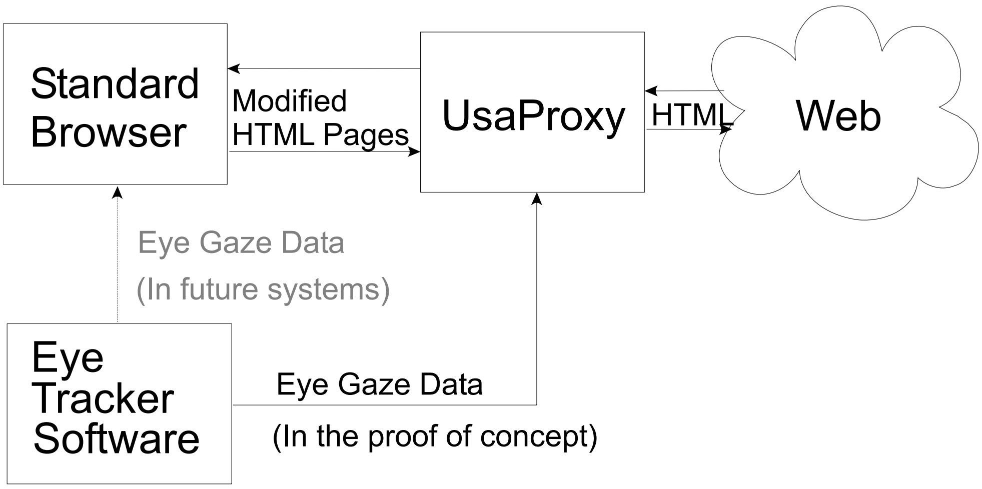
Every entry in the log file starts with the IP address of the reporting computer followed by a date and timestamp. The “sd” (server data) value represents the ID passed by the proxy as an argument at the insertion of the script tag into the header. The unique session ID “sid” identifies each browser instance. The “event” value specifies the reported event followed by values specific to the event. The “dom” value is a string which encodes the exact position of the element in the DOM tree. For a more detailed explanation, see the corresponding publication [Atterer, Wnuk, Schmidt 2006] or the thesis of Richard Atterer [Atterer 2008].

6.3.2 Extending UsaProxy to Record Eye Movements

To record eye movements with UsaProxy there was a technical problem to solve. The problem lies in the fact that JavaScript is not aware of a connected eye tracker and does not report eye tracker events.

Figure 74 shows the solution for the problem. The integration of eye tracker notifications as a regular event in JavaScript would take much effort. It is even impossible for commercial web browsers where the source code is not open. For this reason, the program written as a proof of concept sends eye tracker events to the proxy in the same format used by UsaProxy. As the events come from the same IP address UsaProxy treats them the same way as events reported from JavaScript, which means UsaProxy writes the events to the log file. There was no modification of the proxy necessary. The additional event, called **`gaze`**, delivers the coordinates of the position of the gaze in the **`offset`** parameter. The ERICA eye tracker delivers the same coordinates during a fixation and no data during a saccade. To avoid overhead the extension of UsaProxy sends the gaze event only when the coordinates change, means at the end of a saccade. The **`start`** parameter tells the start of the saccade.

Figure 74: Extension of UsaProxy to track also eye-gaze activity



Tthe measurement of times caused a small modification in the JavaScript. The existing JavaScript code used a time resolution of one second for the time stamps, but for the demands of quick eye movements, the reported time stamps should report milliseconds. With the program sending gaze activity directly to the proxy and the modification of the JavaScript, the result is a system which logs all mouse, keyboard and gaze events on the local system for surfing the web.

With the modifications the log file of UsaProxy looks like this:

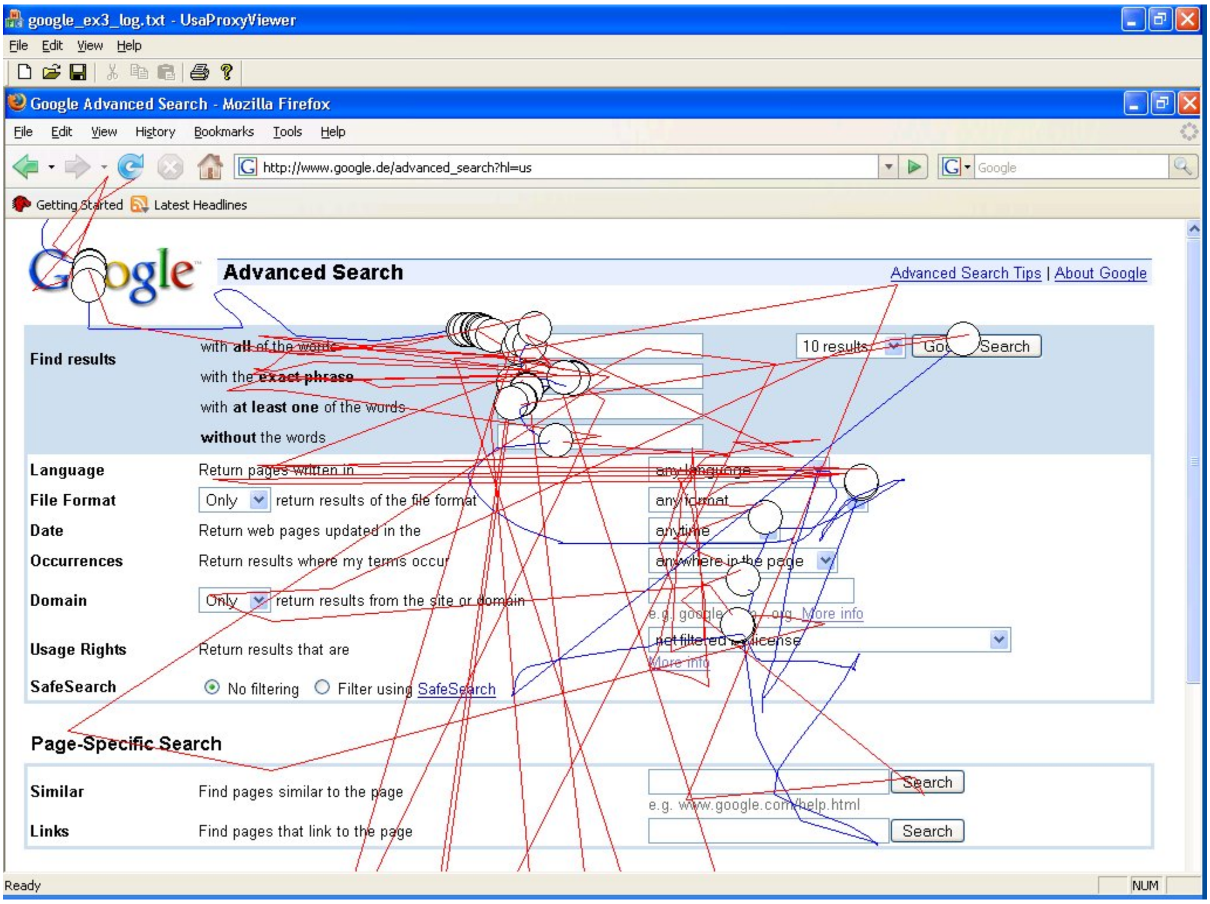
```
127.0.0.1 2007-06-22,16:13:17.192 sid=1
    event=gaze offset=423,217 start=2007-06-22,16:13:17.192
127.0.0.1 2007-06-22,16:13:17.592 sid=1
    event=gaze offset=430,232 start=2007-06-22,16:13:17.572
127.0.0.1 2007-06-22,16:13:16.612 sd=2656 sid=kIhXPMNyI4c6
    event=keypress key=%08
127.0.0.1 2007-06-22,16:13:16.792 sd=2656 sid=kIhXPMNyI4c6
    event=keypress key=%08
127.0.0.1 2007-06-22,16:13:17.693 sd=2656 sid=kIhXPMNyI4c6
    event=mousemove offset=440,219 dom=abbbaaaaaaaaaabaaaba
127.0.0.1 2007-06-22,16:13:18.023 sid=1
    event=gaze offset=897,748 start=2007-06-22,16:13:17.873
```

6.3.3 Discussion of UsaProxy's Extension

The extension of UsaProxy to record eye movements provides an interesting tool for interaction research. There are two possibilities to use it.

The first and most obvious possibility is the analysis of interaction with web applications. Beside a general scientific interest there is also a big commercial interest in understanding the users' interaction. For the owner of an online store the web pages are the only contact to the customer and well-designed web pages increase the profit. The information on the position of the gaze is crucial for understanding the users' behaviour. By evaluating the log files of UsaProxy it is possible to find out how much time the users spend for visual search to find an item in a list of products. The log files can also reveal that the users did not choose an option because they never looked at the corresponding check box. The log files can also tell which text the users read before they cancelled their order. The analysis of the log file and the interpretation of the data are not trivial. Attempts to visualize interaction and gaze data (see Figure 75) show that there is a potential for improvement. However, further research in this direction is out of the scope of this thesis and falling into the topic of information graphics and information visualization.

Figure 75: Visualization of eye-gaze activity (red lines), mouse path (blue lines), and mouse clicks (circles) for Google's advanced search. To type in the search term the user's gaze leaves the screen to look at the keyboard.



The other possibility to use UsaProxy is real-time evaluation of the interaction to provide smarter interaction based on the context information given by the gaze. For example, menus or drop-down list could open when looking at it. However, with the existing extension of UsaProxy it is cumbersome to realize such interaction. The reason lies in the fact that gaze event reports coordinates on the screen but does not report to which object of the DOM (document object model) the coordinates belong. For future Internet browsers it would be desirable to have a **`gazemove`** event similar to the existing **`mousemove`** event. For the design of gaze-aware web pages it would be nice to have an **`ongazeover`** option for the HTML objects like the existing **`onmouseover`** option. For easy programming a dwell time based **`ongazeclick`** option would also be useful.

This raises the general question of appropriate events for gaze control of user interfaces. Mouse devices and eye trackers deliver very similar data, typically x-y positions with a timestamp. Therefore, the first idea is to provide a notification mechanism for eye movements in the same way as done for mouse movements. However, there are also differences between mouse and eye movements. The ERICA eye tracker delivers x-y coordinates at a constant rate during a fixation and does not deliver data while the eye is moving. This is just the opposite of mouse notifications which do not report any data while the mouse is in rest, but sends data while the mouse is moving. The presented extension of UsaProxy delivers gaze events only when the gaze coordinates change. This means it reports only the ends of saccadic movements and reports the start time of the saccade within this notification. It is worth to discuss whether the notification mechanism should report the end of a fixation. However, as the ERICA eye tracker does not deliver such event it was not possible to implement this alternative. Within this context it is also worth to think about a lost-signal notification. A mouse pointer is always somewhere on the screen. The gaze, however, can leave the screen.

In addition to the differences in the notification mechanism of mouse and gaze events there are also differences in the way to treat such notifications. The example of gaze-triggered tool tips or balloon help as mentioned above (see 6.1) shows that it is not possible to transfer the handling of mouse events directly to analogue gaze events. A tool tip from mouse events is visible as long as the mouse pointer is on the object and disappears when the mouse leaves the object. In contrast, a tool tip displayed when the gaze is over the object should not disappear when the gaze leaves the object to read the tool tip text. In consequence, it is worth to think about gaze-event mechanisms. However, the introduction of a **`gazemove`** event would be a first step on the way to develop gaze-aware applications which do not demand a specific eye tracker.

Further development of UsaProxy allows surfing the internet together from two distant workstations [Atterer, Schmidt, Wnuk 2007]. That means the two persons see the same content in the browser and even the other mouse cursor. It would be an interesting project to extend the UsaProxy so that the two persons can also see where the other one's gaze is. However, this requires a second eye tracker which was not available.

Lessons learned:

- Recording interaction and gaze activity is easy, but it is difficult to analyse the recorded data.
- A gaze-aware browser should provide a **`gazemove`**, a **`gazeover`**, and a **`gazeclick`** event.

6.4 Reading Detection

Reading is a very important activity when using a computer. Today, many researchers in the field of human computer interaction are interested in reading. For an example see Nielson's web page [Nielson@]. Psychologists intensively researched the reading process, which is understood in detail [Reichle, Pollatsek, Fisher, Rayner 1998]. Because there is already research for reading detection as activity [Campbell, Maglio 2001], [Foo, Ranganath, Venkatesh 2003], [Bulling, Ward, Gellersen, Tröster 2008], this work focuses on detecting whether a particular text was read. One possibility to detect whether a text was read is to use the reading detection mentioned above, collect the x- and y-coordinates, and find out whether these coordinates completely cover the text area. This will result in a rather complex algorithm and as the error rate for reading detection in the mentioned research is still above 10 percent; such an algorithm will not produce reliable results. One algorithm presented here provides feedback to the user and therefore achieves a perfect detection with the help of the user. The other algorithm gives numerical values on how intense a text was read. It is worth mentioning that these algorithms work fine with standard sized fonts, i.e. with font heights which are much smaller than the accuracy limit of the eye.

6.4.1 Analysis of the Gaze Path while Reading

Recording the gaze path in initial experiments resulted in patterns as shown in Figure 76.

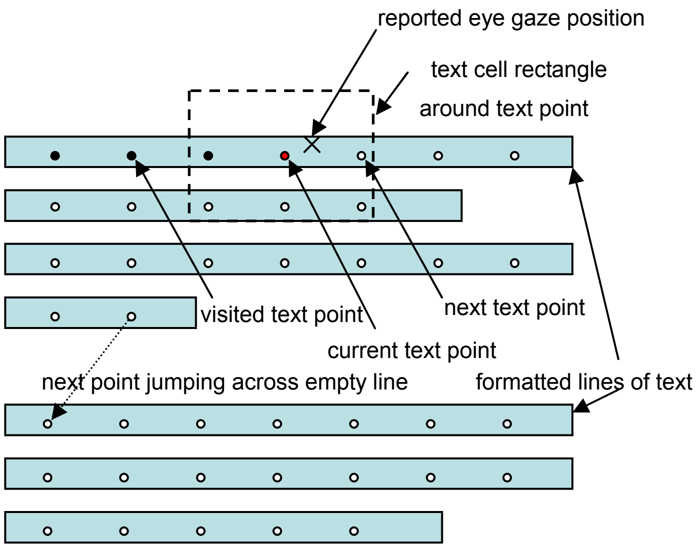
Figure 76: Gaze path reading a text

It is easy to see that the gaze moves in saccades with brief fixations. Revisits can occur especially in difficult texts. A preliminary evaluation of the data gave saccade lengths of around 30 pixels within a line and backward saccades which are slightly smaller than the length of the line. The geometry of the system results in the 1° visual angle being 36 pixels wide. Thus, the angle of the forward saccades is about the same as the angle of the fovea or the angle of accuracy (see 3.2). It is interesting to note that in consequence the first fixation points in a line are about half a degree behind the beginning of the line and the last fixation is the same distance away from the end of the line. In the vertical direction the low accuracy causes the problem that it is not possible to detect reliably which line the gaze is reading if the line height is within a normal range. The line height in Figure 76 is 19 pixels (≈12 pt) which are about 0.5°. Sibert et al. avoided the problem for their reading assistant [Sibert, Gokturk, Lavine 2000] by using a 42 pt font. The algorithm presented here has the ambition to work with standard size fonts.

6.4.2 An Algorithm for Reading Detection

The reading detection algorithm uses the fact that the gaze visits the text in sequential order. The basic parameters of the algorithm are the width *w* and the height *h* of a rectangle around text points and the distance *d* of text points. The height *h* should be a little bit bigger than the accuracy so that the eye tracker will catch the gaze position when reading the line. The width *w* must not be smaller than the distance *d* of text points to cover the line completely and should be bigger than a typical forward saccade so that the saccade does not jump over it. The algorithm starts with a text point which is *d*/2 from the beginning of the first line. As soon as the reported gaze position is within a rectangular area around the current text point, the text point is marked as read. Then the algorithm chooses the next text point on the same line, *d* ahead of the previous one. In case of a line wrap, the next point is at *d*/2 from the beginning of the next line. A line wrap happens when the next text point in the line is less than *d*/2 away from the end of line (or behind the end of line). This rule in the algorithm is crucial, as people tend to skip the end of the line even when they read carefully. Once the gaze visited all text points the text was read completely. See Figure 77 for an illustration.

Figure 77: Illustration of the reading detection algorithm



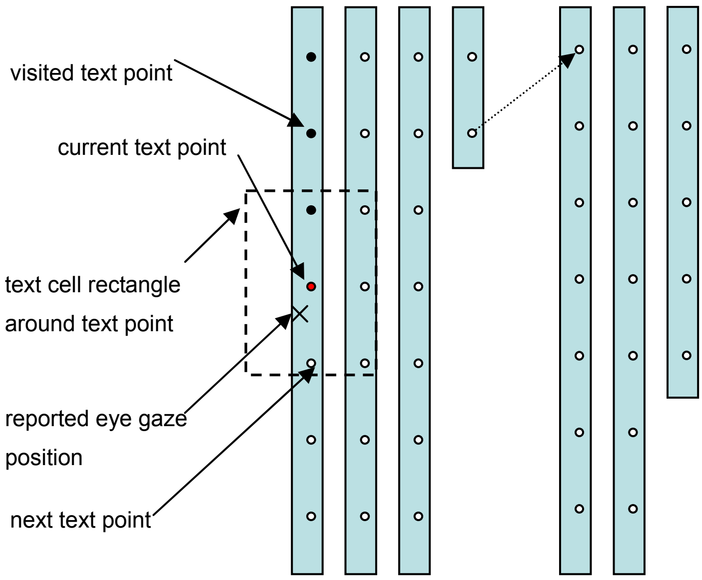
By design, this algorithm allows the reader to take a break, perhaps to have a look at the system clock, before continuing to read. It is also robust against calibration shift because the dimensions of the rectangle can be relatively large (compared to a calibration shift) without decreasing the reliability. A problem of this algorithm is the case that some words or lines are left out by the reader. In this case the algorithm ‘hangs’ that means the algorithm waits for the left out word to be read and ignores all further reading. For this reason the system should provide feedback on the portion of text being read for instance by a change of the background colour. As mentioned already it is important not to have text points close to the end of the line, because such ends of the line would be places where the algorithm hangs. In Figure 80 it is visible that some lines exceed the coverage by the text cells.

The interesting aspect of the algorithm is that it works with standard font sizes, i.e. with font heights below the accuracy limit. This property is a result of the demand that the height of the text cell has to be bigger than the accuracy of the eye. In consequence the algorithm marks a text cell as read even if the gaze is a little bit above or below the line. The ambiguity between the current text line and the upper or lower text line dissolves by the concept of a current text point which implies also a current line.

A problem arises with very short texts which can be read with one glance. To detect reading in a line of text there should be at least two or three fixations. Together with the fact that the first and last fixation points are about 0.5 degrees from the beginning and end of the line and that the fixations steps are around one degree, a line should have a minimum length of 3° visual angle. This is about 100 pixel on the system used.

As traditional reading always works in sequential order independent of the script used, the algorithm can be applied to right-to-left scripts (Arab) and to vertical scripts (traditional Chinese) too (see Figure 78).

Figure 78: Illustration of the reading detection algorithm for vertical scripts



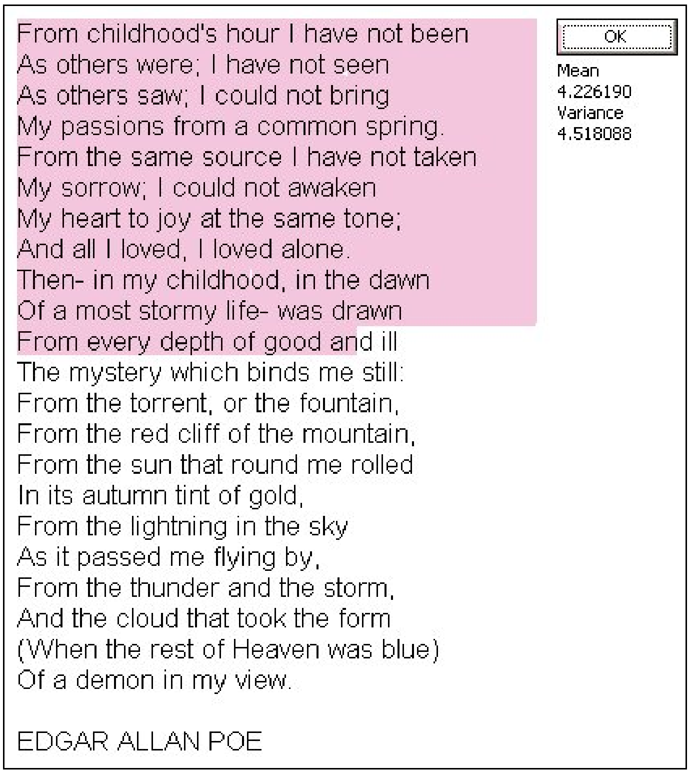
It is worth to emphasize that the algorithm only ensures that the gaze visited all text points, but does not tell whether the reader understood the text. It is possible that the eye read the text mechanically while the mind is absent. It is also possible that a user reads a text in a language she or he does not understand if the system demands it.

6.4.3 User Study for Reading Detection

For the conduction of a user study special software was written including the implementation of the reading detection algorithm. The width *w* of the rectangle was set to 100 pixels, the height *h* to 50 pixels, and the distance *d* to 100 pixels. See 2.6.2 for the geometry of the setup and the conversion of pixels in degrees. With the short lines presented in the study there were only two text points in a line.

A group of nine people, six male, three female, aged from 23 to 47, took part in the user study. Every subject got three tasks. The first task presented a dialog box with a poem of Edgar Allen Poe (23 lines, 150 words). In the first task the dialog presented feedback for the text already read (see Figure 79). The dialog box disappeared when text was totally read. The task was to read the text to make the dialog disappear. The second task was another dialog box with a poem in the native language of the user (13 lines, 69 words), this time without providing feedback. The third task was a repetition of the second task but this time with the demand to read the poem loudly.

Figure 79: Feedback for compulsory reading



In the first task all test users were able to let the dialog disappear (100%). Some of them complained that the feedback was faster than their reading and they had the feeling of being chased. Consequently, further research should test feedback with delay. There are two possibilities to achieve this. One possibility is a delay based on time. The other possibility is not to provide feedback for the text cell currently read, but for the text cell read before (and special treatment for the very last text cell).

In the second task, the dialog detected correctly complete readings in six cases (67%) and failed in the other three cases (33%). In the third task, the dialog correctly detected complete readings in five cases (56%) and failed in the other four cases (44%).

The results of the second task were much better than expected while the third task was worse than expected. Without visual feedback the expectation was that users do not read carefully and the algorithm will not complete. Because the task given was to read the text, the users did this carefully. Perhaps this would not be the case if another task had been given for which it is necessary to read something. The third task, reading loudly, had the intention to force the users to read the text slowly and carefully. However, the chin rest for head fixation was an obstacle for speaking and the users moved their heads, which had a negative effect because the eye tracker calibration assumes a fixed head position.

A detection rate of 67% is sub-optimal and consequently compulsory reading should provide feedback, where the success rate is 100%.

Lessons learned:

- For compulsory reading provision of feedback is necessary.
- Instant feedback disturbs the reading process.

6.4.4 Values for the Quality of Reading

Detecting whether a text was read carefully or just skimmed needs another approach. A heat map is the typical way to visualize how good a text was read but it needs a human mind to interpret. A programmatical approach needs a method to reduce the data to a few values.

The proposed algorithm starts with formatting the text to find text points and surrounding rectangles. Every fixation inside the rectangle increases a counter for this area. It is also possible to accumulate the fixation time. This mechanism is a kind of a low-resolution heat map and illustrated in Figure 80. The data allows calculating two values – a mean, which tells the number of fixations per text cell, and the variance or standard deviation, which tells how uniformly the fixations are distributed over the text cells.

If there are *n* text cells and $f_1, f_2, \ldots, f_n$ are fixations for each cell the intensity of reading *I* can be defined by the mean and a normalizing factor *c*:

$$I = \frac{c}{n} \sum_{i=0}^{n} f_i$$

Using the mean makes *I* independent of the length of a text. The factor *c* adjusts the value of *I,* so that it will be 1.0 if a text was read carefully once, and depends on the size of the text cell used in the algorithm. If the vertical extent of the text cell is the height of a text line, the horizontal extent is the distance of the text points, and distance of the text points is an average length of a forward saccade, then *c* is 1. If the text cells are bigger, several fixations can fall into the text cell, and if cells overlap, one fixation is counted multiple times because it is inside several text cells. The factor c will correct this to get the value 1 for a complete read.

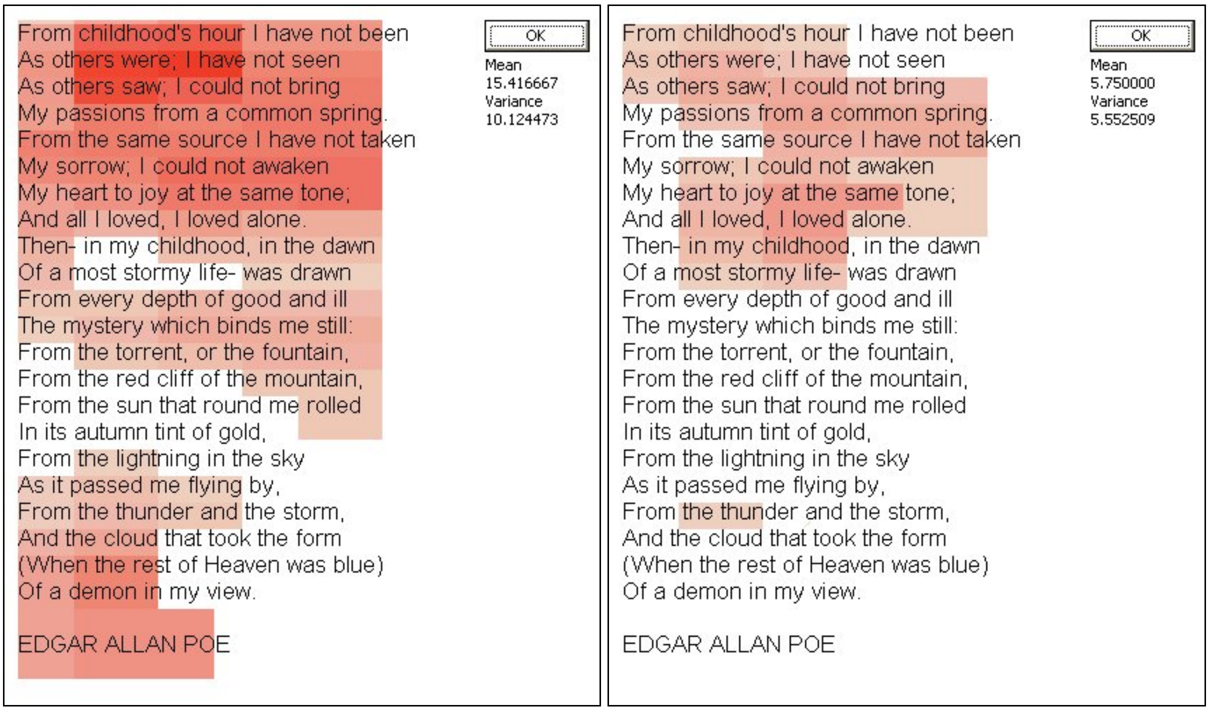
To decide whether a text was read carefully I is not enough because reading half of the text twice results in the same value as reading the whole text once. Consequently, a second parameter D is necessary to know whether the gaze positions are distributed equally over the text cells. The variance provides such a parameter.

$$D = \frac{1}{n}\sum_{i=0}^{n}(cf_i - I)^2$$

If the text cells were all visited equally D will be close to zero, otherwise D will be higher. A perfect read will result in $I = 1$ and $D = 0$.

It is possible to visualize the quality of reading by colouring the background of the text cells according to the number of fixations counted for each text cell. Text cells with a high number of fixations get a dark background colour while text cells with a low number of fixations get a light one. The result is a so-called heat map but with a resolution of a text cell and not with a pixel. Figure 80 shows examples for such visualizations.

Figure 80: Examples for the quality of reading when skimming a text



Lessons learned:

- A single numerical measurement is not enough to judge how well a text was read. At least two measures – the number of fixations and their distribution over the text – are required.

6.5 Gaze-Awareness in E-Learning Environments

E-learning environments comprise a field of application for gaze-awareness and benefits may be derived from gaze awareness for learners and teachers. A student who revises her or his documents for an exam can see how often she or he read the text already and which text has not been read up to now. In a community of students it is possible to share such a gaze-aware system. Then the system can tell a student on which paragraphs all other students spent more time than her or him. For the lecturer it is possible to see which paragraphs make the students struggle and which sentences the students look over and never read. With a gaze-aware e-learning system, the measured time of attention by eye gaze spent to the presented material is reliable and pedagogic decision can be build on that. Recording the eye-gaze activity provides useful feedback for teacher and student.

It is reasonable to assume that future e-learning will be built on internet technologies because these provide a hardware and operating system independent platform as well as worldwide access to the lessons. The extension of the UsaProxy and the reading detection algorithms, both presented in the previous sections, are useful components to build a gaze-aware e-learning environment.

For further research in this field it would be helpful to involve scientists with knowledge in pedagogics.

6.6 Summary of the Results for Eye Gaze as Context Information

The extension of the UsaProxy presented here allows the recording of all interaction with a web browser including the movement of the eyes and makes it a powerful tool for usability research. The use of web technologies and of the internet protocol allows collecting the interaction data on any host in the net. This means that the host presenting the web page can analyze the interaction and the gaze, so that the host can use this information to control the presentation of further content. For example, a web page can analyze the fixations on the page to find out that a user did not look at an available option, and therefore, ask the user to confirm the value of the unseen option. The general problem of the approach is the latency of several hundred milliseconds for data transmission in the net which makes real-time response to eye movements difficult. For this reason, future operating systems should integrate eye movements in a similar way like mouse movements and report the eye moments with messages to the application. This would allow to extend the message handlers in JavaScript for processing eye movements and to provide real-time response to eye movements in the browser.

Reading detection is an example of eye-gaze context information which is not trivial, but also not too vague. In contrast to the reading detection algorithms published so far, which detect reading as activity, the algorithm presented here detects whether a particular piece of text was read. The remarkable property of the reading detection is that it works quite well even with line heights below the eye tracker’s accuracy. That means that the reading detection works with the standard font sizes used in existing GUIs. It is in the human nature to skip some words when reading a lengthy text quickly. To deal with this human behaviour the second algorithm provides numerical values for how good a text was read. For the same reason, a system should provide feedback on what text was read already when demanding to read the text carefully and complete. The user studies showed that a direct feedback disturbs the reading process. The users reported that they felt chased by the immediate feedback. Therefore, future research should try feedback appearing with a delay.

6.7 Conclusions on Eye Gaze as Context Information

Eye gaze provides useful information even if this information is not used for active interaction. This means that the context information is used for analysis purposes and not for context awareness. In the case of UsaProxy the extension to record the gaze position gives the possibility to analyze the users' interaction with respect to what the users have seen. Such kind of analysis has big commercial impact for web shops, as the web interface is the only contact to the customer. If the customers of a web shop do not look at a special offer presented on the page, they will never order it. The eye gaze recording extension for UsaProxy is a powerful tool for this kind of analysis. The results of such analysis in form of general design rules for the arrangement of text, graphics and interaction elements, is still a subject of research. In addition, the information visualization of gaze and interaction recordings needs further research for easy and quick evaluation of such data.

The use of context information derived from eye gaze for building context-aware applications is still in the very beginning. The use of eye gaze as an indicator for attention is simple and powerful. Higher levels of context awareness could detect the users' activity. The example researched here is reading detection. The algorithms presented are simple and work quite well despite font sizes below the accuracy threshold of the eye. The detection whether a text was read completely requires visual feedback for 100 percent reliability, but this is not a restriction as providing feedback is a recommended feature for the design of user interfaces. Without the visual feedback, the algorithm presented delivers numerical values for the quality of reading. This is valuable context information in a world where the reading of documents is a major part of the job. When looking at the elaborate models on reading from the field of psychology [Reichle, Pollatsek, Fisher, Rayner 1998] it becomes clear that there is further potential in reading detection. From the speed of reading, the number of backward saccades and the time needed to read difficult words it should be possible to get information on the reading skills and formation of the reader. An online bookshop could use such information for book recommendations in the future. There is definitely much space left for further research.

Eye gaze is very important for human communication. If we want good assistance from computers, the computers have to know where we look at. However, the interpretation of gaze can be difficult because even humans cannot always read the wishes from the eyes of others. The correct interpretation of the way people look requires a kind of social intelligence and it will not be easy to implement this on a computer. It is not clear whether it is enough to analyse the gaze of a person to conclude something or whether is needs additional information on the person, i.e. to know this particular person. Research in that direction raises many interesting questions, for example, whether the computer can finally be the better observer as the computer supposed to be neutral and not biased by emotions.

7 Conclusions

7.1 Summary of the Results

The focus of this thesis was to find out whether and how eye-tracking technology can provide new interaction techniques for the masses. The increase of publications in this field during the writing of this thesis shows that there is a growing interest in the topic and the results presented here show that eye tracking is a promising interaction technique for the future.

The thesis divides the field into four different aspects represented by the chapters 3 to 6.

The research presented in chapter 3 (the eye and its movements) deals with the fundamentals of eye movements. The goal was to find out the limits for eye movements, i.e. the maximum speed and the accuracy of eye movements. The results are not expected to be new because psychologists researched eye movements for more than 30 years, but the results are presented here in the special context of looking into a computer display and the discussion is specific to the HCI community.

Chapter 3 gave the following picture: The eye gaze can be used as a quick pointing method but with a limited accuracy. The natural inaccuracy is in the range ± 0.5° and seems to be intrinsic to the eye and its movements. This means that a better accuracy is not a question of improved eye tracker's accuracy. The eye gaze normally moves to a target with a single movement (saccade). The movement time for long saccades (>5°) typically is between 100 and 150 milliseconds and does not much depend on the distance of the gaze position to the target. Because of anatomic variability among the humans, a calibration procedure is always necessary for best accuracy in pointing by gaze. Novel eye-tracking technologies, which are not commercially available yet, allow a single point calibration which can be done implicitly without being noticed by the user.

The statistics on fixation times and saccades per second show that it is possible to detect the user's activity by analyzing her or his eye movements. The data also show that eye movements depend more on the task than on the individual and that the differences among individuals are not extreme.

There is no reason that Fitts' law is applicable to the eye movements. The debate on Fitts' law for the eyes in the HCI community ignores the standard knowledge of psychology. There is no user study clearly showing that the positioning speed of the eye depends on target size. Target size dependency in experimental data is an effect of distance definition or an accuracy issue as demonstrated in the circle experiment. Additionally, the accuracy of the eye requires a minimum target size. With a minimum size for targets Fitts' law is of no importance because it is not possible to achieve a better accuracy by spending more time.

The question behind the studies from chapter 4 (eye gaze as pointing device) was how to deal with the limited accuracy of the eye when using the gaze as pointing device and whether the introduction of an extra modality (touch sensor) can help to solve the problem. Chapter 4 summarizes to the following:

High precision pointing with the eye is a difficult issue and it is not clear whether it is possible at all. Consequently, gaze is no direct substitute for a mouse device. The eye does not have the precision to position a text cursor between two letters. One solution to the problem is the enlargement of all GUI elements. Another possibility is to use the gaze for raw positioning for example on a word and navigation keys for the fine positioning within the word. The approach presented here – the MAGIC touch – uses the gaze for raw positioning and the mouse for fine positioning. The MAGIC touch reduces the mouse paths for interaction enormously and feels comfortable for the users. The speed benefit is marginal and in many cases only a subjective feeling. The MAGIC touch approach is an extension which leaves the existing GUI unchanged and therefore could sell as an add-on product immediately. The feedback from the participants of the user studies showed that there is a potentially high acceptance for the technology.

The intention behind chapter 5 (gaze gestures) was to try an alternative approach to direct the computer. In summary, the user studies show that the gaze gestures presented here work quite well. The gaze gestures are a special purpose gaze-only interface and not as intuitive as eye gaze pointing. The big advantage of gestures is that they do not require a calibration procedure, allowing instant use by different people. Whether this would be accepted by the masses as an interface – for example as remote TV control – is still an interesting question.

The approach in chapter 6 (eye gaze as context information) is to use unintentional eye movements to provide context information. It leaves the concept of directing the computer intentionally with commands given by the eye. The computer analyses the movements of the eye to give smart assistance to the user. Chapter 6 had the following outcome:

Eye gaze provides valuable context information. For the aim to provide human-computer interfaces, which work similar to human-human interaction, the information from the eye gaze is essential. Context information in general is still a vague concept in a range from trivial applications to social intelligence. The simplest form of context information from the eyes is attention. Most of the time we look at the things we pay attention to and attention is a powerful concept for interaction. The extension of UsaProxy to record the eye gaze together with standard interaction activities makes it possible to understand the relations of attention and interaction. The reading detection algorithms studied in this work are a step towards higher forms of context awareness. As reading is a very frequent activity when using computers there are practical applications of this form of context. Higher context levels such as detecting whether the user is searching for something or even the estimation of the users' physical and emotional state are desirable and topics for further research.

7.2 Conclusions for Eye Gaze User Interfaces

The most fundamental decision for the design of eye gaze UIs (User Interfaces) is whether to use the voluntary eye movement of the user or the information of more or less unconscious eye movements. The first option can be called **active gaze control**, while the other possibility can be called **gaze context control**. Typical examples for active gaze control are eye typing or gaze gestures. Examples for gaze context control are attention sensors for a display that turns on when somebody is looking at it. The general difference between both approaches is that for active gaze control the computer expects commands initiated intentionally by a user while for gaze context control the computer observes the user and tries to assist the user without explicitly given commands.

There are different opinions about which approach is the preferable one. Some people think that using the eyes as output for commands will conflict with the eyes' task of vision and that such use of the eyes is unnatural [Zhai, Morimoto, Ihde 1999]. However, eye typing works quite well. The reason not everybody is using it seems to be not only that this technology is still expensive, but also that keyboard and mouse are more efficient. The concerns against the gaze context approach could be a lack of reliability. Context information is fuzzy and may lead to unwanted actions. Control by context information also implies some loss of control for the user, which some users may dislike. However, one option to deal with the fuzzy nature of context is to use it in assistive but 'non mission critical' ways.

The question of how to design eye gaze UIs depends on the goal. The different forms of eye gaze interaction have specific advantages and disadvantages. Table 29 gives an overview on the specific properties of the different eye gaze interaction techniques.

Table 29: Goals and problems of different eye-gaze interface techniques

Goal / advantage	**How to achieve**	**Problems solved**	**Problems that stay**
Extension of existing GUIs	a) gaze for raw and mouse for fine positioning b) temporary enlargement of interaction items	Existing GUIs use targets smaller than the accuracy Extensive use of hand Users can not find the mouse pointer	Calibration is necessary No relevant speed benefit
Fast interaction	Gaze, key and big targets	Time needed for selection	Calibration is necessary Space limitations New GUI development
Public, hygienic and remote	Gaze gestures	Need of calibration	Gestures are not intuitive No speed benefit
Smart assistance	Attention sensors	Amount of interaction	Only simple interaction possible
Smarter assistance	High level analysis of gaze data	Stupidity of computers	Useful applications are still to research Reliability

There are many reasons why it is desirable to use gaze interaction in combination with existing GUIs. One reason is that the users can keep the GUI they are familiar with. Another reason is that the development of new user interfaces costs a lot of effort. The general problem with the existing GUIs is that the interaction elements, buttons, menu item, etc. are smaller than the accuracy of eye pointing. There are several suggestions in the literature to solve this problem. The approach of Salvucci [Salvucci, Anderson 2000] with adding intelligence to the target selection is interesting but it is easy to construct situations where conflicts occur. Another way to solve the accuracy problem is a temporary enlargement of the interaction targets. There are several ways to do so – with a fisheye lens [Ashmore, Duchowski, Shoemaker 2005], expansion of targets [Miniotas, Špakov, MacKenzie 2004] or expansion of parts of the screen as done for EyePoint [Kumar, Paepcke, Winograd 2007]. These methods have the potential to make the mouse obsolete, but they all require changes in the coding of the GUI and it is not clear whether all interaction of existing GUIs can be re-mapped to the full satisfaction of the user. Gaze positioning with an extra key or a touch sensitive mouse need the least changes in the user interface and in the coding of the GUI. The MAGIC pointing principle [Zhai, Morimoto, Ihde 1999] is very intuitive if the gaze input is triggered by a key or touch sensor. Using the eye gaze for interaction with existing GUIs reduces the motor tasks for the hand, but does not bring much more benefit. The speed benefit in particular is marginal as long as the displays are not very large.

Eye gaze interaction makes it possible to create interfaces for fast interaction. Fast interfaces are useful for gaming, military purposes, or real time editing of media streams. For an eye-gaze-only interface the Midas Touch problem, as identified by Jacob [Jacob 1990], causes dwell times which slow down the interaction process. The way to achieve fast interaction, already shown by Ware and Mikaelian [Ware, Mikaelian 1987], is the introduction of an additional modality, normally a key. Nevertheless, regarding the need of keys it should be considered that a keyboard is a very fast interface already. The other requirement for fast eye-gaze interfaces are target sizes above the accuracy threshold. This causes a space problem which means that a display can provide only limited number of interaction targets, similar to a keyboard which can only have a limited number of keys. Consequently, it does not make sense to design an eye-gaze interface to compete with the keyboard for speed. The application that makes sense for fast eye pointing should deal with spatial mapping. For instance if there are nine windows with live content of nine cameras on the display and the task is to choose the camera for the broadcast in real-time, the number pad of the keyboard is definitely a proper and fast interface. An eye-gaze interface that selects the picture by gaze and the press of a confirm key will not be faster. However, the eye-gaze interface is more intuitive and convenient. For the eye-gaze interface it is not necessary to learn and remember the mapping to the keys. Although the mapping to the keys is trivial it still needs a time of training to get familiar with the sizes and arrangement of the keys.

One obstacle for eye-gaze interfaces is the need of calibration. On a personal system it does not really matter because the calibration is needed only once, but it matters for public systems. Even stereoscopic eye trackers require a single point calibration. Gaze gestures are a calibration-free interface technique and consequently are well suited for public access. As gaze gestures are a gaze-only interface technique and work over some distance they provide a remote control without the need of an extra device. Because natural eye movements mostly consist of small saccades, and big saccades occur rather seldom, a gaze gesture consisting of several big saccades is easy to separate from natural eye movements. The problem with the gaze gestures is that they are not very intuitive and for around 10% of the users in the studies it seemed to be a hard task to perform the gaze gestures. The introduction of the mouse as standard device for the masses showed that people do not have equal abilities in learning new motor skills but nowadays nearly everybody can use a mouse (for pointing tasks). This gives rise to the hope that after some practice nearly everybody can perform gaze gestures with ease. Nevertheless, the usage of gestures is not only a question of ability but also a question of acceptance. Mouse gestures for example are popular but only for a small group of specialists. In contrast, the finger gestures to operate devices with touch surface seem to be accepted by the masses. Beside the advantage of being calibration-free, which allows instant use by different users, gaze gestures have further advantages for the use in public. An eye tracker does not have moving parts and is maintenance-free. Compared to a keyboard or track-ball an eye tracker is less vulnerable against vandalism. As gaze gestures are a gaze-only interface there is no need to touch anything for interaction and so the gaze gestures are a very hygienic interface – a big advantage for a public interface.

Following the concept of gaze as context makes it possible to build smart interfaces that react on eye contact as humans do. Eye contact sensors are simple eye trackers which do not need a calibration because of their low demand for accuracy. Eye contact is closely related to attention and attention is a useful context. Simple attention detection cannot only help making devices smarter but also can make interaction safer by ensuring attention. A mobile phone that requires eye contact for an outgoing call will not initiate a call because of accidentally pressed keys. Speech recognition will not interpret a conversation with somebody else as command when eye contact is a precondition. However, eye contact alone does not enable complex interaction.

There is more context information in eye movement than just attention. Most languages have idioms containing ‘look’ and ‘eye’; for example, somebody can have a look of despair in his or her eyes. We are able to judge somebody’s physical and emotional state from how she or he looks. Most probably we do not only analyze the motion of the eyes but also the facial expression and even the posture and movements of the body, but the eyes seem to have a central meaning for the estimation of somebody’s physical and emotional state. Moving eyes are an indicator of being alive. A person whose eyes move hardly and slow is mentally absent or sleepy. If somebody’s eyes move quickly she or he is nervous and alerted. People that constantly gaze at objects that seem not to be of any interest makes us doubt their mental condition. Therefore, the eyes can tell a lot about a person. For example, an analysis of eye movements during reading can detect which words are difficult or unfamiliar to the reader and tell something about the literacy of the person. Interfaces that utilize such context information have to solve three main problems: what kinds of stimulus to give to the eyes and how to integrate it into the user interface, how to analyze the eye movements, and how to react on the results of the analysis. Solving these problems would cause that the computer appears less stupid to us and that it could be a more human-like communication partner as discussed in 1.2. Up to now there is no general concept to answer these questions. This field needs further research and the last row in Table 29 reflects the results of future work.

The first three lines in Table 29 follow the concept of active gaze control and the last two lines follow the concept of gaze context control. It seems to be worth following both approaches and perhaps a combination of both will yield the perfect eye-gaze interfaces. There is no sharp borderline between both approaches. The reading detection algorithms are a good example for this. Reading detection normally works as a background process and represents the gaze context approach. The computer observes that the user is reading and assists for example with automatic scrolling. In the case of compulsory reading however, for instance a dialog that does not close until its text was read, the reading becomes a conscious command and can be seen as the active gaze approach. The situation is similar for a gaze assisted navigation system which assists the user by automatically zooming when looking for a while at a same area of the display. In the beginning, the system observes the user and gives assistance, but as soon as the user has developed an understanding of the system's behaviour, she or he can use this to control the system. An incidental interaction becomes an expected interaction through knowledge of the system behaviour, and finally turns into intentional interaction by explicitly moving the eyes to get the expected behaviour (see also [Dix 1998, p. 649ff]). The discussion whether it is natural or unnatural to control the computer with the eyes dissolves at this point.

7.3 How to go on with Eye-Tracking Research for Interaction

As already mentioned in the introduction the knowledge for building gaze-aware interfaces spreads over many disciplines. The situation of a researcher or research team from one discipline is difficult because of missing knowledge from the other disciplines. This causes a waste of time by repeating experiments which were already done or by becoming an expert of another discipline first. Consequently, an interdisciplinary team can boost eye-tracking research for interaction. It is clear that interaction research needs a specialist from the field of HCI. Beside that HCI researcher, a psychologist would help a lot. Psychology looks back on several decades of eye-tracking research and there are many results which could contribute to the goal. Psychology knows about attention and recognition. It studied the reading process in detail and could answer the questions on motor learning or the memorising of gaze gestures. Medicine too uses eye tracking for many years now, mostly for diagnosis and physical rehabilitation. Medicine knows about fatigue of eye muscles and how high the intensity of infrared illumination used for the eye tracker may be. For the development of an eye tracker with properties a commercial eye tracker does not have, it requires a computer scientist with skills in image processing. Finally, it helps to have experts in the team for special applications. In the case of gaze-aware advertisement this would be a communication scientist and for gaze-assisted e-learning an expert in the field of pedagogy.

The research on eye tracking for interaction increased steadily over the last years and created ideas which were tested in the laboratory. For the goal of a product, which could sell in the mass market, these ideas should be tested outside the laboratory in a real environment. For example, the gaze positioning with the touch-sensitive mouse presented in chapter 4.3 has the potential to be made into a product. It needs an eye tracker which allows free head movement (and not an eye tracker with a chin rest) to build a prototype for a user study in a real working environment. For such a user study the participants should work at least a week with the system and not only half an hour as is typical for a study in a laboratory environment. As a user study requires a certain number of participants to produce statistically significant results, the availability of more than one prototype will bring quicker results by conducting the study in parallel instead of sequentially. If the gaze-positioning system should sell as a medical product to cure repetitive strain inquiry with a subsidy from health insurance it would be of advantage to have the medical department involved. Despite falling prices eye trackers allowing free head movement are still costly. The patents for the eye tracking methods allowing free head movement may concern people or institutions that finance such research.

For the gaze gestures presented in chapter 5 it is not clear whether they have the potential for a mass market product, for example as a remote control for a media centre, or whether they are only applicable for a niche product. Nevertheless, the situation is similar to gaze-assisted pointing and it needs a prototype for tests outside the laboratory.

The usage of eye gaze as context information is still an open field and definitely good for further scientific publications. There are many open questions to answer. It starts with practical questions like what do we see on a web page and what we do not see? What are the implications for a good design of web pages? Further questions are to which extent and how reliable a computer can detect the user’s activity or intention from the gaze. How should a system react when knowing the user’s activity or intention? Is it useful or even necessary to combine the information from gaze with other context information? Is there a general concept to utilize eye gaze to build smart devices or does it depend on specific applications?

Beside the technological questions there are also social or human related questions. Does eye-tracking technology help us to keep our privacy, as it is the intention of the research presented in section 5.4.5, or does it threaten our privacy? An eye tracker can reveal that an employee did not concentrate on his or her work and looked out of the window most of the time. The other way round an eye tracker can detect that we worked to hard for too long and encourage us to have a break and protect our health and ensure the quality of our work. An eye tracker can also find out if a person's degree of literacy is low because of low reading speed and because of reading long sentences and difficult words twice. Again, this raises the question in what way we want to use the capabilities of eye-tracking technology.

It is also possible to ask application specific questions. Does a gaze-aware e-learning application increase the success or motivation of the learner and does it matter whether the learner is an adult or a child? Does a gaze-assisted video conference system, which does not only present the document to talk about, but also where on the document the gaze of the other is, support the communication? Can gaze-aware advertisement increase the sales?

Finally, eye-tracking technology without the aspect of interaction needs further research, especially on eye trackers for outdoor use.

The list of interesting research questions given above is long and far from being complete. It is not clear which question is most urgent to answer or most promising for results. Therefore, it is up to the researcher which question to pick.

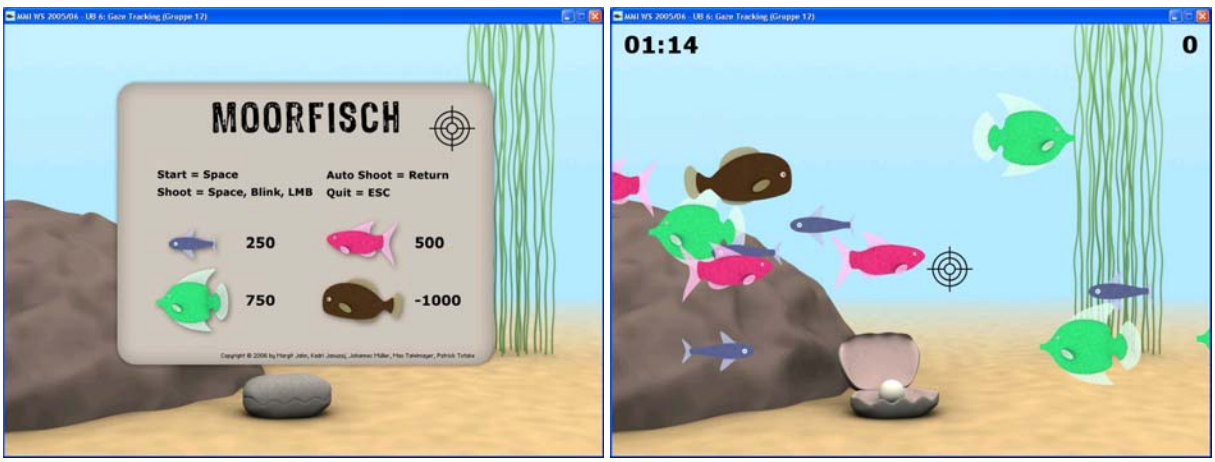
7.4 The Future of Gaze-Aware Systems

The integration of eye tracking with the computer is technically possible. The question remains if and when they will become standard. Whether this will happen depends mainly on the availability of useful concepts for user interaction as they are discussed in this thesis. Several factors influence the future of gaze-aware systems, the availability, the demand, and the costs. These factors influence each other; if the demand grows it will cause more availability at lower costs. In addition, sinking costs can increase the demand. At the moment eye-tracker systems are available but they are not prepared for working as an input device except for accessibility systems which do not bring benefit to the regular users. Most eye trackers existing today serve the purpose of recording and analyzing gaze data and the demand in this market is low compared to the demand for input devices like mouse devices or webcams. Consequently, the prices for eye trackers are still high, again compared to mouse devices or webcams.

The future of gaze-aware systems will depend on an application. The graphical user interface was the application that pushed the mouse device from a special input device for CAD (computer aided design) engineers to an input device for the masses. Although it is possible to direct a graphical user interface solely with the keyboard and without a mouse, in many cases even more efficiently, most people are not able to operate such a system if the mouse is missing or not working. This is the reason for the big success of the mouse device. There is no comparable application for gaze-aware systems on the horizon that everybody wants to have and which does not work well without eye tracker as it was the case for graphical user interface and mouse.

An application that could have the potential to create a mass market for eye-tracking technologies is a computer game. Computer games are a growing market and special input devices for game stations fill the shops. An eye tracker for computer games could come along as a head-mounted device, a headset with earphones and microphone, which are commonly used for gaming already, and with two extra cameras. One camera is mounted near the microphone and focuses on one eye and the other camera next to one earphone with the same view as the eye. The camera at the microphone tracks the eye and because it has a rigid connection with the head, head movements do not influence it. The camera at the earphone sees the display and can calculate the head position from detecting the corners of the display or from matching the known display content to the camera picture. The hardware for such an eye tracker consists of a headset, two webcams and perhaps an infrared LED and all together is available for less than 100 Euro or Dollars. The main costs are the software development and the effort to make it a product. These costs are small per piece if produced in high quantities. The reason why an eye tracker could be successful in the market as input device for computer games lies in the speed. Eye-tracking interaction is fast if the targets are not too small and if an extra input modality is used. For a typical shooting game the targets are big enough in size and the extra input modality is the fire button. While a saving of 300 milliseconds for a pointing operation does not make much difference for a spreadsheet application or a word processor, it makes a big difference for an action game. The excitement and finally the level of adrenalin in the body are directly related to the speed of the game. The experiences with a gaze-aware game "Moorfisch" (inspired from the game "Moorhuhn") written by students as an exercise support this assessment. The task of the game is to shoot different kinds of fishes with the gaze and every shot fish adds points to the score. To make the game more interesting a special type of fish (could be a diver too) results in minus points. Picking up a pearl from a seashell that only opens occasionally gives a bonus. Although the game and the graphics is quite simple compared to the games commercially available, there was big demand for it when presented at the open day of the university.

Figure 81: Screenshots of the Moorfisch game written by students as an exercise based on an interface developed in the course of this thesis.



Research on eye tracker input in first person shooter games [Isokoski, Martin 2006] could not (yet) show an increase in performance compared to classical mouse input. This result is in contradiction to the findings for the "hardware button" of Ware and Mikaelian [Ware, Mikaelian 1987] and the user study presented in 4.4.2 where gaze positioning together with key input were significantly faster than a classical mouse. The master thesis of Jönsson [Jönsson 2005] reports high acceptance for eye trackers as input for computer games and this is in accord with the observations made with the "Moorfisch" game. As the computer game industry always searches for new ideas it is only a question of time until cheap eye trackers for gaming will be in the shops. The availability of cheap eye trackers will lead to the development of further applications for such an eye tracker.

Attention sensors for mobile devices are a further possibility to introduce eye tracking to the mass market. The costs for an attention sensor are negligible and the manufacturers of mobile devices always look for new features to have an advantage in the highly competitive market. Such an attention sensor in a mobile video viewer can provide the functionality of pausing the video when not looking at it. It can also provide a power-saving function by switching off the display when nobody is looking at it. The careful use of energy is very important for mobile devices as the capacity of the batteries is limited. A laptop typically switches off the display after a certain time without key or mouse input. This concept does not work when watching a video as there is no input from the mouse or keyboard. The problem is also well known from mobile MP3 players which try to solve the problem with a HOLD switch. While it seems to be nearly impossible to detect whether somebody listens to audio content an eye tracker can detect whether somebody is watching video content.

A good chance for eye tracking in the smaller high-end market is the trend to large displays. Interaction with large displays or multiple monitor setups by a mouse has problems. One problem is that people cannot find the mouse pointer on the large display area; another problem is how to adjust the control-gain ratio for the mouse. A high gain causes problems for the precision of the mouse movement as explained by Fitts' law while a low gain will lead to mouse movements which exceed the range of the hand. Concepts like focus activation by gaze (as suggested by Vertegaal) or MAGIC pointing (as suggested by Zhai) and preferably MAGIC touch (as suggested here) can help. As large displays are still expensive, the cost for the eye-tracking device does not contribute to the total costs too much. The MAGIC touch principle also saves many hand movements and for this reason helps people who suffer from RSI (repetitive strain injuries). As medical treatment is expensive an eye tracker and a touch-sensitive mouse can be the cheaper alternative.

All these visions of gaze-aware systems could become reality within the next years and some probably will. Prophecies for longer periods are speculations. Nevertheless, it is clear that the evolution of human-computer interfaces will lead to systems that are more 'human' and not to systems where the humans have to act like computers. As the eye gaze is very important for the human-human interaction, it will definitely be very important for the development of future human-computer interfaces.

References

[Abrams, Meyer, Kornblum 1989] Abrams R.A., Meyer D.E., and Kornblum, S. Speed and Accuracy of Saccadic Eye Movements: Characteristics of Impulse Variability in the Oculomotor System. *Journal of Experimental Psychology: Human Perception and Performance* (1989), Vol. 15, No. 3, 529 – 543. (http://www-personal.umich.edu/~kornblum/files/journal_exp_psych_HPP_15-3.pdf)

[Ashdown, Oka, Sato 2005] Ashdown, M., Oka, K., and Sato, Y. Combining Head Tracking and Mouse Input for a GUI on Multiple Monitors. In *Extended Abstracts on Human Factors in Computing Systems,* CHI '05. ACM Press (2005), 1188 – 1191. (http://www.hci.iis.u-tokyo.ac.jp/~ysato/papers/Ashdown-CHI05.pdf)

[Ashmore, Duchowski, Shoemaker 2005] Ashmore, M., Duchowski, A. T., and Shoemaker, G. Efficient Eye Pointing with a Fisheye Lens. In *Proceedings of Graphics interface 2005*. ACM International Conference Proceeding Series, (2005), vol. 112, 203 – 210. (http://portal.acm.org/ft_gateway.cfm?id=1089542&type=pdf&coll=GUIDE&dl=GUIDE&CFID=34761484&CFTOKEN=86712597)

[Atterer 2008] Atterer, R. Usability Tool Support for Model-Based Web Development, PhD Thesis. Ludwig-Maximilian University München, 2008. (http://edoc.ub.uni-muenchen.de/9296/2/Atterer_Richard.pdf)

[Atterer, Schmidt, Wnuk 2007] Atterer, R. Schmidt, A., and Wnuk, M. A. Proxy-Based Infrastructure for Web Application Sharing and Remote Collaboration on Web Pages. In *Proceedings of the 11th IFIP TC13 International Conference on Human-Computer Interaction,* INTERACT 2007, Springer (2007), 74 – 87. (http://www.medien.ifi.lmu.de/pubdb/publications/pub/atterer2007interact/atterer2007interact.pdf)

[Atterer, Wnuk, Schmidt 2006] Atterer, R. Wnuk, M., and Schmidt, A. Knowing the User's Every Move - User Activity Tracking for Website Usability Evaluation and Implicit Interaction. In *Proceedings of the 15th International World Wide Web Conference,* WWW2006, ACM Press (2006), 203 – 212. (http://www.medien.ifi.lmu.de/pubdb/publications/pub/atterer2006www/atterer2006www.pdf)

[Bee, Andre 2008] Bee, N. and Andre, E. Writing with Your Eye: A Dwell Time Free Writing System Adapted to the Nature of Human Eye Gaze. In *Perception in Multimodal Dialogue Systems*, LNCS 5078/2008, Springer (2008), 111 – 122.

[Beymer, Russell 2005] Beymer, D., and Russell, D.M. WebGazeAnalyzer: A System for Capturing and Analyzing Web Reading Behavior Using Eye Gaze. In *Extended Abstracts on Human Factors in Computing Systems,* CHI '05. ACM Press (2005), 1913 – 1916. (http://www.almaden.ibm.com/cs/projects/eyegaze/eyegaze.V2.1.pdf)

[Bolt 1981] Bolt, R.A. Gaze-orchestrated Dynamic Windows. In *Proceedings of the 8th Annual Conference on Computer Graphics and interactive Techniques*, SIGGRAPH '81. ACM Press (1981), 109 – 119.

[Bolt 1982] Bolt, R.A. Eyes at the Interface. In *Proceedings ACM Human Factors in Computer Systems Conference,* ACM Press (1982), 360 – 362.

[Böning, Bartl, Dera, Bardins, Schneider, Brandt 2006] T. Böning, G., Bartl, K., Dera, T., Bardins, S., Schneider, E., and Brandt, T. Mobile Eye Tracking as a Basis for Real-Time Control of a Gaze-Driven Head-Mounted Video Camera. In *Proceedings of the Eye Tracking Research & Applications Symposium*. ETRA 2006, ACM Press (2006), 56. (http://www.forbias.de/papers/boening_etra_2006.pdf)

[Bulling, Ward, Gellersen, Tröster 2008] Bulling, A. Ward, H. Gellersen, and G. Tröster. Robust Recognition of Reading Activity in Transit Using Wearable Electrooculography. In Pervasive Computing 2008, Springer Verlag 2008.

[Bulling, Roggen, Tröster 2008a] Bulling, A., Roggen, D., and Tröster, G. It's in Your Eyes: Towards Context-Awareness and Mobile HCI Using Wearable EOG Goggles. In *Proceedings of the 10th international Conference on Ubiquitous Computing*, UbiComp '08, vol. 344. ACM Press (2008), 84 – 93.

[Bulling, Roggen, Tröster 2008b] Bulling, A., Roggen, D., and Tröster, G. EyeMote - Towards Context-Aware Gaming Using Eye Movements Recorded From Wearable Electrooculography. In *Proc. of the 2nd International Conference on Fun and Games*, FnG, Springer (2008), 33 – 45,

[Campbell, Maglio 2001] Campbell, C. S., and Maglio, P. P. A Robust Algorithm for Reading Detection. In *Proceedings of the 2001 Workshop on Perceptive User interfaces*. PUI '01, vol. 15. ACM Press (2001), 1 – 7.

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

[Card, Moran, Newell 1983] Card, S. K., Moran, T. P., and Newell, A. The Psychology of Human-Computer Interaction, Lawrence Erlbaum, 1983.

[Carpenter 1977] Carpenter, R. H. S. Movement of the Eyes, London: Pion (1977) (as referenced in [Goldstein 2002])

[Chua, Boland, Nisbett 2005] Chua, F. H., Boland J. E. and Nisbett, R. E.: Cultural Variation in Eye Movements During Scene Perception, In *Proceedings of the National Academy of Sciences*, PNAS (2005) 102, 12629 – 12633. (http://www.pnas.org/content/102/35/12629.full.pdf)

[DE102005047160] Vorrichtung, Verfahren und Computerprogramm zum Ermitteln einer Information über eine Form und/oder eine Lage einer Ellipse in einem graphischen Bild. Patentschrift DE102005047160, Internationale Veröffentlichungsnummer WO 2007/039202 A1

[DE 102 51 933 A1] Aufnahmevorrichtung für die kopfgestützte Bilderfassung und Verfahren zur Steuerung der Aufnahmevorrichtung. Offenlegungsschrift DE 102 51 933 A1

[De Luca, Weiss, Drewes 2007] De Luca, A. Weiss, R., and Drewes, H. Evaluation of Eye-Gaze Interaction Methods for Security Enhanced PIN-Entry, In *Proceedings of the 19th Australasian Conference on Computer-Human interaction,* OZCHI 2007. vol. 51. ACM Press (2007), 199 – 202. (http://www.medien.ifi.lmu.de/pubdb/publications/pub/deluca2007ozchi/deluca2007ozchi.pdf)

[Deubel, Schneider 1996] Deubel, H., and Schneider, W. X. Saccade Target Selection and Object Recognition: Evidence for a Common Attentional Mechanism. In *Vision Research*, 36, (1996), 1827–1837. (http://wexler.free.fr/library/files/deubel (1996) saccade target selection and object recognition. evidence for a common attentional mechanism.pdf)

[Dickie, Vertegaal, Sohn, Cheng 2005] Dickie, C., Vertegaal, R., Sohn, C., and Cheng, D. eyeLook: Using Attention to Facilitate Mobile Media Consumption. In *Proceedings of the 18th Annual ACM Symposium on User interface Software and Technology*. UIST '05. ACM Press (2005), 103 – 106. (http://www.hml.queensu.ca/files/papers/2005/Dickie_Vertegaal_Sohn_Cheng_2005.pdf)

[Dix 1998] Dix, A., Finlay, J., Abowd, G., and Beale, R. Human Computer Interaction, 2nd Ed., Prentice Hall Europe, 1998.

[Dodge, Cline 1901] Dodge, R. and Cline, T. S. The angle velocity of eye movements. Psychological Review (1901), 8, 145 – 157. (as referenced in [Jacob, Karn 2003])

[Dongheng, Winfield, Parkhurst 2005] Dongheng Li, Winfield, D. and Parkhurst, D. J. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In *IEEE Computer Society Conference* on *Computer Vision and Pattern Recognition,* Volume 3, (2005), 79. (http://thirtysixthspan.com/openEyes/Li_etal05.pdf)

[Drewes 2010] Drewes, H. Only one Fitts' law formula please! In *Proceedings of the 28th of the international Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '10. ACM Press (2010), 2813 – 2822. (http://www.medien.ifi.lmu.de/pubdb/publications/pub/drewes2010chi/drewes2010chi.pdf)

[Drewes, Atterer, Schmidt 2007] Drewes, H., Atterer, R., and Schmidt, A. Detailed Monitoring of User's Gaze and Interaction to Improve Future E-Learning. In *Proceedings of the 12th International Conference on Human-Computer Interaction* HCII '07, Springer LNCS 4555, (2007), 802 – 811. (http://www.medien.ifi.lmu.de/pubdb/publications/pub/drewes2007hcii/drewes2007hcii.pdf)

[Drewes, De Luca, Schmidt 2007] Drewes, H., De Luca, A., and Schmidt, A. Eye-Gaze Interaction for Mobile Phones. In *Proceedings of the 4th international Conference on Mobile Technology* Mobility '07. ACM Press (2007), 364 – 371. (http://www.medien.ifi.lmu.de/pubdb/publications/pub/deluca2007mobility1/deluca2007mobility1.pdf)

[Drewes, Hußmann, Schmidt 2007] Drewes, H., Hußmann, H., and Schmidt, A. Blickgesten als Fernbedienung, In *Mensch & Computer 2007*. Weimar, Germany, September 2007, Oldenburg-Verlag, ISBN 978-3-486-58496-7, 79 – 88. (http://www.medien.ifi.lmu.de/pubdb/publications/pub/drewes2007muk/drewes2007muk.pdf)

[Drewes, Schmidt 2006] Drewes, H., and Schmidt, A. Mauszeigerpositionierung mit dem Auge, In *Mensch & Computer 2006*, Oldenburg-Verlag, ISBN 978-3-486-58129-4, 185 – 194. (http://www.medien.ifi.lmu.de/pubdb/publications/pub/drewes2006muc/drewes2006muc.pdf)

[Drewes, Schmidt 2007] Drewes, H., and Schmidt, A. Interacting with the Computer using Gaze Gestures. In *Proceedings of Human-Computer Interaction* - INTERACT 2007, Springer LNCS 4663, (2007), 475 – 488. (https://www.medien.ifi.lmu.de/pubdb/publications/pub/drewes2007interact/drewes2007interact.pdf)

[Drewes, Schmidt 2009] Drewes, H., and Schmidt, A. The MAGIC Touch: Combining MAGIC-Pointing with a Touch-Sensitive Mouse. In *Proceedings of Human-Computer Interaction* - INTERACT 2009, Springer LNCS 5727. (2009), 415 – 428. (http://www.medien.ifi.lmu.de/pubdb/publications/pub/drewes2009interact/drewes2009interact.pdf)

[Duchowski 2002] Duchowski, T. D.: Eye Tracking Methodology: Theory and Practice. London: Springer-Verlag (2002), ISBN 1-85233-666-8.

[Eibl-Eibesfeldt 2004] Eibl-Eibesfeldt, I. Die Biologie des menschlichen Verhaltens – Grundriss der Humanethologie. 5.te Auflage 2004, Piper-Verlag 1984, ISBN 3-937501-01-0

[Engbert, Mergenthaler 2006] Engbert, R. and Mergenthaler K. Microsaccades are triggered by low retinal image slip. In *Proceedings of the National Academy of Sciences of the USA* 103. 7192 – 7197.

[EP 1 228 414 B1] Europäische Patentschrift EP 1 228 414 B1

[Fitts 1954] Fitts, P. M. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement, In *Journal of Experimental Psychology* (1954), 47, 381 – 391. (http://e.guigon.free.fr/rsc/article/Fitts54.pdf)

[Fitts, Jones, Milton 1950] Fitts, P. M., Jones, R. E., and Milton, J. L. Eye movements of aircraft pilots during instrument-landing approaches. In *Aeronautical Engineering Review*, (1950), 9(2), 24 – 29. (as referenced in [Jacob, Karn 2003])

[Fono, Vertegaal 2005] Fono, D., and Vertegaal, R. EyeWindows: Evaluation of Eye-Controlled Zooming Windows for Focus Selection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '05. ACM Press (2005), 151 – 160.

[Foo, Ranganath, Venkatesh 2003] Foo Tun Keat, Ranganath, S., and Venkatesh, Y. V. Eye Gaze Based Reading Detection. In *Conference on Convergent Technologies for Asia-Pacific Region*, TENCON 2003. Volume 2, (2003) 825 – 828.

[Goldberg, Richardson 1993] Goldberg, D., and Richardson, C. Touch-Typing With a Stylus. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* CHI '93. ACM Press (1993), 80 – 87.

[Goldstein 2002] Goldstein, E. B. Wahrnehmungspsychologie (Originaltitel: Sensation and Perception) Hrsg. Von Martin Ritter, Heidelberg, Berlin, Spektrum Akademie Verlag 2002

[Hansen, MacKay, Hansen, Nielsen 2004] Hansen, D. W., MacKay, D. J., Hansen, J. P., and Nielsen, M. Eye Tracking off the Shelf. In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*. ETRA '04. ACM Press (2004), 58 – 58.

[Heathcote, Brown, Mewhort 2000] Heathcote, A. Brown, S., and Mewhort D. J. K The Power Law repealed: The case for an Exponential Law of Practice, In *Pyschonomic Bulletin and Review* Vol. 7, Issue 2, (2000), 185 – 207. (http://science-it.newcastle.edu.au/~sdb231/PDFS/exp.pbr.2000.pdf)

[Hennessey, Noureddin, Lawrence 2006] Hennessey, C., Noureddin, B., and Lawrence, P. A Single Camera Eye-Gaze Tracking System with Free Head Motion. In *Proceedings of the 2006 Symposium on Eye Tracking Research & Applications*. ETRA '06. ACM Press (2006), 87 – 94.

[Hinckley, Sinclair 1999] Hinckley, K., and Sinclair, M. 1999. Touch Sensing Input Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* CHI '99. ACM Press (1999), 223 – 230. (http://research.microsoft.com/en-us/um/people/kenh/papers/touch-sensing.pdf)

[Holleis, Otto, Hussmann, Schmidt 2007] Holleis, P., Otto, F., Hussmann, H., and Schmidt, A. 2007. Keystroke-Level Model for Advanced Mobile Phone Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* CHI '07. ACM Press (2007), 1505 – 1514. (https://www.medien.ifi.lmu.de/pubdb/publications/pub/holleis2007chiklm/holleis2007chiklm.pdf)

[Hornof, Cavender, Hoselton 2004] Hornof, A., Cavender, A., and Hoselton, R. EyeDraw: A System for Drawing Pictures with the Eyes. In *Extended Abstracts on Human Factors in Computing Systems*. CHI '04. ACM Press (2004), 1251 – 1254.

[Hull, Neaves, Bedford-Roberts 1997] Hull, R., Neaves, P., and Bedford-Roberts, J. Towards Situated Computing. In *Tech Reports*: HPL-97-66, HP Labs Bristol, 1997.

[Iqbal, Bailey 2004] Iqbal, B., and Bailey P. Using Eye Gaze Patterns to Identify User Tasks, In *Proceedings of The Grace Hopper Celebration of Women in Computing* (2004) (http://interruptions.net/literature/Iqbal-GHC04.pdf)

[Isokoski 2000] Isokoski, P. Text Input Methods for Eye Trackers Using Off-Screen Targets. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*. ETRA '00. ACM Press (2000), 15 – 21. (http://www.cs.uta.fi/~poika/etra_2000_paper/ETRA_2000_final.htm)

[Isokoski, Martin 2006] Isokoski, P., and Martin, B. Eye Tracker Input in First Person Shooter Games. In *Proc. COGAIN 2006*. (2006). (http://www.cs.uta.fi/~poika/cogain2006/cogain2006.pdf)

[Jacob 1990] Jacob, R. J. K. What You Look At is What You Get: Eye Movement-Based Interaction Techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* CHI '90. ACM Press (1990), 11 – 18.

[Jacob 1991] Jacob, R. J. The Use of Eye Movements in Human-Computer Interaction Techniques: What You Look at is What You Get. *ACM Trans. Inf. Syst.* 9, 2 (1991), 152 – 169.

[Jacob 1995] Jacob, R. J. Eye Tracking in Advanced Interface Design. In *Virtual Environments and Advanced interface Design*, W. Barfield and T. A. Furness, Eds. Oxford University Press, New York, (1995), 258 – 288. (http://www.cs.tufts.edu/~jacob/papers/barfield.pdf)

[Jacob, Karn 2003] Jacob, R. J. K., and Karn K. S. Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises (Section Commentary). In *The Mind's Eyes: Cognitive and Applied Aspects of Eye Movements*, ed. by J. Hyona, R. Radach, and H. Deubel, Oxford, Elsevier Science, 2003. (http://www.cs.tufts.edu/~jacob/papers/ecem.pdf)

[Jönsson 2005] Jönsson, E. If Looks Could Kill – An Evaluation of Eye Tracking in Computer Games. Master Thesis (2005), Royal Institute of Technology, Stockholm, Sweden. (http://www.nada.kth.se/utbildning/grukth/exjobb/rapportlistor/2005/rapporter05/jonsson_erika_05125.pdf)

[Kumar, Paepcke, Winograd 2007] Kumar, M., Paepcke, A., and Winograd, T. EyePoint: Practical Pointing and Selection Using Gaze and Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* CHI '07. ACM Press (2007), 421 – 430. (http://hci.stanford.edu/publications/2007/chi253-kumar.pdf)

[Kumar, Garfinkel, Boneh, Winograd 2007] Kumar, M., Garfinkel, T., Boneh, D., and Winograd, T. Reducing Shoulder-surfing by Using Gaze-based Password Entry. In *Proceedings of the 3rd Symposium on Usable Privacy and Security* SOUPS '07, vol. 229. ACM Press (2007), 13 – 19. (http://hci.stanford.edu/research/GUIDe/publications/SOUPS%202007%20-%20Reducing%20Shoulder-surfing%20by%20Using%20Gaze-based%20Password%20Entry.pdf)

[Land 2006] Land, M. F. Eye movements and the control of actions in everyday life. In *Prog Retinal & Eye Res* (2006) 25: 296 – 324. (http://www.cis.rit.edu/pelz/scanpaths/papers/eye-movements-every-day-life-land-2006.pdf)

[Li, Babcock, Parkhurst 2006] Li, D., Babcock, J., and Parkhurst, D. J. openEyes: a low-cost head-mounted eye-tracking solution. In *Proceedings of the 2006 Symposium on Eye Tracking Research & Applications*. ETRA '06. ACM Press (2006), 95 – 100. (http://thirtysixthspan.com/openEyes/li_etal06.pdf)

[MacKenzie 1989] MacKenzie, I. S. A Note on the Information-Theoretic Basis for Fitts' Law. Journal of Motor Behavior (1989), 21, 323 – 330. (http://www.yorku.ca/mack/JMB89.html)

[MacKenzie 1995] MacKenzie, I. S. Movement time prediction in human-computer interfaces. In R. M. Baecker, W. A. S. Buxton, J. Grudin, & S. Greenberg (Eds.), *Readings in human-computer interaction* 2nd ed. Los Altos, CA: Kaufmann (1995), 483 – 493. (reprint of MacKenzie, 1992) (http://www.yorku.ca/mack/GI92.html)

[MacKenzie, Sellen, Buxton 1991] MacKenzie, I. S., Sellen, A., and Buxton, W. A. A comparison of input devices in element pointing and dragging tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '91. ACM Press (1991), 161 – 166.

[MacKenzie, Soukoreff 2003] MacKenzie, I. S. and Soukoreff, R. W. Card, English, and Burr (1978): 25 Years Later. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*. CHI '03. ACM Press, (2003), 760 – 761. (http://www.yorku.ca/mack/chi03a.html)

[Majaranta, Räihä 2002] Majaranta, P., and Räihä, K. Twenty Years of Eye Typing: Systems and Design Issues. In *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*. ETRA '02. ACM Press (2002), 15 – 22. (http://www.cs.uta.fi/ucit/ETRA2002-Majaranta-Raiha.pdf)

[Majaranta, Aula, Räihä 2004] Majaranta, P., Aula, A., and Räihä, K. Effects of Feedback on Eye Typing with a Short Dwell Time. In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*. ETRA '04. ACM Press (2004), 139 – 146. (http://www.cs.uta.fi/%C2%A0curly/publications/ETRA2004-Majaranta.pdf)

[Majaranta, MacKenzie, Aula, Räihä 2003] Majaranta, P., MacKenzie, I. S., Aula, A., and Räihä, K. Auditory and visual feedback during eye typing. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*. CHI '03. ACM Press (2003), 766 – 767. (http://www.cs.uta.fi/~curly/publications/CHI2003-Majaranta.pdf)

[Mankof, Abowd 1998] Mankof, J., and Abowd, G. D. Cirrin: a word-level unistroke keyboard for pen input. In *Proceedings of the 11th Annual ACM Symposium on User interface Software and Technology* UIST '98, ACM Press (1998), 213 – 214. (http://www.cc.gatech.edu/fce/pendragon/publications/cirrin98.pdf)

[McGuffin, Balakrishnan 2002] McGuffin, M. and Balakrishnan, R. Acquisition of Expanding Targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* CHI '02. ACM Press (2002), 57 – 64. (http://www.dgp.toronto.edu/~mjmcguff/research/expandingTargets/AcquisitionOfExpandingTargets.pdf)

[Merten, Conati 2006] Merten, C., and Conati, C. Eye-Tracking to Model and Adapt to User Meta-cognition in Intelligent Learning Environments. In *Proceedings of the 11th international Conference on intelligent User interfaces*. IUI '06. ACM Press (2006), 39 – 46. (http://www.cs.ubc.ca/~conati/my-papers/IUI06eyetrackingCamera.pdf)

[Milekic 2003] Milekic, S. The More You Look the More You Get: Intention-based Interface using Gaze-tracking. In *Museums and the Web 2002* Selected Papers from an International Conference, Archives & Museum Informatics, D. Bearman, and Trant, J. (eds.), Pittsburgh, PA (2003) (http://www.uarts.edu/faculty/smilekic/paper/milekicMW03.pdf)

[Miniotas 2000] Miniotas, D. Application of Fitts' Law to Eye Gaze Interaction. In *CHI '00 Extended Abstracts on Human Factors in Computing Systems*. CHI '00. ACM Press (2000), 339 – 340.

[Miniotas, Spakov, Evreinov 2003] Miniotas, D., Špakov, O., and Evreinov, G. Symbol Creator: An Alternative Eye-Based Text Entry Technique With Low Demand For Screen Space. In *Proceedings of INTERACT 2003*, Pub. IOS Press, IFIP, (2003), 137 – 143. (http://www.idemployee.id.tue.nl/g.w.m.rauterberg/conferences/INTERACT2003/INTERACT2003-p137.pdf)

[Miniotas, Špakov, MacKenzie 2004] Miniotas, D., Špakov, O., and MacKenzie, I. S. Eye Gaze Interaction with Expanding Targets. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*. CHI '04. ACM Press (2004), 1255 – 1258. (http://www.yorku.ca/mack/chi04b.pdf)

[Miniotas, Špakov, Tugoy, MackKenzie 2005] Miniotas, D., Špakov, O., Tugoy, I., and MackKenzie, I. S. Extending the limits for gaze pointing through the use of speech. *Information Technology and Control*, (2005) 34, 225 – 230. (http://itc.ktu.lt/itc343/Miniotas343.pdf)

[Morimoto, Amir, Flickner 2002] Morimoto, C. H., Amir, A., and Flickner, M. Free Head Motion Eye Gaze Tracking Without Calibration. In *Extended Abstracts on Human Factors in Computing Systems* CHI '02. ACM Press (2002), 586 – 587.

[Morimoto, Koons, Amir, Flickner 1999] Morimoto, C. H., Koons, D., Amir, A., and Flickner M. Frame-Rate Pupil Detector and Gaze Tracker. In *Proceedings of the IEEE ICCV'99 frame-rate workshop*. (1999). (http://www.ime.usp.br/~hitoshi/framerate/framerate.ps.gz)

[Morimoto, Koons, Amir, Flickner 2000] Morimoto C. H., Koons D., Amir A., and Flickner M. Pupil Detection and Tracking Using Multiple Light Sources. In *Image and Vision Computing*, Volume 18, Number 4, 1. Elsevier (2000), 331 – 335.

[Nguyen, Wagner, Koons, Flickner 2002] Nguyen, K., Wagner, C., Koons, D., and Flickner, M. Differences in the Infrared Bright Pupil Response of Human Eyes. In *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*. ETRA '02. ACM Press (2002), 133 – 138.

[Noris, Benmachiche, Billard 2008] Noris, B., Benmachiche, K., and Billard, A. Calibration-Free Eye Gaze Direction Detection with Gaussian Processes, In *Proceedings of the International Conference on Computer Vision Theory and Applications*, VISAPP 08, (2008) (http://infoscience.epfl.ch/record/114053/files/visapp08-basilio_noris.pdf)

[Ohno, Mukawa, Yoshikawa 2002] Ohno, T., Mukawa, N., Yoshikawa, A. FreeGaze: A Gaze Tracking System for Everyday Gaze Interaction. In *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*. ETRA '02. ACM Press (2002), 125 – 132. (http://www.brl.ntt.co.jp/people/takehiko/papers/etra2002.pdf)

[Ohno, Mukawa 2004] Ohno, T., and Mukawa, N. A Free-head, Simple Calibration, Gaze Tracking System That Enables Gaze-Based Interaction. In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*. ETRA '04. ACM Press (2004), 115 – 122. (http://www.brl.ntt.co.jp/people/takehiko/papers/etra2004.pdf)

[Pan, Hembrooke, Gay, Granka, Feusner, Newman 2004] Pan, B., Hembrooke, H. A., Gay, G. K., Granka, L.A., Feusner, M. K., and Newman, J. K. 2004. The Determinants of Web Page Viewing Behavior: An Eye-Tracking Study. In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*. ETRA '04. ACM Press (2004), 147 – 154. (http://www.hci.cornell.edu/projects/pdfs%20of%20pubs/ETRA04.pdf)

[Porta, Turina 2008] Porta, M., and Turina, M. Eye-S: a Full-Screen Input Modality for Pure Eye-based Communication. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*. ETRA '08. ACM Press (2008), 27 – 34.

[Qvarfordt, Zhai 2005] Qvarfordt, P. and Zhai, S. Conversing with the User Based on Eye-Gaze Patterns. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* CHI '05. ACM Press (2005), 221 – 230. (http://www.almaden.ibm.com/u/zhai/papers/iTouristQvarfordtZhaiCHI2005.pdf)

[Ravyse, Sahli, Reinders, Cornelis 2000] Ravyse, I., Sahli, H., Reinders, M. J. T., and Cornelis, J. Eye Activity Detection and Recognition Using Morphological Scale-Space Decomposition. In *Proceedings of the international Conference on Pattern Recognition* ICPR. IEEE Computer Society, Volume 1, (2000), 1080 – 1083. (http://csdl.computer.org/comp/proceedings/icpr/2000/0750/01/07505080.pdf)

[Ravyse, Reinders, Cornelis, Sahli 2000] Ravyse, I., Reinders, M. J. T., Cornelis, J., and Sahli, H. Eye Gesture Estimation; IEEE Benelux Signal Processing Chapter, Signal Processing Symposium, SPS2000 (2000), published on proceedings cd-rom (http://www.etro.vub.ac.be/Personal/icravyse/Ilse_Old/sps2000_publ.pdf)

[Reichle, Pollatsek, Fisher, Rayner 1998] Reichle, E. D., Pollatsek, A., Fisher, D. L., and Rayner, K. Toward a Model of Eye Movement Control in Reading. In *Psychological Review, 105*, (1998), 125 – 157. (http://www.ling.ohio-state.edu/~speer/ling871et/readings/Reichle_etal98.pdf)

[Salvucci, Goldberg 2000] Salvucci, D. D., and Goldberg, J. H. Identifying Fixations and Saccades in Eye-Tracking Protocols. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*. ETRA '00. ACM Press (2000), 71 – 78. (http://www.cs.drexel.edu/~salvucci/publications/ETRA00b.pdf)

[Salvucci, Anderson 2000] Salvucci, D. D., and Anderson, J. R. Intelligent Gaze-Added Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '00. ACM Press (2000), 273 – 280. (http://www.cs.drexel.edu/~salvucci/publications/CHI00.pdf)

[Santella, Agrawala, DeCarlo, Salesin, Cohen 2006] Santella, A., Agrawala, M., DeCarlo, D., Salesin, D., and Cohen, M. Gaze-Based Interaction for Semi-Automatic Photo Cropping. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* CHI '06. ACM Press (2006), 771 – 780. (http://research.microsoft.com/en-us/um/people/cohen/gazebasedcropping_chi.pdf)

[Schmidt 2002] Schmidt, A. Ubiquitous Computing – Computing in Context, PhD Thesis. Lancaster University, 11/2002. (http://www.comp.lancs.ac.uk/~albrecht/phd/Albrecht_Schmidt_PhD-Thesis_Ubiquitous-Computing_ebook1.pdf)

[Schneider 2004] Schneider, E. Torsionelle Augenbewegungen bei galvanischen, natürlichen und pathologischen Reizzuständen des menschlichen Gleichgewichtssystems: Grundlagen, Methoden und klinische Anwendung, Dissertation in Humanbiologie an der Medizinischen Fakultät der Ludwig-Maximilians-Universität zu München, 2004 (http://edoc.ub.uni-muenchen.de/2517/1/Schneider_Erich.pdf)

[Shannon 1948] Shannon, C. E. A Mathematical Theory of Communication, Bell System Technical Journal, vol. 27, (1948) 379–423, 623 – 656. (http://plan9.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf)

[Shell, Vertegaal, Cheng, Skaburskis, Sohn, Stewart, Aoudeh, Dickie 2004] Shell, J.S., Vertegaal, R., Cheng, D., Skaburskis, A. W., Sohn, C., Stewart, A. J., Aoudeh, O., and Dickie, C. ECSGlasses and EyePliances: Using Attention to Open Sociable Windows of Interaction. In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*. ETRA '04. ACM Press (2004), 93 – 100.

[Shih, Wu, Liu 2000] Shih, S.-W., Wu, Y.-T., and Liu, J. A Calibration-Free Gaze Tracking Technique. In *15th International Conference on Pattern Recognition* ICPR 2000. IEEE Computer Society (2000), Vol. 4, 201 – 204.

[Sibert, Gokturk, Lavine 2000] Sibert, J. L., Gokturk, M., and Lavine, R. A. The Reading Assistant: Eye Gaze Triggered Auditory Prompting for Reading Remediation. In *Proceedings of the 13th Annual ACM Symposium on User interface Software and Technology*. UIST '00. ACM Press (2000), 101 – 107. (http://www.alia.ir/school_files/EyeTracking/p101-sibert.pdf)

[Sibert, Jacob 2000] Sibert, L. E., and Jacob, R. J. Evaluation of Eye Gaze Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing*. CHI '00. ACM Press (2000), 281 – 288. (http://www.cs.tufts.edu/~jacob/papers/chi00.sibert.pdf)

[Sibert, Templeman, Jacob 2000] Sibert, L. E., Templeman, J. N., and Jacob, R. J. K. Evaluation and Analysis of Eye Gaze Interaction. NRL Report, Naval Research Laboratory, Washington, D.C., 2000. (http://www.eecs.tufts.edu/~jacob/papers/nrlreport.pdf)

[Sobotta 2004] Sobotta, J. Atlas der Anatomie des Menschen, Hrsg. R. Putz und R. Pabst, Elsevier GmbH München, 21. Auflage, 2004

[Špakov, Miniotas 2005] Špakov, O., and Miniotas, D. Gaze-Based Selection of Standard-Size Menu Items. In *Proceedings of the 7th international Conference on Multimodal interfaces*. ICMI '05. ACM Press (2005), 124 – 128.

[Tanriverdi, Jacob 2000] Tanriverdi, V., and Jacob, R. J. Interacting with eye movements in virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '00. ACM Press (2000), 265 – 272.

[Thomae, Plagwitz, Husar, Henning 2002] Thomae, N., Plagwitz, K.-U., Husar, P., and Henning, G. Hough-Transformation zur Bildverarbeitung bei der Blickrichtungsbestimmung. In *Proceedings 36. Jahrestagung der Deutschen Gesellschaft für Biomedizinische Technik.* Biomedizinische Technik, Band 47, Ergänzungsband 1, Teil 2, (2002), 636 – 638.

[US 6578962] Calibration-free eye gaze tracking. US Patent Issued on June 17, 2003

[Vertegaal 2008] Vertegaal, R. A Fitts Law comparison of eye tracking and manual input in the selection of visual targets. *Conference on Multimodal interfaces* IMCI '08. ACM (2008.) 241 – 248.

[Vertegaal, Dickie, Sohn, Flickner 2002] Vertegaal, R., Dickie, C., Sohn, C., and Flickner, M. Designing Attentive Cell Phone Using Wearable EyeContact Sensors. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems*. CHI '02. ACM Press (2002), 646 – 647.

[Vertegaal, Mamuji, Sohn, Cheng 2005] Vertegaal, R., Mamuji, A., Sohn, C., and Cheng, D. Media EyePliances: Using Eye Tracking For Remote Control Focus Selection of Appliances. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*. CHI '05. ACM Press (2005), 1861 – 1864.

[Vertegaal, Slagter, van der Veer, Nijholt 2001] Vertegaal, R., Slagter, R., van der Veer, G., and Nijholt, A. Eye Gaze Patterns in Conversations: There is More to Conversational Agents than Meets the Eyes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '01. ACM Press (2001), 301 – 308.
(http://www.cs.queensu.ca/~roel/publications/2001/CHI2001%20Gaze%20Conversations.pdf)

[Wade, Tatler, Heller 2003] Wade N. J., Tatler B. W., and Heller D. Dodge-ing the issue: Dodge, Javal, Hering, and the measurement of saccades in eye-movement research. Perception (2003) 32: 793 – 804. (as referenced in [Schneider@])

[Wagner, Bartl, Günthner, Schneider, Brandt, Ulbrich 2006] Wagner, P., Bartl, K., Günthner, W., Schneider, E., Brandt, T., and Ulbrich, H. A Pivotable Head Mounted Camera System that is Aligned by Three-Dimensional Eye Movements. In *Proceedings of the 2006 Symposium on Eye Tracking Research & Applications*. ETRA '06. ACM Press (2006), 117 – 124.

[Wang, Zhai, Su 2001] Wang, J., Zhai, S., and Su, H. Chinese Input with Keyboard and Eye-Tracking - An Anatomical Study. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '01. ACM Press (2001), 349 – 356.

[Ware, Mikaelian 1987] Ware, C., and Mikaelian, H. H. An Evaluation of an Eye Tracker as a Device for Computer Input. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics interface* CHI '87. ACM Press (1987), 183 – 188

[Welford 1968] Welford, A. T. *Fundamentals of skill*. London: Methuen (1968) as referenced in [MacKenzie 1989].

[Wobbrock, Myers 2006] Wobbrock, J., and Myers, B. Trackball Text Entry for People with Motor Impairments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* CHI '06. ACM Press (2006), 479 – 488. (http://faculty.washington.edu/wobbrock/pubs/chi-06.pdf)

[Wobbrock, Myers, Kembel 2003] Wobbrock, J. O., Myers, B. A., and Kembel, J. A. EdgeWrite: A Stylus-Based Text Entry Method Designed for High Accuracy and Stability of Motion. In *Proceedings of the 16th Annual ACM Symposium on User interface Software and Technology.* UIST '03. ACM Press (2003), 61 – 70. (http://faculty.washington.edu/wobbrock/pubs/uist-03.pdf)

[Wobbrock, Rubinstein, Sawyer, Duchowski 2007] Wobbrock, J. O., Rubinstein, J., Sawyer, M., and Duchowski, A. T. Not Typing but Writing: Eye-based Text Entry Using Letter-like Gestures. In *Proceedings of COGAIN.* (2007), 61 – 64. (http://faculty.washington.edu/wobbrock/pubs/cogain-07.pdf)

[Wobbrock, Rubinstein, Sawyer, Duchowski 2008] Wobbrock, J. O., Rubinstein, J., Sawyer, M. W., Duchowski, A. T. Longitudinal Evaluation of Discrete Consecutive Gaze Gestures for Text Entry. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications.* ETRA '08. ACM Press (2008), 11 – 18. (http://faculty.washington.edu/wobbrock/pubs/etra-08.pdf)

[Ying-li, Kanade, Cohn 2000] Ying-li Tian, Kanade, T., and Cohn, J. F. Dual-State Parametric Eye Tracking, In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*. (2000), 110 – 115. (http://www.ri.cmu.edu/pub_files/pub2/tian_ying_li_2000_3/tian_ying_li_2000_3.pdf)

[Zhai 2004] Zhai, S. Characterizing computer input with Fitts' law parameters – The information and non-information aspects of pointing. *Int. J. Hum.-Comput. Stud*. 61, 6 (2004), 791 – 809. (http://www.almaden.ibm.com/u/zhai/papers/FittsThroughputFinalManuscript.pdf)

[Zhai, Conversy, Beaudouin-Lafon, Guiard 2003] Zhai, S., Conversy, S., Beaudouin-Lafon, M., Guiard, Y. Human On-line Response to Target Expansion. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '03. ACM Press (2003), 177 – 184. (http://www.almaden.ibm.com/u/zhai/papers/TargetExpansion.pdf)

[Zhai, Morimoto, Ihde 1999] Zhai, S., Morimoto, and C., Ihde, S. Manual And Gaze Input Cascaded (MAGIC) Pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* CHI '99. ACM Press (1999), 246 – 253. (http://www.almaden.ibm.com/u/zhai/papers/magic/magic.pdf)

[Zhang, Imamiya, Go, Mao 2004] Zhang, Q., Imamiya, A., Go, K., and Mao, X. Resolving Ambiguities of a Gaze and Speech Interface. In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications.* ETRA '04. ACM Press (2004), 85 – 92

[Zhang, MacKenzie 2007] Zhang, X., MacKenzie, I. S. Evaluating Eye Tracking with ISO 9241 – Part 9. *Proceedings of HCI International 2007*, Heidelberg: Springer (2007), 779 – 788. (http://www.yorku.ca/mack/hcii2007.html)

[Zhu, Fujimura, Ji 2002] Zhu, Z., Fujimura, K., and Ji, Q. Real-Time Eye Detection and Tracking Under Various Light Conditions. In *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*. ETRA '02. ACM Press (2002), 139 – 144. (www.ecse.rpi.edu/homepages/qji/Papers/AcmPaper.pdf)

Web References

Last date of checking availability and content: 24th July 2009

[Arrington@]	http://www.arringtonresearch.com
[COGAIN Bibliography@]	http://www.cogain.org/bibliography/
[COGAIN Eyetrackers@]	http://www.cogain.org/eyetrackers/eyetrackers-for-eye-movement-research
[COGAIN Low cost@]	http://www.cogain.org/eyetrackers/low-cost-eye-trackers
[Chronos Vision@]	http://www.chronos-vision.de/scleral-search-coils
[Deubel@]	http://www.paed.uni-muenchen.de/~deubel/
[Dix@]	http://www.hcibook.com/e3/online/fitts-cybernetic/
[Buxton@]	http://www.billbuxton.com/fitts93.html
[ERICA@]	http://www.eyeresponse.com
[ETRA@]	http://www.e-t-r-a.org/
[Schneider@]	http://www.forbias.de/papers/dgkn06.pdf
[Schneider2@]	http://www.forbias.de/papers/poster_sfn05.pdf
[ISO 9241-9@]	http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=30030
[interactive-minds@]	http://www.interactive-minds.com/en/eye-tracker/eyefollower
[Jacob@]	http://www.cs.tufts.edu/~jacob/papers/
[MacKenzie@ 1995]	http://www.yorku.ca/mack/GI92.html
[MacKenzie@ 2002]	http://www.yorku.ca/mack/RN-Fitts_bib.htm
[MacKenzie@ 2007]	http://www.yorku.ca/mack/hcii2007.html
[metrovision@]	http://www.metrovision.fr/mv-po-notice-im18.jpg
[mouse gestures@]	http://optimoz.mozdev.org/gestures
[Nielson@]	http://www.useit.com/alertbox/reading_pattern.html
[OpenEyes@]	http://thirtysixthspan.com/openEyes/
[OpenEyes Tech. Report@]	http://thirtysixthspan.com/openEyes/ISU-HCI-2005-04.pdf
[OpenEyes Construct@]	http://thirtysixthspan.com/openEyes/openEyes_ConstructingaLow-CostMobileEyeTracker.pdf
[OpenEyes Starburst@]	http://thirtysixthspan.com/openEyes/starburst.pdf

[opengazer@] http://www.inference.phy.cam.ac.uk/opengazer/

[opengazer sourceforge@] http://sourceforge.net/projects/opengazer

[Qprox@] http://www.qprox.com/

[SMI@] http://www.smivision.com/

[Tobii@] http://www.tobii.com

[xuuk@] https://www.xuuk.com/Images/eyebox2productspecs.pdf

Acknowledgements

First of all I thank Professor Heinrich Hußmann (Ludwig-Maximilians-University of Munich) for accepting me as his student without knowing me before. I am sincerely grateful for his constant and great support.

I thank Professor Albrecht Schmidt (University of Duisburg-Essen) for guiding me on the path of science. He motivated me and gave me many inspirations. He taught me how to write and submit scientific papers.

I thank Professor Allan Dix (University of Lancaster) for being the external supervisor of this work. Our discussion on Fitts' law was very helpful.

I thank Jamie Ward and Tobias Landes who spent several days each to improve my English and sharpen my writing. All readers of this thesis will appreciate it.

Last not least, I thank all my colleges for the nice atmosphere at the department, interesting discussions at the coffee machine and joining me travelling to conferences. In particular, I thank Richard Atterer, Alexander De Luca, and Roman Weiss with whom I published papers together.