TEKNILLINEN KORKEAKOULU
HELSINKI UNIVERSITY OF TECHNOLOGY

Kristian Lukander

Mobile usability - Measuring gaze point on handheld devices

Master's thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology

Helsinki, 01.12.2003

Supervisor Professor Marko Nieminen
Instructor M.Sc. Jussi Virkkala

HELSINKI UNIVERSITY OF TECHNOLOGY	ABSTRACT OF MASTER'S THESIS
Author:	Kristian Lukander
Title of Thesis:	Mobile usability – Measuring gaze point on handheld devices
Date:	01.12.2003
Number of pages:	99
Department:	Department of Automation and Systems Technology
Professorship:	T-121 User-centered Product Development
Supervisor:	Professor Marko Nieminen
Instructor:	M.Sc. Jussi Virkkala

Eye tracking methods offer a valuable addition to the toolkit of usability research and analysis. They provide sensitive and accurate information on the direction of visual attention, information processing, and strategies for visual search and information collection methods of the user during a task. These measures are not available through the traditional methods of usability research.

Current eye tracking methods tacitly assume the stabilization of the head of the user and the user interface during measurement, resulting in non-realistic scenarios and the study of gaze and eye movements in exclusion of the natural head, hand, and body movements.

The rapid increase in the number of mobile devices with complex user interfaces has resulted in the research methods lagging behind the development. This is probably partially due to the lack of proper equipment for studying small screen interfaces.

This thesis develops a prototype of a novel research system for tracking the gaze point of a user while using a mobile, handheld device, without placing restrictions on the natural movements of the user. The system is implemented as a software prototype, integrating a video-oculography device with a magnetic positional tracker.

The results of the evaluation tests suggest, that the prototype system is capable of tracking the gaze with an accuracy of one degree of visual angle without limiting the users natural movements.

Keywords:	User interfaces, usability, eye movements, gaze tracking, mobile devices, handheld devices, small screen user interfaces

TEKNILLINEN KORKEAKOULU	DIPLOMITYÖN TIIVISTELMÄ
Tekijä:	Kristian Lukander
Diplomityön nimi:	Katseen paikan seuraaminen mobiililaitteen näytöllä
Päivämäärä:	01.12.2003
Sivujen lukumäärä:	99
Osasto:	Automaatio- ja systeemitekniikan osasto
Professuuri:	T-121 Käyttäjäkeskeinen tuotekehitys
Valvoja:	Professori Marko Nieminen
Ohjaaja:	FM Jussi Virkkala

Silmänliikkeiden mittausmenetelmät tarjoavat houkuttelevan lisän käytettävyystutkimuksen työkalupakkiin. Nämä mahdollistavat herkempien ja tarkempien mittareiden käytön käyttäjän tehtävänaikaisen visuaalisen attention, informaation prosessoinnin, sekä visuaalisen haun ja informaation keräämisen strategioiden tutkimiseen. Näitä mittareita ei pystytä korvaamaan käytettävyystutkimuksen perinteisillä menetelmillä.

Nykyiset silmänliikkeiden mittausmenetelmät vaativat oletusarvoisesti pään ja käyttöliittymän pitämistä paikallaan mittauksen aikana. Oletus johtaa epärealistisiin koeasetelmiin sekä silmänliikkeiden tutkimiseen irrallaan luonnollisista pään, käsien sekä vartalon liikkeistä.

Pienillä käyttöliittymillä varustettujen mobiililaitteiden lukumäärän nopea kasvu on johtanut tilanteeseen, jossa tutkimus ei ole enää kehityksen tasalla. Syynä tähän on todennäköisesti osaltaan puute pienten käyttöliittymien tutkimiseen sopivista laitteistoista.

Tämän diplomityön tavoitteena oli kehittää uudenlaisen tutkimusjärjestelmän prototyyppi käyttäjän katseen paikan seuraamiseen mobiilinäytöllä, asettamatta rajoituksia käyttäjän luonnolliselle liikkumiselle mittauksen aikana. Järjestelmä on toteutettu ohjelmistona, joka yhdistää video-okulografialaitteen magneettiseen paikantimeen.

Suoritettujen evaluointitestien tulokset osoittavat, että järjestelmän prototyyppi pystyy seuraamaan katseen paikkaa yhden visuaalisen kulman asteen tarkkuudella sallien käyttäjän luonnollisen liikkumisen mittausten aikana.

Avainsanat: Käyttöliittymät, käytettävyys, silmänliikkeet, katseen paikan seuranta, mobiililaitteet, pienen ruudun käyttöliittymät

Preface

This thesis was written at the Brainwork Laboratory of the Finnish Institute of Occupational Health as part of a project partially funded by the National Technology Agency (Tekes).

I would like to thank my instructor Jussi Virkkala M.Sc. for his help and support during the project, and my supervisor, Professor Marko Nieminen for his constructive feedback.

I would also like to extend my gratitude to Kiti Müller Ph.D. for her constant support and optimism. Additionally, I owe to the people sharing the 6th floor workroom with me (Kati, Tomas, Markku, and Mika), for keeping up the spirit with the proper mixture of irony and encouragement. A big thank you goes to Elina Nygrén for making the sacrifice of posing for the illustrations in this thesis, while wearing the not so flattering test equipment.

I am also sincerely grateful to Kiti Müller, Jussi Virkkala, and Ronald Lukander for their thorough comments on my work, helping me structure and sharpen my writing.

Finally, my love and gratitude to my fiancée Sanna Ilvonen, for sharing my troubles and enduring the long nights of frantic clickety-clack on the keyboard.

Writing this thesis took the better part of the year 2003. My almost favourite quote by Douglas Adams aptly sums it up: “I love deadlines. I love the whooshing sound they make as they fly by.” In contrast to this, however, I finally did meet the deadline. Hooray for me!

Helsinki, 01.12.2003

Kristian Lukander

Table of contents

List of abbreviations

3D	Three-dimensional
AC	Alternating current
API	Application programming interface
ASCII	American Standard Code of Information Interchange. A 7-bit character set of 128 characters for exchanging information.
DOF	Degrees-of-freedom
EOG	Electro-oculography
fMRI	Functional Magnetic Resonance Imaging
IDE	Integrated Development Environment
IR	Infra-red
IROG	Infra-red oculography
LED	Light emitting diode
OOP	Object oriented programming
PDA	Personal digital assistant
RMS	Root mean square
RS232	Recommended Standard 232 (computer serial interface, IEEE)
UI	User interface
VOG	Video-oculography

List of figures

In appendices:

1 Introduction

A considerable part of information work is already being done on mobile, handheld devices, equipped with screens, whose size is only a fraction of the current desktop standard. The trend of future work – and play – seems to be towards the mobile with the rapid increase of mobile devices, such as phones, personal digital assistants (PDAs), and laptop computers. Portable displays and use "on the move" have created a whole new way of using and perceiving the user interface.

These devices are crammed with features from phones to calendars and from games to digital cameras, and need good user interfaces to penetrate the market beyond technology enthusiasts. Studies in enterprises show that the economic viability of a software [implies hardware as well] system depends on its ergonomic design (Zülch & Stowasser, 1994).

However, the research of the implications of this development is lagging behind. This is probably partly because of the tendency of the academic world to avoid applied research and partly because of the lack of proper equipment for studying mobile user interfaces and use scenarios. (Kuutti, 2000)

In natural environments, eye movements are made toward task-relevant targets even when high spatial resolution is not required. Such attentional eye movements, made without conscious intervention, can reveal attentional mechanisms and provide a window into cognition. Thus, monitoring observers' eye movements during a task can provide a tool to better understand visual perception. (Pelz et al., 2000)

A large body of research on perception and eye movements, affecting the design of user interfaces, is available from the basic oculomotor search to, for instance, research on visual search, or contrast and color sensitivity. However, these findings cannot be directly extended to the realm of mobile computing. In contrast to work in the office, the moving display, display size, differences in input devices, and changing lighting conditions, all contribute to the currently unknown factors affecting user performance and user experience on mobile devices.

Visual perception is an inherently complex task, yet the majority of studies in the past were undertaken with subjects performing relatively simple tasks under reduced laboratory conditions (Pelz et al., 2000). So far, researchers have been content with (and have actually favoured) the study of eye movements in isolation, with the exclusion of head movements. This has partly been inspired by a reductionist attitude, but even more dictated by the available equipment.

The direction of gaze is determined by orienting the eyes in the head, and the head in space. Despite this, most studies performed measuring eye movements have required immobilizing the head using a headrest or a bite bar. This may largely be due to the fact that most current techniques for measuring eye movements rely on the measurement of visual angle, where it is often tacitly assumed that the head is located at a fixed distance, and usually also perpendicular, to the stimulus screen (Duchowski et al., 2002).

Although there will be a continued need for further study of aspects of oculomotor control under strictly controlled and isolated conditions, oculomotor research is evolving to broader questions, in which eye movements are investigated in the context of general visuomotor behaviour, including head, hand, and body movements in natural tasks (Collewijn, 1999).

So far, there exists only a small body of usability research performed on mobile devices. An extensive literature search using the individual keywords *mobile*, *handheld* and *small screen* together with the keywords *usability*, *gaze*, *gaze tracking* and *eye movement* led to the discovery of a handful of articles on mobile usability issues, web design issues considering small screen displays, presenting dynamic content on small screen devices, rapid serial visual presentation (RSVP), and using context-dependent information to guide the user interface. None of these articles included the use of eye movement or gaze position measurements.

This thesis is a multidisciplinary study in the context of usability, cognitive technology and applied biophysics. It gives a definition of an eye tracking device suitable for tracking the gaze point on a mobile user interface in usability studies. It also includes a description of a prototype tracker aiming to repond to the requirements for such a system. The future applications of the device include both usability studies, as well as, more medically oriented studies in vision sciences.

The initial idea for this project came from several presentations and articles on eye movement and usability research, stating the need for a more sophisticated, accurate and suitable approach for studying the ever increasing number of mobile devices and their user interfaces. There is a genuine need for equipment for studying the usability of mobile user interfaces. This thesis aims at responding to that need.

1.1 Goals of research

This thesis has two objectives:

1. To recognize the needs and define requirements of a system that extends usability studies based on gaze tracking to the realm of mobile computing, and
2. To design and implement a working prototype of such a system.

The implemented system should respond to the needs and requirements defined. The system should be capable of recording gaze data on a number of mobile user interfaces for studying usability aspects and information retrieval on the given interface, on natural use scenarios and use positions for the user.

The resulting system will be used in future research projects of mobile devices in the Brainwork laboratory of the Finnish Institute of Occupational Health.

1.2 Thesis organization

This thesis is organized as follows. Chapter 2 describes the conceptual background of the thesis. The chapter reviews the basics of the human visual system, eye movements, eye movement tracking methodology, and positional tracking methodology.

Chapter 3 gives a definition for a system for tracking gaze point on a mobile, small screen user interface in usability studies. The definition is based on a review of the literature found on the subject and the results of a questionnaire performed during the project.

Chapter 4 describes the structure and implementation of the developed tracker prototype. It includes descriptions of the principle of operation and technological components used. The chapter explains the implementation and includes screenshots and a demonstration version of the tracker application.

Chapter 5 describes the evaluation of the tracker prototype. The performed tests and the produced data is reviewed. Chapter 6 summarizes the results and includes error considerations for the prototype.

Chapter 7 presents the conclusions and discussion. It also gives viewpoints on future research and the future development of the prototype.

2 Conceptual background

2.1 The human visual system

This chapter reviews the human visual system at a general level. The main features, interesting from the viewpoint of eye movement, usability research, and user interface design, are described.

2.1.1 Anatomy and physiology of the eye

The human eye is a fragile extension of the brain, encased and protected by the facial bones of the skull. Due to the separation of the eyes, each eye is able to see further around an object in front of it than the opposite eye. The brain receives two slightly different images of the same object and superimposes them to derive distance or depth and the perception of a three dimensional world. The distance between the left and right eye is approximately 6-7 centimeters, depending on the individual.

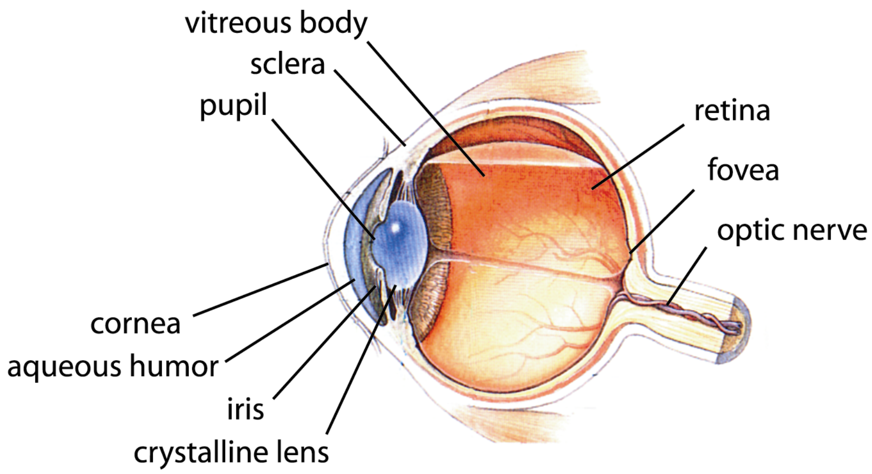


Figure 1 The structure of the eye

The anatomy of the eye is presented in Figure 1. A hard, transparent layer called the *cornea* forms the front of the eyeball. Surrounding this is the opaque *sclera*, inside of which the blood vessels form the *choroid*. On the front side, under the cornea, the *iris* extends the choroid. The iris has a round aperture in the middle called *pupil*. The eyeball is filled by a glasslike humor, the *vitreous body*. And finally, the inside of the choroid holds the *retina*. (Haines, 1997)

The iris is responsible for regulating the amount of light that is admitted on the retina, and does this by expanding and contracting the pupil. Behind the pupil lies a soft membrane, the *crystalline lens*, responsible for accommodating and focusing the image on the retina. The retina, in turn, is responsible for transforming the received image or visual stimuli to electric signals, and passing them on via the *optic nerve* to the *visual cortex*, located in the occipital lobes of the brain, see chapter 2.1.2.

The axons of the receptor cells of the retina coil together at a single spot before they exit the back of the eye through an area called the *optic disk*. Since no receptor cells are located in this area, for each eye, a blind spot exists in the representation of the external world. Because each eye compensates for the blind spot of the opposite eye, we are not usually aware of its presence.

Eye muscles

The eye is rotated by two pairs of direct muscles and a pair of oblique muscles functioning as antagonist pairs, see Figure 2. The rotations are approximately symmetrical.

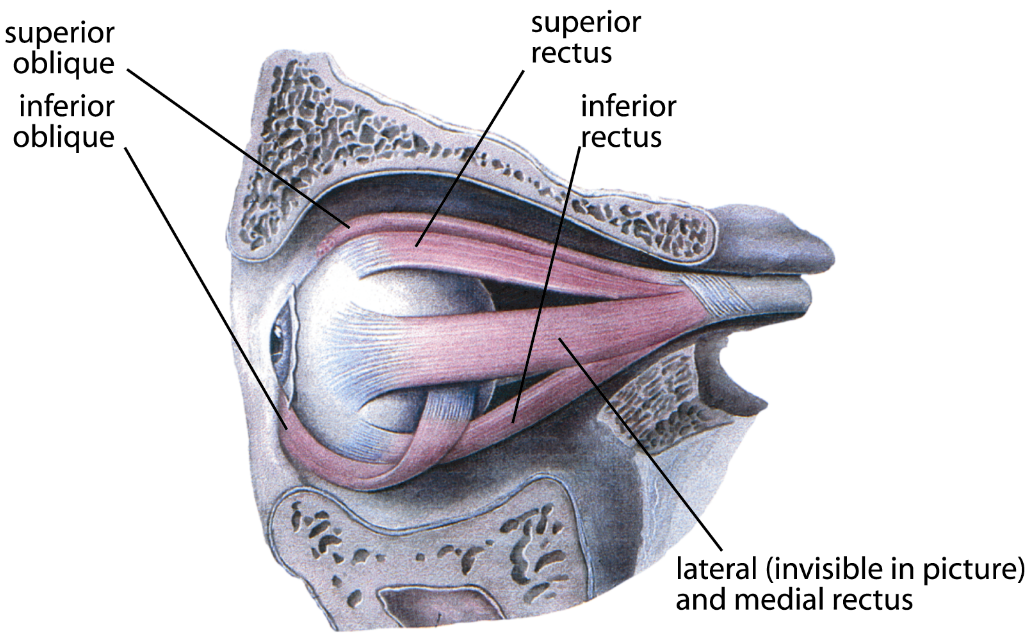


Figure 2 The muscles of the eye (Sobotta, 1989)

The muscles control the six degrees of freedom of the eye presented in Figure 3. The *lateral rectus* abducts the eye toward the nose and the *medial rectus* adducts the eye away from the nose. These muscles move the eye in the horizontal plane. The remaining four muscles, the *superior* and *inferior rectus* (elevating and depressing the eye) and the *superior* and *inferior oblique* (controlling intorsion and extorsion) control the vertical motion of the eye. The optic nerve is encased by the muscles of the eye as it is lead backwards from the eye.

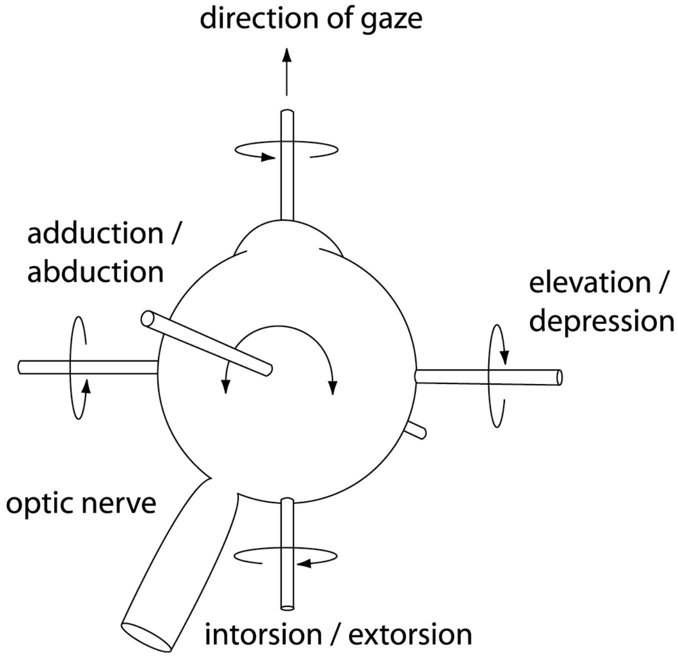


Figure 3 The six degrees of freedom of the eye (Goldberg et al, 1991)

Retina

The retina is the receptive sensory organ for visual stimuli and is of special interest to anyone trying to understand how the eye functions.

The retina houses seven layers of nerve cells involved in signal transduction, see Figure 4. Light enters the retina from the side of the ganglion cell layer, and must penetrate all other layers before reaching the photosensitive rods and cones. (Weedman Molavi, 1997)

The three most significant layers in understanding the function of the retina are the *photoreceptor layer* holding the photoreceptor cells, the *inner nuclear layer* holding the bipolar and amacrine cells, and the *ganglion cell layer*.

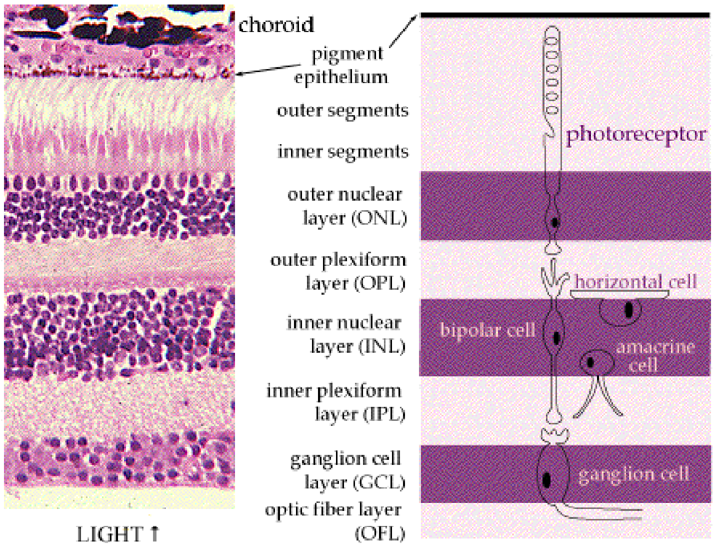


Figure 4 Cross section of the retina (Weedman Molavi, 1997)

In simple terms, the outer segments of the photoreceptor cells transduce the absorbed light to electric signals, and send these signals forward. The latter two layers start processing the received visual stimuli to features described as borders and contours. The bipolar and amacrine cells filter and compile signals from the photoreceptor cells, and pass them on to ganglion cells that form different receptive fields responding to different kinds of stimuli. The ganglion cells in turn send the signals in their axons on the optic fiber layer to the optic disk, to make up the optic nerve.

There are two groups of photoreceptor cells on the retina: rods and cones. Cones, in turn, come in three different varieties that can be categorized as red, green and blue, according to their peak sensitivities to different wavelengths of the visible spectrum of light, see Figure 5.

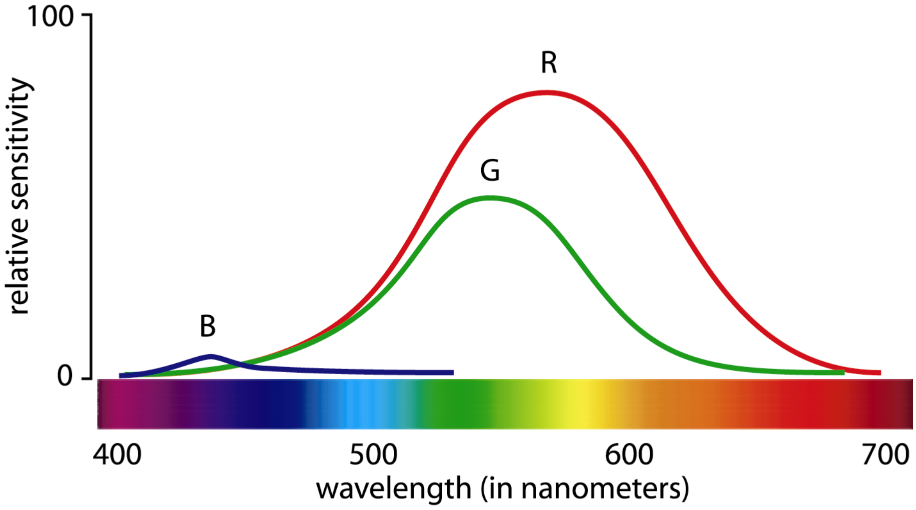


Figure 5 Relative cone sensitivity to visible spectrum
(adapted from Smith & Pokorny, 1975)

Photoreceptors are not evenly distributed throughout the retina. The cellular distribution is thickest in the fovea, degrading rapidly with increasing distance from the fovea. The fovea is dominated by cones, whereas the distribution of rods is thicker on the macula, see Figure 6.

Cones provide the focus on fine detail and distinguish color. They require relatively high levels of illumination to operate. Cones provide our straight ahead focused line of sight with the degree of perception and accuracy influenced by the level of available light. Rods, on the other hand, are much more sensitive to light, providing superior capability to detect movement in low levels of illumination. Rods provide our peripheral vision and are largely responsible for our visual capability to detect movement.

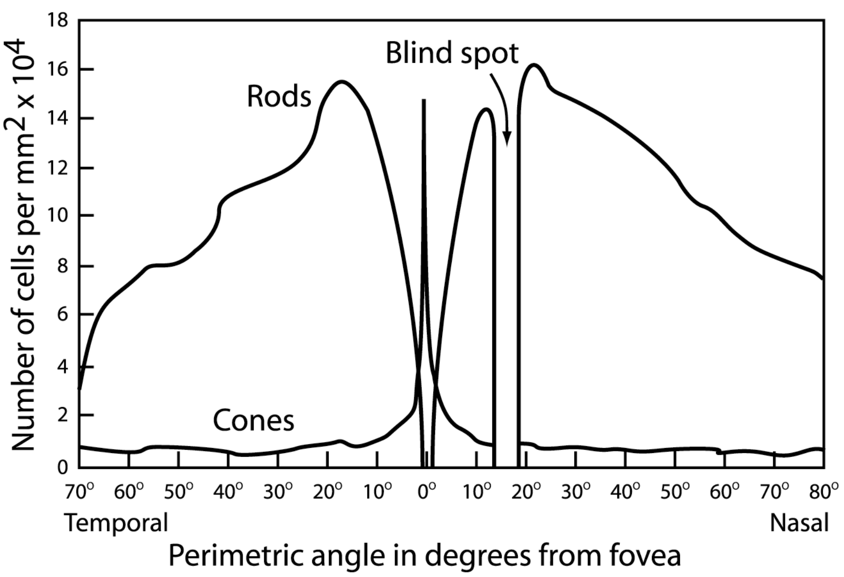


Figure 6 Distribution of rod and cone photoreceptors across the retina (adapted from Gonzalez & Woods, 1992)

The fovea defines the center of the retina, and is the region of highest visual acuity. Eye movements exist to direct the fovea toward the object currently studied. Surrounding the fovea is the macula, responsible for peripheral vision.

2.1.2 The visual system

To shortly summarize the latter phases of visual signal processing without delving further into the neuroscience of the visual system, the optic nerves, containing the axons of retinal ganglion cells, extend from the eyes within the basal frontal lobes to the *optic chiasm* (see Figure 7). At the chiasm, about half of the axons in each optic nerve cross to the optic tract on the opposite side. After the chiasm, the axons travel to the *lateral geniculate nucleus* of the *thalamus*, which passes the information on to the *primary visual cortex* located in the occipital lobes of the brain. Stimuli of the left visual field are processed in the right occipital visual cortex and vice versa.

Human capacity for information processing is limited. The brain processes sensory input by concentrating on specific components of the entire sensory realm, so that the object of central interest can be examined with greater attention to detail than peripheral stimuli. This is particularly true of vision; Human vision is a piecemeal process relying on the perceptual integration of small regions of visual information to construct a coherent representation of the whole (Duchowski, 2003).

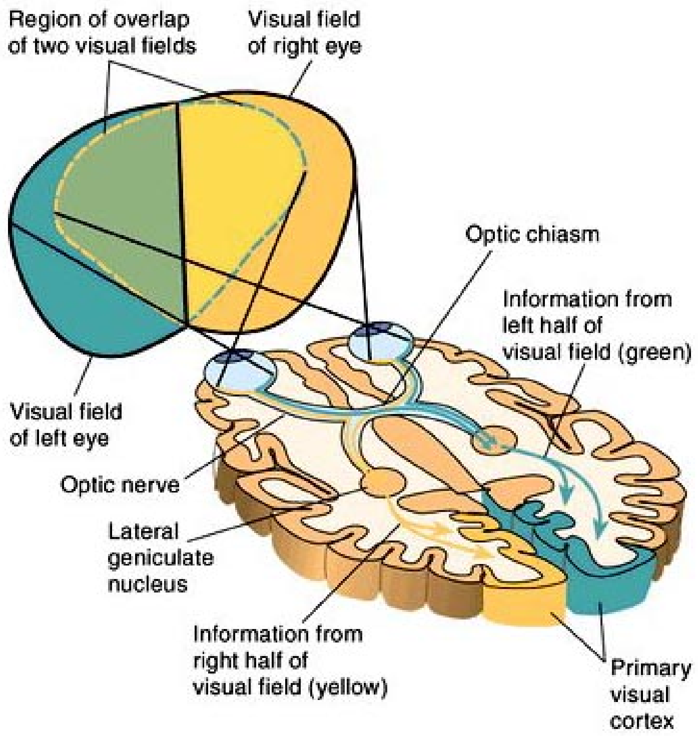


Figure 7 The primary visual pathways (Driesen, 2003)

The distribution of the photoreceptor cells on the retina defines the characteristics and the functional limitations of the human eye. The area of sharp vision, or highest visual acuity, is located on the fovea, and spans only one degree of visual angle. This corresponds approximately to the size of a thumbnail held at an arms distance. Visual acuity degrades rapidly when moving outwards from this area. An internal representation of the surrounding environment, or the user interface at hand, directs top-down, context dependent sequences of eye movements (Noton & Stark, 1971).

These processes occur at a level below conscious awareness, so their complexities do not yield to introspective report (Pelz & Canosa, 2001). The all-around sharp image we think we perceive of the surrounding world, is in fact an illusion created by the brain. The world our eyes actually "see" is altogether different. Our entire visual system exists to see borders and contours between objects with different contrast, luminosity, and color. We see the world as a pattern of lines, even things as complex as a face, and we judge colors and brightness by comparison, not by any absolute scale.

2.2 Eye movements

As explained in the previous chapter, to gather accurate information on the surrounding environment, the eye must be directed so that the image of the object under current scrutiny falls on the fovea. The human visual system has come up with a number of methods for bringing objects of interest to the area of sharp vision, and to stabilize the image on the fovea.

2.2.1 Types of eye movements

Eye movements can be broadly categorized into two main categories. *Stabilizing movements* that try to hold the eye, or rather the image on the retina, steady, and *saccadic movements* that move the eye around the visual field and bring objects of interest to the area of sharp vision. Stabilizing eye movements include fixations, smooth pursuit movements, and nystagmus. Saccadic eye movements include saccades and vergence movements.

Eye movements are typically measured as degrees of visual angle. One degree of visual angle spans approximately 1 cm on a distance of 57 cm from the viewer's eye.

Saccades

Saccades are fast and accurate ballistic eye movements used in repositioning the fovea to a new location in the visual environment. They can reach peak accelerations of 40000 deg/s^2 and a peak velocity of 400-600 deg/s, varying with the amplitude of the saccade. Saccadic eye movements can be executed voluntarily, reflexively as a response to a visual stimulus, and as a corrective movement associated with optokinetic or vestibular movement (Young & Sheena, 1976).

Saccades are ballistic in that their trajectory and destination cannot be altered once the saccade has begun. Since the saccade is ballistic, the target must be selected before the saccade is started, which in turn implies that peripheral vision, the area outside sharp vision, must be the means for selecting the target of each saccade.

The duration of a saccade depends roughly linearly on the distance from one visual object to another (Abrams et al., 1989) lasting 30-120 ms and covering a range from 1 to 40 degrees of visual angle (Sibert & Jacob, 2000). During a saccade, the observer is rendered effectively blind, as for all practical purposes and natural viewing conditions virtually no visual information is extracted during a saccade (an effect called saccadic suppression). It follows, that all visual information is gathered during fixations (Rayner & Pollatsek, 1989).

There is a 100-300 ms delay between the onset of a stimulus that might attract a saccade (e.g., an object appearing in the periphery) and the saccade itself. There is also a 200 ms refractory period after one saccade before it is possible to make another one. (Jacob, 1993)

Saccadic eye movements are inherently superior in speed, when compared to any other human modality for pointing objects (Sibert&Jacob, 2000).

Fixations

When directing gaze onto an object, the eyes move so that the image of the target object appears on the fovea of the retina. This is the high acuity area of vision, and it covers approximately one degree of visual angle.

During fixations, the image of an object of interest is held approximately stable on the retina. Eye movements do occur during fixations (see next chapter), although normal subjects are able to maintain the stability of their gaze within the 0.5° diameter of the "fixational fovea" (Stark, 1981).

At least three processes take place during a fixation (see Figure 8). First, visual information is encoded. Next, the peripheral field of the current gaze is sampled, to determine subsequent areas of information. Finally, the next saccade is planned and prepared. These processes overlap, and may occur in parallel (Viviani, 1990).

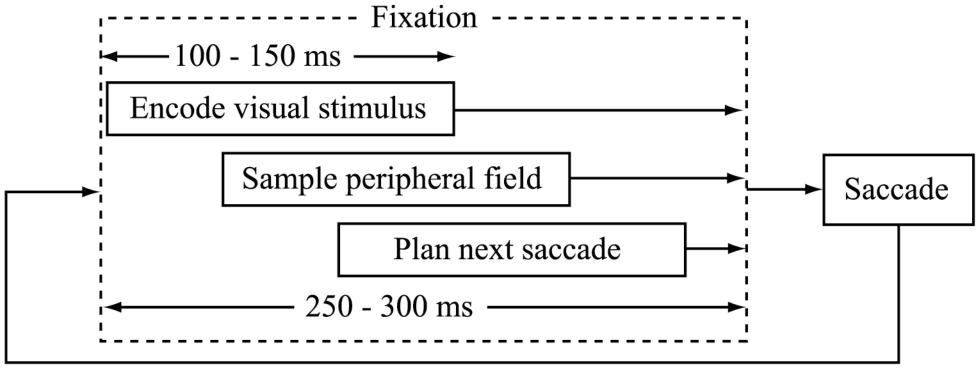


Figure 8 Events occurring within a typical fixation (Goldberg & Kotval, 1999)

Fixations generally last between 100-1000 ms, with the majority being between 200-500 ms, depending mainly on the quality of information being processed and current cognitive load (Goldberg & Kotval, 1999). Virtually all of the information gathered through the visual system is gathered during fixations.

Miniature eye movements

Miniature eye movements are movements occurring during a fixation, namely *tremor*, *drift* and *microsaccades*. Tremor is a high-frequency oscillatory component ranging from 30 to 100Hz, and drift is a slow random motion of the eye away from a fixation point. Velocities of these types of movements are only a few arc min/s, and they have been interpreted as noise in the oculomotor system.

Microsaccades are eye movements that are more or less spatially random, varying over 1 to 2 minutes of arc in amplitude. Their function is under debate; the traditional view being that they are error correcting movements, whereas others have suggested that they serve no practical purpose (Engbert & Kliegl, 2003).

Smooth pursuit

Smooth pursuit movements are involved in the visual tracking of a slowly moving target. Smooth pursuit eye movements follow a slowly moving target, keeping the image of the object on the retina more or less stable. Smooth pursuit movements are capable of tracking an object moving 5-30 deg/s. Above this velocity, saccadic movements compensate for the lag, “catching up” the target. Smooth pursuit movements cannot be induced voluntarily, that is, without a slowly moving target to follow.

Vergence movements

Vergence movements rotate the eyes inwards and outwards, so that they fixate roughly the same point in space regardless of the distance. Vergence movements are slow, 10 °/s, disconjugate movements, i.e. the eyes move in opposite directions relative to one another. The eyes rotate toward each other in order to focus on near targets, and in the opposite direction, or more parallel, for far targets.

Nystagmus

Physiological nystagmus can occur in response to motions of the head (vestibular nystagmus), or patterns in the visual field (optokinetic nystagmus). These are a pattern of smooth motion to track an object (as the head motion causes it to move across the visual field), followed by a rapid motion in the opposite direction to select another object.

Vestibular and optokinetic movements are compensatory movements, and work in conjunction to keep an object in view when the head moves. Vestibular movements are triggered by signals from the inner ear to oppose rotational movement of the head, while optokinetic movements are triggered by optical translations opposing uniform movements in the visual field, such as the view from a moving train's window. The amplitude of nystagmus is variable, generally between 1 to 10 deg. (Young & Sheena, 1976)

Torsional movements

Torsional movements are rotations of the eye about the line of gaze, and are generally limited to angles of less than 10 deg. The rolling motions may be stimulated by rotational optokinetic nystagmus or by vestibular responses. The torsional component of vestibular nystagmus or compensatory eye movement in response to head rotation, is similar to the horizontal and vertical vestibular nystagmus. That is, they respond to the head tilting sideways, compensating for the rotation of the visual field. (Young & Sheena, 1967)

2.2.2 Implications for eye movement analysis

Interpretation of eye movement data in interaction tasks can be based on the empirically validated *eye-mind assumption* (Just & Carpenter, 1984), stating that when a user is performing a cognitive task while looking at a visual display, he or she fixates an object as long as information of the visual target is being processed, and that there is no appreciable lag between what is being fixated and what is being processed. While this is a brave assumption, it is true in most cases, and allows for the process of observing a subject's visual attention by following his/her gaze path.

Based on this, three types of eye movements need to be modeled to gain insight into the overt localization of visual attention: fixations, saccades and smooth pursuit movements. Saccades are considered manifestations of the desire to voluntarily change the focus of attention. Fixations naturally correspond to the desire to maintain one's gaze on an object of interest. Similarly, smooth pursuit movements are used in the same manner for tracking an object in smooth motion. (Duchowski, 2003)

On the other hand, the eye-mind assumption has been deemed questionable. It is difficult to verify, because mental processing is private and cannot be measured directly. Viviani (1990) gives critique for the assumption and other difficulties encountered when eye movements are used to infer mental processing.

Worth noting is also, that the size of the foveal field of vision, about one degree of visual angle, determines the accuracy needed to fixate an object. It has actually been suggested, that it is not possible to discriminate the exact target of visual attention from a person's gaze direction with an accuracy better than one degree of visual angle (Ware & Mikaelian, 1987).

2.3 Tracking eye movements

Probably first remarked upon by Aristotle (Wade, 2002), eye movements have been described since antiquity. However, their detailed measurement is just over a century old practice.

Reviews of the early history of eye movement measurements can be found for instance from Jung (1977) or Heller (1988). Among the first was a study made by Dodge and Clein back in 1901, using a slowly falling photographic plate to measure the position of the first Purkinje image (see chapter 2.3.1). The classification of human eye movements into five subtypes documented by Dodge (1903) is, in effect, still in use today.

Recording techniques have been evolving ever since, and this development still continues. For a more recent review see, for example, Young and Sheena (1975) or Carpenter (1991). Lately, the developments have mainly been concerned with better technical implementations of existing principles, afforded by technical development and reductions in the cost of computational power, rather than creating novel methodology (Collewijn, 1999).

It should be noted, that all of the eye tracking devices provide massive sets of data points, or a stream of samples of the currently measured parameters. For purposes beyond the basic oculomotor research, this data has to be filtered, reduced and refined. However, even the concepts of a fixation or a saccade still lack a standard definition! The process of fixation identification is an essential part of eye movement data analysis, and can have a dramatic impact on higher-level analyses (Salvucci & Goldberg, 2000).

For usability studies, the raw data should be reduced into a more event-oriented form that can be incorporated to the displays being viewed, and to events that occur in the user-system task environment (Benel et al., 1991).

A note should be made on terminology. The term *eye movement* refers to the physical rotations of the eyes in their sockets. The term *gaze* should be interpreted as the point-of-regard on a screen or user interface.

2.3.1 Existing methods

There are a number of principles used in measuring eye movements, including measurements of electric and photoelectric signals, tracking a number of visual features in the image of the eye, measuring relative reflection of infra-red (IR) light, and using either mechanical or optical levers or a magnetic field. The early methods even include using a mechanical lever attached to the corneal bulge of a cocaine-anaesthetized eyeball with a suction cup!

The methods described here in more detail are the ones in use currently and available as commercial systems. A reasonably comprehensive list of eye tracking device manufacturers can be found from the Eye Movement Equipment Database, available online at http://ibs.derby.ac.uk/emed/.

The methods can be divided into two groups: those measuring angular eye position relative to the head, including electro-oculography, IR-reflection oculography, and head mounted video systems, and those measuring eye position relative to the surroundings, including table-top video systems and the magnetic scleral search coil method.

As a general rule, the methods measuring eye position relative to the head are more accurate, and intended for the study of oculomotor dynamics, whereas the methods measuring eye position relative to the surrounding environment are used for gaze point measurements on a user interface.

The choice of an eye tracking method in any study should be based on the particular demands of the application. None of the current methods is the universal best for all applications. The deciding factors in choosing equipment can be reduced to

- temporal and spatial accuracy
- suitability for operational conditions

- invasiveness
- cost

The temporal and spatial accuracy should be considered in relation to the objectives of the study. Higher temporal accuracy means massive data sets, whereas high spatial accuracy tends to require rigorous stabilization of the subject's head, or the use of more invasive methods. Operational conditions restrict the choice of a system in freedom of movement for the subject, ambient lighting requirements, and the requirements imposed by special environments, such as a functional magnetic resonance imaging (fMRI) laboratory used in brain imaging.

Electro-oculography

The Electro-Oculography (EOG) method was introduced by Fenn and Hursh (1934). The method utilizes the difference in electric potential between the cornea and the retina. The corneoretinal potential is probably caused by the electrical charge created in the photoreceptor cells on the retina, and is in the order of 15-200 μV, the cornea being positive relative to the retina. The field created by the dipole can be picked up from skin electrodes placed around the eye, and translated to a signal describing the rotation angles of the eyes. (Collewijn, 1999)

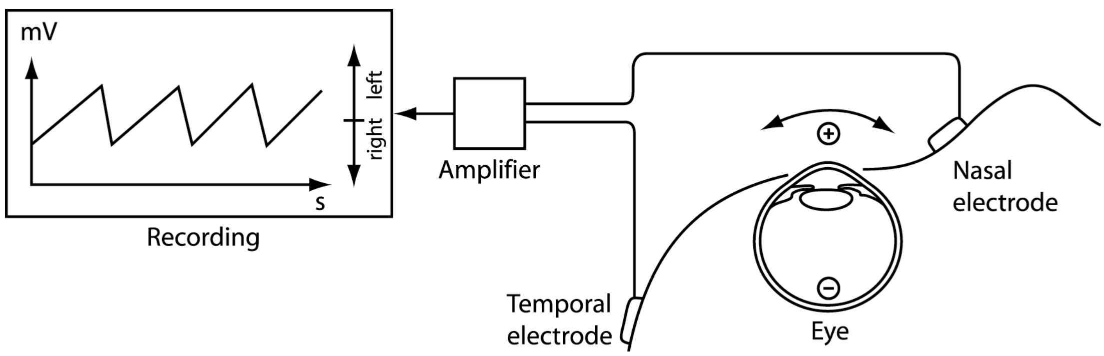


Figure 9 Typical EOG electrode setup (adapted from Grüsser, 1983)

The method, however, has some disadvantages. As the electric potential is induced by the photoreceptor cells, the potential difference is likely to change with the change of ambient lighting conditions and subject adaptation. Also, the electric signals caused by the movement of facial muscles around the eye interfere with the signal. Actually, even the concept of a single, symmetrical dipole moving in homogenous conducting medium is a strong oversimplification, which has actually been shown to be wrong (Berg and Scherg, 1991).

Despite these drawbacks, with a controlled test setup and a well prepared subject, valid data can be collected. EOG provides a high sampling rate with a resolution of about one degree at best. The method works better for horizontal eye movements due to the anatomical structure of the eye socket, and the associated movements of the upper eyelid. Also, the method is more useful for measuring relative movement of the eyes, rather than absolute position.

As such, EOG-based eye tracking may stand out in certain specialized applications, where other techniques are harder to use. EOG is suitable for clinical use, diagnosing neurological problems revealed by eye movement patterns, and the study of certain oculomotor characteristics, rather than use in applied research or usability studies. EOG measurements can be performed also during sleep.

EOG measurements can be made relatively unobtrusively, requiring only the attachment of a number of skin electrodes to the subject. EOG recordings have a range on the order of ± 70º. That said, the relationship between EOG output and angle of gaze is linear only in a limited range of ± 30º vertical and ±15º horizontal. (Joyce et al., 2002)

EOG amplifiers are manufactured by at least

- Cambridge Research Ltd (http://www.crsltd.com)
- Colbourn Instruments (http://www.colbourn.com)
- Metrovision (http://www.metrovision.fr)

Infra-red oculography

These loosely grouped methods utilize measuring the diffuse reflection of infra-red light from the frontal surface of the eyeball. A number of IR light sources are used for illumination, and photo detectors aimed at receiving the reflected light for picking up the signal. The systems track the limbus (the boundary between sclera and iris), or the pupil-iris-boundary, to measure relative eye rotation.

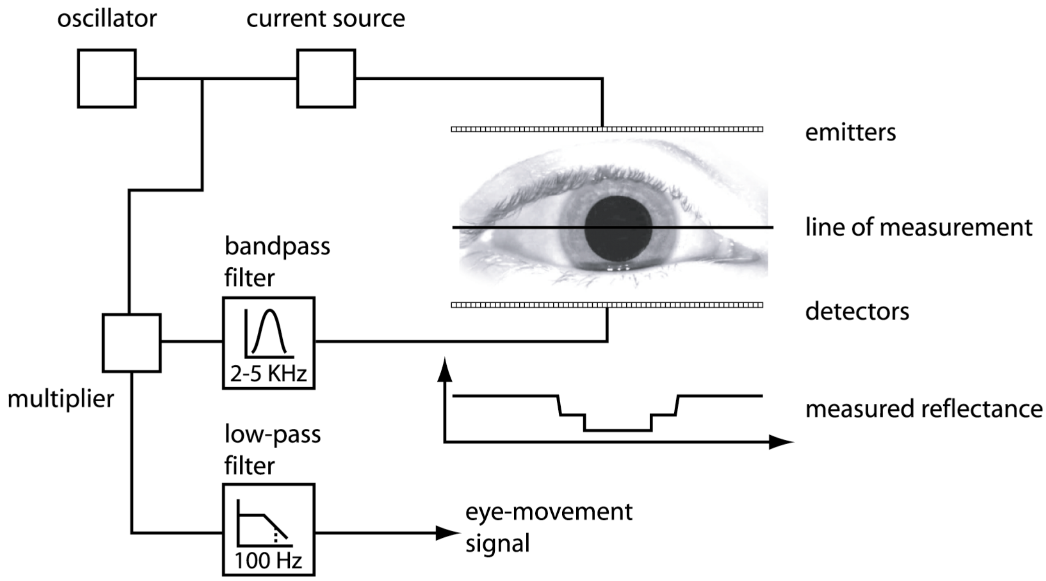


Figure 10 IROG measurement (adapted from Skalar, 2003)

Different systems use different setups, relying either on focal or diffuse illumination and a different number of detectors. The systems provide a high sample rate and resolution, but tend to be difficult to calibrate. Also, the calibration relies heavily on the stability of the light sources and the photodetectors during trials. The slightest movement of these relative to the eye can corrupt the calibration. Drift and accuracy are also problematic issues over longer measurement periods.

These methods have some fundamental limitations. The detection works well for measuring horizontal eye movements over a fairly large range between ±15° and ±40° depending on system design. For vertical movements, the signal is much worse if not non-existent, due to the fact that the eyelids occlude the iris-sclera boundary.

IROG systems include

- ❖ Applied Science Laboratories Model 31

(http://www.a-s-l.com/model310.htm)
- Cambridge Research MR Tracker (http://www.crsltd.com/catalog/mr-eyetracker/)
- Skalar IRIS (http://www.skalar.nl)

Purkinje image trackers

The front and back surfaces of the cornea and the crystalline lens constitute four surfaces with specular reflection, due to the difference in refraction index between the layers. These surfaces form four "Purkinje images" of an external light source. The first of the Purkinje images is also called the corneal reflection. The reflections move differently in relation to the eye, due to the fact that the corneal radius of curvature is smaller than the distance from the corneal surface to the ocular center of rotation.

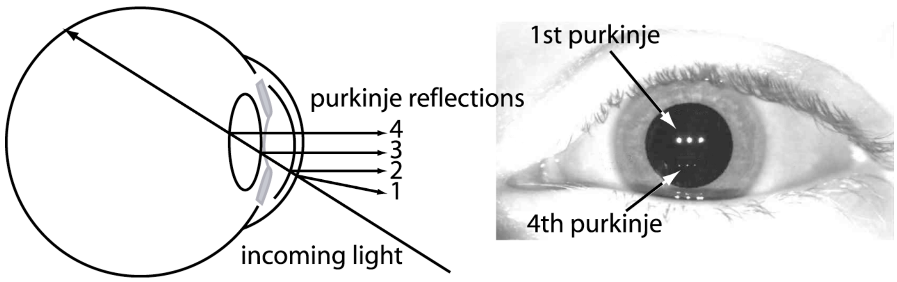


Figure 11 Purkinje images (photo adapted from Pongs, 1998)

Trackers using the first Purkinje image track the displacement of the relatively bright, virtual image of the light source. The relationship between eye rotation and corneal reflection displacement is reasonably linear, and can be calibrated to reflect the direction of gaze.

Trackers using the first and fourth Purkinje image utilize the fact, that the first and fourth Purkinje image move similarly during eye translations, but their relative separation changes proportionally during eye rotations. This method suffers from the difficult acquisition of the relatively dim fourth Purkinje image, and the fact that with larger rotation angles, the iris occludes the fourth image.

These methods are able to deliver high bandwidth and resolution. They are also the only method capable of measuring accommodation (the transformation of the lens within the eye), that can be used to measure the focal depth of the eye. However, the method is very sensitive to translations between the eye, the detector, and the light source, and usually requires rigorous head stabilization.

Probably the only commercial dual purkinje image tracker available is the Fourward DPI Eyetracker (http://www.fourward.com)

Scleral search coil method

The scleral search coil method was first published by Robinson (1963). The method is described by Collewijn (1999) as follows. An alternating current (AC) magnetic field is created with suitable field coils. When a sensor coil is placed within this field, an AC voltage signal is induced in the coil. This signal has the same frequency as the surrounding field, and a magnitude proportional to the sine of the rotational angle between the field lines and the coil, and the number of turns of the sensor coil. According to Faraday's law, the magnitude of the induced signal is also proportional to the field frequency, because the induction depends on the velocity of change in the magnetic flux.

A search coil embedded in a suction lens or silicone annuli is attached to the corneal bulge of the eyeball. The signal induced to the search coil is then measured with an amplifier.

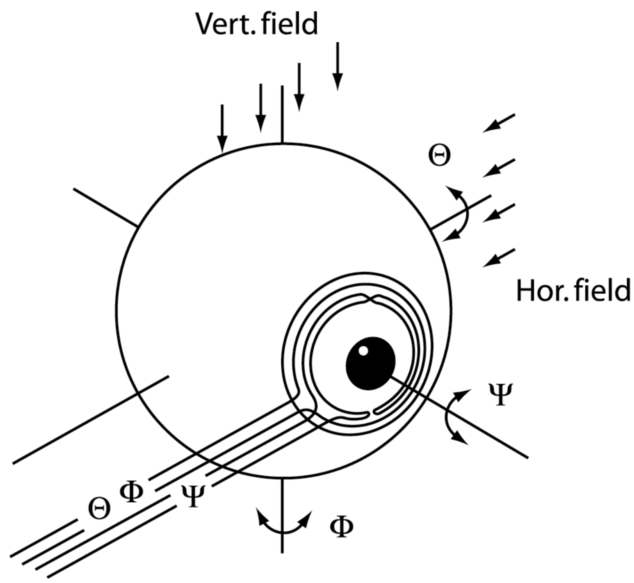


Figure 12 Scleral search coil method principle of operation (adapted from Collewijn, 1999)

By creating two magnetic fields with different phase or frequency, and placing them orthogonal in space, the method can be used to measure both horizontal and vertical (Φ and Θ, see Figure 12) orientation of the coil. The insertion of another coil following a “figure of 8” pattern in its turn enables the measurement of torsional eye movements (Ψ), effectively creating a 3D magnetic measurement system of eye movements.

The scleral search coil method is accepted as the “golden standard” of oculomotor measurement. The signals of the coil are a direct representation of the orientation of the coil, and thus – provided that the lens attachment is rigid – of the eye. The technique can provide 3D positions of both eyes in real-time without restrictions. Due to the nature of measurement, the accuracy, precision, range, bandwidth, and linearity can easily meet realistic requirements for oculomotor recordings.

However, the method is quite invasive because of the need to wear big contact lenses with suction, and with wires coming out of the lenses. This causes considerable discomfort to the subject, effectively limiting the wear time to about 30 minutes. Because of the invasiveness, system cost, and demanding use, the method is not in a wide spread use, and its primary use is in precise oculomotor recordings.

Systems based on the scleral search coil method are marketed at least by Primelec (http://www.primelec.ch), and Skalar (http://www.skalar.nl).

Video-oculography

Video-oculography (VOG) encompasses several methods relying on tracking the visible features of the eye, or reflections on the eye surface. The primary features tracked are the position and apparent shape of the pupil, the corneal reflection (first Purkinje image) and, for some systems tracking torsional movement, the iris. Figure 13 shows the center of the pupil marked with the red cross, and the corneal reflection marked with the blue cross. The methods typically use infra-red light for illuminating the eye. The methods can be divided into two groups, the table-top systems and the head-mounted systems.

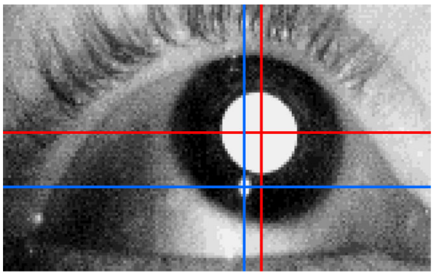


Figure 13 Typical features tracked with VOG

VOG tracking methods can be based on tracking either the center of the pupil, or tracking both the center of the pupil and the corneal reflection. The corneal bulge has both a different amount of curvature and center of curvature than the eyeball. The pupil moves in relation to the eye center of rotation, while the corneal reflection moves in relation to the center of curvature of the corneal bulge. This difference can be used to distinguish between eye rotations in its socket and eye translations in relation to the camera.

VOG systems typically provide a relatively low sampling frequency of 60 Hz, although at least the Eyelink (SR-Research, 250 Hz) and Chronos (Skalar, 400 Hz) systems provide much higher frequencies. VOG methods typically have an accuracy of 0.5° - 1° of visual angle with good resolution.

These methods are currently the most promising approaches, and their speed and accuracy are rapidly improving with the introduction of better camera technology and brute computational power. In their study, van der Geest and Frens (2001) compared a video tracking system (Eyelink, SR Research, similar to the one used in this work) with the more invasive but accurate scleral search coil technique. Their study showed that the results between the systems were highly correlated, both systems practically providing the same data.

Table-top systems

The table-top methods include systems with the camera and the light-source sitting on top of a table or a monitor in front of the user. The user has to sit relatively still, the systems allowing for a variable amount of head movement. These are the most non-invasive methods available for eye tracking.

The systems typically acquire the image of the eye with a camera equipped with a tele lens and a servo-controlled mirror. These systems track both the pupil and the corneal reflection. Most of the systems calculate the gaze point on a screen defined through a calibration sequence.

A system manufactured by Tobii Technology (http://www.tobii.se) uses a camera and several light sources attached to a monitor to provide calibration free eye tracking through the use of an advanced mathematical method for extracting gaze direction.

Head-mounted systems

The head-mounted systems use a helmet, a headband, or a form of goggles to attach the cameras and the IR light source to the head, near the eye. Wearing the helmet adds some discomfort to the user; usually, however, not a significant amount.

These systems measure eye movements relative to the head, although some of the systems have a method for tracking or compensating head movement and measuring the gaze relative to the surrounding environment or a stimulus screen. The eye tracking device used in this thesis is the Eyelink I (SR Research), which is a head-mounted VOG system.

The plethora of manufacturers providing VOG based eye tracking equipment include

- Senso Motoric Instruments GmbH (http://www.smi.de)
- SR-Research (http://www.eyelinkinfo.com)
- Skalar Medical (http://www.skalar.nl)
- Applied Science Laboratories (http://www.a-s-l.com)
- Tobii (http://www.tobii.se)

2.4 Eye movements and usability

The ISO standard 9241-11 (ISO, 1998) defines usability as consisting of three distinct aspects:

- *Effectiveness*, which is the accuracy and completeness with which users achieve certain goals.
- *Efficiency,* which is the relation between the accuracy and completeness with which users achieve certain goals, and the resources expended in achieving them.
- *Satisfaction*, which is the users' comfort with and positive attitudes towards the system.

Indicators of effectiveness include quality of solution and error rates. Measuring efficiency can utilize such metrics as task completion time and learning time. User satisfaction can be measured by attitude rating scales and questionnaires.

Frøkjær et al. (2000) point out, that all of the three aspects of usability should be studied in usability testing. The selection of measures depends on application domain and context of use. They emphasize that especially critical is the discovery of solid measures of effectiveness.

Eye movement measures could provide more finely grained information for measuring the quality of solution and errors made during the process. For instance, an amateur and an expert user are in complex cases likely to deliver different solutions for a given problem. Eye tracking can provide a detailed description on the information retrieval process of the two users, and give information on the differences between their performance.

Crowe & Narayanan (2000) list three questions that should be studied to form a comprehensive view of the usability of an interface. The first question is *whether the interface produces the desired outcome*. That is, does the interface allow the user to accomplish a given task? The second question asks *whether the interface provides better means of producing the desired outcome than optional interfaces*. The third question is concerned with *what makes the interface better or worse* than optional solutions.

Typical usability studies collect

- *process data*, including task completion times, verbal protocols, error rates, experimenter observations, and user interaction logs
- *outcome data*, including pre- and post-tests to measure the change in users' skills or knowledge, and percentage of participants succeeding
- *survey data* from interviews and user satisfaction questionnaires

These techniques can capture most observable, behavioral aspects of work and task performance. Studies using these techniques can provide answers to the first two questions. All these, however, miss one crucial aspect of interaction: When there are multiple information carrying entities simultaneously presented on the screen, how do users allocate and shift their attention visually among these? (Crowe & Narayanan, 2000)

A comprehensive picture of the interaction process includes not only what navigation and control actions users undertake, but also how they distribute their visual attention among multiple visual entities present on various screens of the interface of an interactive application. Eye movement, or gaze data, can provide a tool to collect the data necessary to answer the third question by providing

- information about which areas of the interface the user is concentrating on, or
- implications on which cognitive processes the user is going through

within a given time frame. A significant advantage in using eye movements as a source of data, is that they are automatic, giving an objective measure of where the subject's visual attention is directed.

For instance, think-aloud protocols are one of the tools regularly used in usability studies. In these, the user tries to describe what he or she is thinking throughout performing the given task. Micro-level behaviors, such as the focus of attention during a task, distractions, or the visibility of an icon, usually have little awareness to an individual, and are thus not reported in a think-aloud protocol. Reading, mental computations, problem solving, and thinking about the content of an application, are also difficult to quantify using this protocol. (Goldberg & Wichansky, 2003) Also, the think-aloud protocol itself adds an extra task for the user, and could have an effect on the results.

A number of situations in assessing product or system usability would benefit from the knowledge of where a person is looking at a given time. Measures such as scan paths and patterns, pupil size, and gaze direction are useful for inferring usability, helping designers select the best of several displays, and characterizing vigilance, mental workload, and attention (Benel et al. 1991). Goldberg (2000) confirms that eye tracking can provide an effective indication of certain aspects of usability. Visual clarity and recognizability of target objects should be quite accessible through eye tracking approaches.

Several studies have reported results using eye tracking measures to infer usability issues of user interfaces. For instance, Crosby & Peterson (1991) found several scanning strategies of ordered lists, and showed that these correlated with the cognitive style of the subjects. The Stanford-Poynter Project (2000) extended the Poynter Institute's study of reading newspapers to reading news online on a web page. Their results concluded, that in general, users were first drawn to headlines, article summaries, and captions. Also, the users often did not look at the images at all until the second or third visit to a page.

Typically, however, in usability, eye tracking data is used for recommendations on how a user interface should be changed, rather than assessing global product usability (Goldberg & Wichansky, 2003).

2.4.1 Metrics

The choice of metrics depends on the underlying objectives of the study. There are basically two ways of doing research based on eye movement data:

1. top-down : use the data to test hypotheses based on a cognitive theory
2. bottom-up: collect large amounts of data and try to find patterns and behaviors to form a theory.

Benel et al. (1991) list the following typical parameters of interest, extracted from gaze data, for usability assessment:

- The proportion of time spent fixating each region of interest
- The mean and standard deviation of ocular dwell times on each region of the screen
- The number of fixations while a template is being installed
- The latency to look at each region of the screen after it became available
- For each region viewed, the number of times that each other region was viewed immediately prior (i.e. scan pattern)
- The mean and standard deviation of pupil size changes
- Blink frequency
- Mean and standard deviation of blink durations

For instance, saccade amplitude measures the distance between successive fixation points. Large mean values for saccade amplitude could be interpreted as the user being able to position his/her gaze accurately on the display without random searching. This in turn could be interpreted as either the user having a good mental model and familiarity of the interface, or as the interface enabling better planning of saccades through the effective use of visual cues in the area of peripheral vision.

Kotval & Goldberg (1998) found, that generally the most sensitive measures of rated usability differences were:

- scanpath length and duration,
- number of fixations,
- number of saccades,

whereas parameters such as fixation duration, saccade amplitude, or fixation/saccade ratio, were not as successful in predicting subjective usability. They also found that blinks per time unit and pupil size function as indicators of cognitive load. Cowen's study (2001) confirmed the relationship between the sensitivity of ocular measures and page layout variables in web pages.

Karn et al. (1999) list the following "degrees of information" for eye tracking data:

1^{st} degree information	raw data points
2^{nd}	fixations and saccades
3^{rd}	gaze paths, total fixation time per area of interest (AOI), transition probabilities between AOIs
4^{th}	shape, complexity and variability of the gazepath

A higher degree corresponds to a higher level of signal filtering and analysis.

According to Salvucci & Goldberg (2000), fixation identification is, from a data-reductionist point-of-view, a convenient method of minimizing the complexity of eye tracking data, while retaining its most essential characteristics for the purposes of understanding cognitive and visual processing behaviour. Also, Sibert and Jacob (2000) argue, that of all the different eye movements, only fixations and saccades are of value for computer interfaces.

This is intuitively true on a stationary display with the subject's head held stable. However, mobile, handheld devices and use "on the move" bring a whole new dimension to the question. Pelz et al. (2000) reported, that during measurements of an unrestricted subject free to make head and body movements, the eyes were frequently moving with respect to the head, even during fixations. Vestibular nystagmus and smooth pursuit eye movements that stabilized the retinal image, often occurred during periods defined as fixations in video analysis.

Lee (1999) showed in his study, that at least in reading, head movements constitute an integral part of gaze saccades for gaze shifts as small as 3 degrees of visual angle.

Despite this, most studies measuring eye movements have required immobilizing the head using a headrest or a bite bar.

2.4.2 Eye tracking in usability evaluations

Goldberg & Wichansky (2003) list requirements in knowledge and methodology for using eye tracking in usability evaluations:

1. Eye tracking must not slow down usability evaluations. Test sessions are usually under strict time constraints, which calls for faster setup-times and more robust calibration.
2. Eye tracking must work in a relatively unconstrained participant testing environment. The use of eye tracking shouldn't restrict the user from moving and behaving normally during a measurement session. Current techniques compensating for head movement need improvement.
3. Better tools for analysis. Automatic parsing of raw eye tracking data, or fixations and saccades, to more meaningful parameters, quantitative indications, AOI-analysis, and transitions between these regions.
4. Standards for eye tracking in usability should be developed and published. Eye tracking derived measures cannot currently be correlated to standard usability metrics. There is a strong need for research relating these two sets of measures. Minimum standards, such as frequency of calibration, data sampling rates, and equipment specifications, would make it easier to cross-interpret data from multiple studies.
5. Better knowledge on the contribution of task factors to eye tracking-derived metrics. For instance, how the density of a display (usually defined as the area occupied by target objects on the screen divided by the area occupied by white space) or visibility of icons influences eye tracking results, and perceived usability.

Karn et al. (1999) reported the results of a workshop held at the CHI'99 (ACM SIGCHI Conference on Human Factors in Computing Systems) with participants having their background in computer science, usability evaluation, human factors, user interface design, psychology, and information sciences. As a result of the workshop, all participants agreed, that it is worth the time and effort to collect eye tracker data when the domain is well understood. The workshop generated the following list of reasons for using eye tracking during usability tests:

- Integration with other types of data
- Help discriminate “dead time”, or the time that the user is “not productive”
- Measure how long a user looks at an area of interest
- Capture a sequential scan path
- Evaluate a specific interface
- Extract general design principles
- Demonstrate scanning efficiency
- Understand expert performance for training
- Help sell usability testing
- Provide a quantitative comparison of UI designs
- Provide domain specific benefits (web pages, cockpits, text design)
- Help explain individual differences in performance

The workshop also produced a wish list for an *ideal* eye tracking system for usability testing:

- Easy to use
    - The system should be quick to set up and calibrate
    - Data collection and analysis is simple and rapid
    - The system provides a real-time view of the scan path
- Unobtrusive
    - No contact with the subject
    - No restrictions for the subject’s movement
    - Non-contact head tracking
- Track multiple users simultaneously
- Accurate, with a resolution better than 0.5 degree
- Fast, with a sampling rate of at least 60 Hz
- Robust
    - Able to track any participant
- Small and low-cost
- Compatible with commercial off-the-shelf software
- Fully integrated and synchronized data streams

2.5 Positional tracking

There are various positional tracking methods, including mechanical, acoustic, magnetic, inertial, and optical trackers. Trackers can be classified using the following set of metrics (Bhatnagar, 1993):

Accuracy	The amount of error in the position and orientation reported by the tracker
Resolution	The smallest change detected by the tracker in position and orientation
Update rate	The rate with which the measurements are reported by the tracker to the host computer
Lag or latency	The delay between a change in position and/or orientation and the report of the change to the host computer
Working space	The volume within which the tracker can measure position and orientation with its specified accuracy and resolution

Mechanical position trackers have a low lag, and are insensitive to their environment. However, they have a small working volume, and they create motion restrictions and discomfort in users because of their mechanical linkages.

Acoustical trackers are small and light weight, so they can be comfortably worn by the users. They do not suffer from distortions from electric or magnetic fields, and set no special requirements for the environment. However, they suffer from low update rates due to the relatively slow speed of sound, they need an unobstructed path from the transmitter to the receiver, and are sensitive to external noise and echoes.

Optical position trackers are able to work over a large area, but need to maintain a line-of-sight from the set of reference points to the camera. They also require an elaborately designed environment and a relatively large separation between tracked targets for accurate measurements.

Inertial trackers use accelerometers and gyroscopes to compute relative change in position and orientation, based on velocity and acceleration measurements. As the measurements are relative and have no point of reference, they tend to accumulate drift and error over time, and require recalibration during measurements.

Magnetic trackers are generally very flexible. Their sensors are small; hence they can be comfortably worn by the subject. They give good resolution and accuracy with sufficient update rates. The main disadvantage is their sensitivity to ferrous objects and magnetic fields produced by electronic equipment, which create distortions in the field of measurement. They are immune to line-of-sight issues, so they place no restrictions to the design of the environment, other than their sensitivity to distortions in the transmitted magnetic field.

3 Definition of a gaze tracking system for usability studies of mobile devices

The practical part of this thesis was to

- define a system for studying the usability of mobile devices using gaze tracking
- implement a prototype system for recording gaze position data while using a mobile device in a free-use-environment

This chapter defines the features and the requirements of a system for measuring gaze position data, intended for studying usability aspects on a mobile screen or user interface.

3.1 Questionnaire

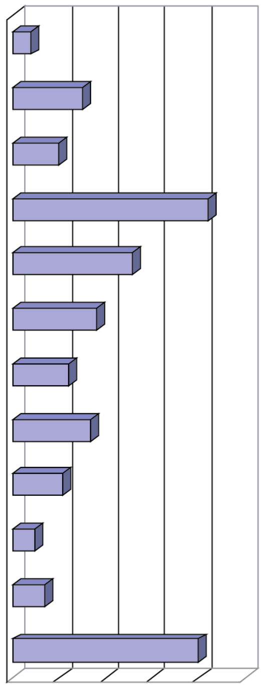
At the beginning of the project, a questionnaire was delivered to twelve Finnish researchers doing eye movement-related work, part of them in the field of mobile, handheld devices. The results summarizing data from the eight participants that answered the questionnaire are presented in this chapter.

The questionnaire was divided into three parts. The first part was designed to enable the participants to deliver personal ideas of the requirements and future applications for a system tracking gaze on a mobile device. In the second part, the participants had to prioritize six of the most important features of such a system from a list of twelve. The prioritizing was done by rating the selected six features into an order of importance. The third part was concerned with what kind of questions could be answered with qualitative versus quantitative analysis of collected gaze point data. The questionnaire was delivered in Finnish, the translated version can be found in appendix 5.

The results from the second part are presented in Table 1. The results from the first and third parts of the questionnaire were used as a basis of the early steps of system definition. These included some expected uses for the system, including research on dynamic visual perception (using a device while walking), and research on the perception of three-dimensional objects, for instance a box of cereals with logos and graphics, text, and nutritional information.

Table 1 Questionnaire results

Property	Rating
The strain exerted by the system on a subject should be minimized	12
System should set no restrictions on subject during measurement	6
The system should be easy to calibrate	9
The calibration should be accurate	1
The system should be reliable and robust	3
The system should deliver data for all measurable parameters	4
The data should be easy to use and analyze	7
Different experiments should be easy to set up	5
Experiment set up should be flexible	8
The user interface of the system should be easy to use	11
Temporal resolution	10
Spatial resolution	2



3.2 System definition

Collewijn (1999) lists the ideal properties of an eye movement recording system. An ideal system should deliver measurements for the three degrees of freedom (horizontal, vertical and torsional orientation) for both eyes, and define the geometry as the position and rotation of the head and both eyes, plus the position and orientation of the target objects and the surrounding environment. The system should deliver linear data over a wide range of eye movements, and should give good signal-to-noise ratio with sampling rates preferably as high as 500 Hz, and in real time. A somewhat similar definition can be found in Hallett (1986).

These definitions, however, apply to systems for studying the properties of the oculomotor system. As the current system is designed for studying usability, some of these properties can be compromised. For instance, a user generally need not position his/her eye more accurately than the width of the fovea (about one degree) to see an object sharply. Finer accuracy from an eye tracker is needed for studying the operation of the eye muscles but adds little for the current purpose. The eye's normal jittering further limits the practical accuracy of eye tracking (Jacob, 1993).

The properties for an ideal system for usability testing has been described at least in Karn et al. (1999) and Goldberg & Wichansky (2003), see chapter 2.4.2 (in this thesis). In addition to these, Table 2 lists the properties prioritized in the questionnaire performed (see chapter 3.1).

Table 2 Properties for an ideal eye tracking system for usability studies

Property	Questionnaire rating	Karn et al.	Goldberg & Wichansky
Quick and easy set-up and accurate calibration	1, 9	✓	✓
Good spatial resolution (Karn et al.: < 0.5º)	2	✓	
Robustness and reliability	3	✓	
Integrated and synchronized data streams / data coverage	4	✓	
Ease of setting up experiments	5, 8		
System sets no restrictions on subject during measurement	6	✓	✓
Simple, rapid data collection & analysis	7	✓	✓
Good sampling rate (Karn et al.: min 60 Hz)	10	✓	
Real-time view of scan path		✓	
No contact with subject		✓	
Non-contact head tracking		✓	

All in all, the researchers seem to set their hopes on a system for reliable and accurate data gathering with simple test set-ups and easy post-measurement analysis, and a system, that permits as much of natural movement for the user as possible with minimum invasiveness.

The *quick and easy set-up* and *accurate calibration* resulting in *good spatial accuracy* was seen as the most important property. *Good spatial resolution* is needed to accurately match gaze position to distinctive portions of the user interface or stimuli. *Robustness and reliability* in practice demand no dropped frames during measurement, and no repeated tests due to device malfunctions. The system should preferably set *no restrictions for the user* during measurements and *not make contact* with the user. The system should provide the experimenter with *extensive data*, although a high sampling rate for use in usability research was not seen as a priority. For subjective evaluation of the ongoing test, the system should provide the experimenter with a *real-time view* of the user's scan path.

4 Tracker prototype

In this chapter, the prototype system built for measuring gaze position on a handheld user interface is described. The chapter outlines the system definition, explains the principle of operation, takes the reader through the process of system development and software structure, and finally discusses the related performance issues.

As a historical note, probably the first mobile gaze tracker was described by Shackel (1960). In their system, eye movements were tracked with OG and the resulting gaze cursor superimposed on the image of a TV camera attached to the helmet of the user, pointing approximately at the same scene as the user (see image). The system was awkward and heavy, and limited the time of use to a few minutes. However, even with this equipment and hand made data analysis, they were able to demonstrate the usefulness of such a system.

The current prototype was built as a proof of concept, and will work as a basis of building new research equipment. A typical measurement session is seen in Figure 14.

Figure 14 A typical measurement session

4.1 System definition

The objective of the system is to measure the binocular gaze points of a user on a mobile screen, or user interface. Table 3 lists the prototype specifications in relation to the requirements presented in chapter 3.2.

Table 3 Prototype specification

Requirement	Prototype specification
Quick and easy setup and accurate calibration	The set-up requires attaching the headgear to the user and going through the following three-phase calibration (described in detail below): - standard Eyelink calibration - locating the sighting centers[1] of the eyes - defining reference vectors for translating the Eyelink coordinates to real world vectors.
Spatial accuracy	As the human fovea extends an area of 1° of visual angle, the resulting accuracy for a fixation is within that 1° of the target stimulus. Therefore, the accuracy of reporting a gaze point within 1° should be sufficient for the system and is set as the objective. For example, at a distance of 50 cm from the eye to the screen surface, this corresponds to (50 * tan(1°) =) 0.87 cm.
Temporal resolution	A temporal resolution of 60 Hz should be sufficient for usability purposes. As the system is software-based, this is primarily affected by the underlying hardware, see chapter 6.1.2.
Robustness and reliability	The robustness of the current prototype is largely dependent on the stability of the calibration of the Eyelink. As long as the Eyelink cameras, and the magnetic sensor attached to the head gear, stay fixed in relation to the eyes, the calibration should stay stable.
User restrictions and invasiveness	The system, except for the eye tracker calibration sequence, does not restrain the movement of the user or the screen. The only requirement is staying within 75 cm (30 in.) of the magnetic transmitter due to the working range of the magnetic tracker. Taking either of the magnetic sensors further away rapidly degrades the accuracy of the data gathered from the positional tracker, resulting in poor signal quality.
Data coverage	The system can be configured to provide the experimenter with all of the measured and calculated data.
Real time view of the current measurement	The user interface of the prototype provides the experimenter with a 3D view of the current measurement, and the gazepath of the user.

[1] The term "sighting center" refers to the point within the eye, through which all line-of-sights pass. See chapter 4.2.2 for a definition.

4.2 Principle of operation

To successfully measure the gaze point on a moving display surface, for any instant of time, three things need to be defined:

- the location of the sighting centers of both eyes in space
- the direction of gaze from the eye sighting center
- the position and orientation of the screen

A gaze point is defined as the intersection point of the gaze vector and the display surface. The on-screen gaze points for each eye (left and right, p_L and p_R in Figure 15) are defined by subtracting that location from the upper left-hand corner of the screen.

The gaze vector has its origin in the sighting center of the respective eye (see chapter 4.2.2 for definition), and its direction is defined with the use of calibrated reference vectors mapping the gaze angle reported by the eye tracker to a directional vector in real world space (see chapter 4.2.1).

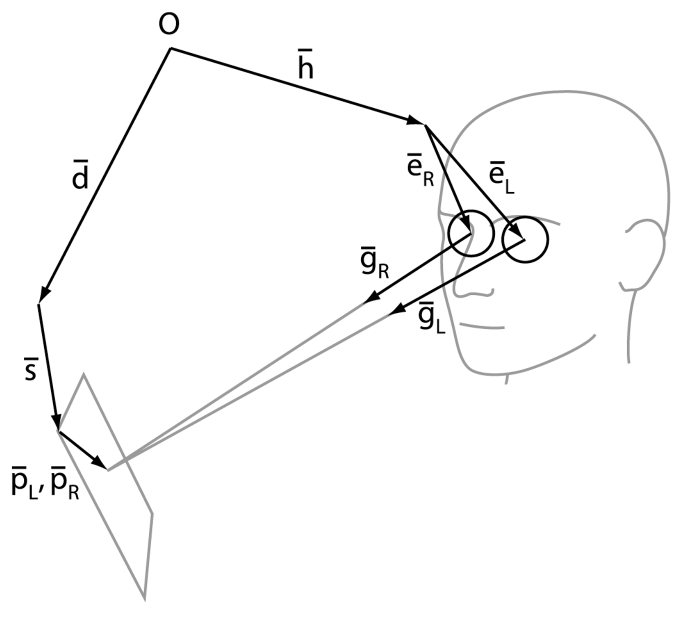


Figure 15 Principle of operation

The positional point of reference defines the origin *O*. The head sensor is located at the end of vector *h*, and the screen sensor at the end of vector *d*. The upper left hand corner of the screen is located at the end of vector *s*. The sighting centers of the eyes are defined in relation to the head sensor with vectors e_L (left eye) and e_R (right eye), defined through the calibration procedure. The gaze vectors g_L and g_R are delivered by the eye tracker. The mathematics of calculating the intersection point of a vector and a plane can be found in appendix 3.

The current implementation uses an Eyelink I head-mounted video-oculography device for measuring binocular gaze direction, and a Polhemus Fastrak magnetic positional tracker to measure the position and orientation of two magnetic sensors, attached to the head frame and the screen frame (from here on referred to as "head sensor" and "screen sensor"). These devices are described in more detail in chapter 4.3 below.

For each measured sample, the prototype system goes through the following cycle, presented in Figure 16.

First, the system acquires the newest samples available from the Eyelink (binocular eye data) and the Polhemus (position and orientation of both the head and screen sensors). The Eyelink sample is then transformed to a directional gaze vector through a linear transformation using three reference vectors from the calibration sequence (see chapter 4.2.1 below).

Next, the coordinates of the head sensor are used to calculate the position of each eye, utilizing the head sensor related eye position vectors obtained from the calibration. The position and orientation of the screen is calculated using screen sensor coordinates and the given sensor-to-screen calibration vector.

Finally, after combining the eye position and gaze direction vectors to obtain the gaze origin and direction, the gaze point on the screen is defined for each eye as the intersection point of the respective gaze vector and the screen surface.

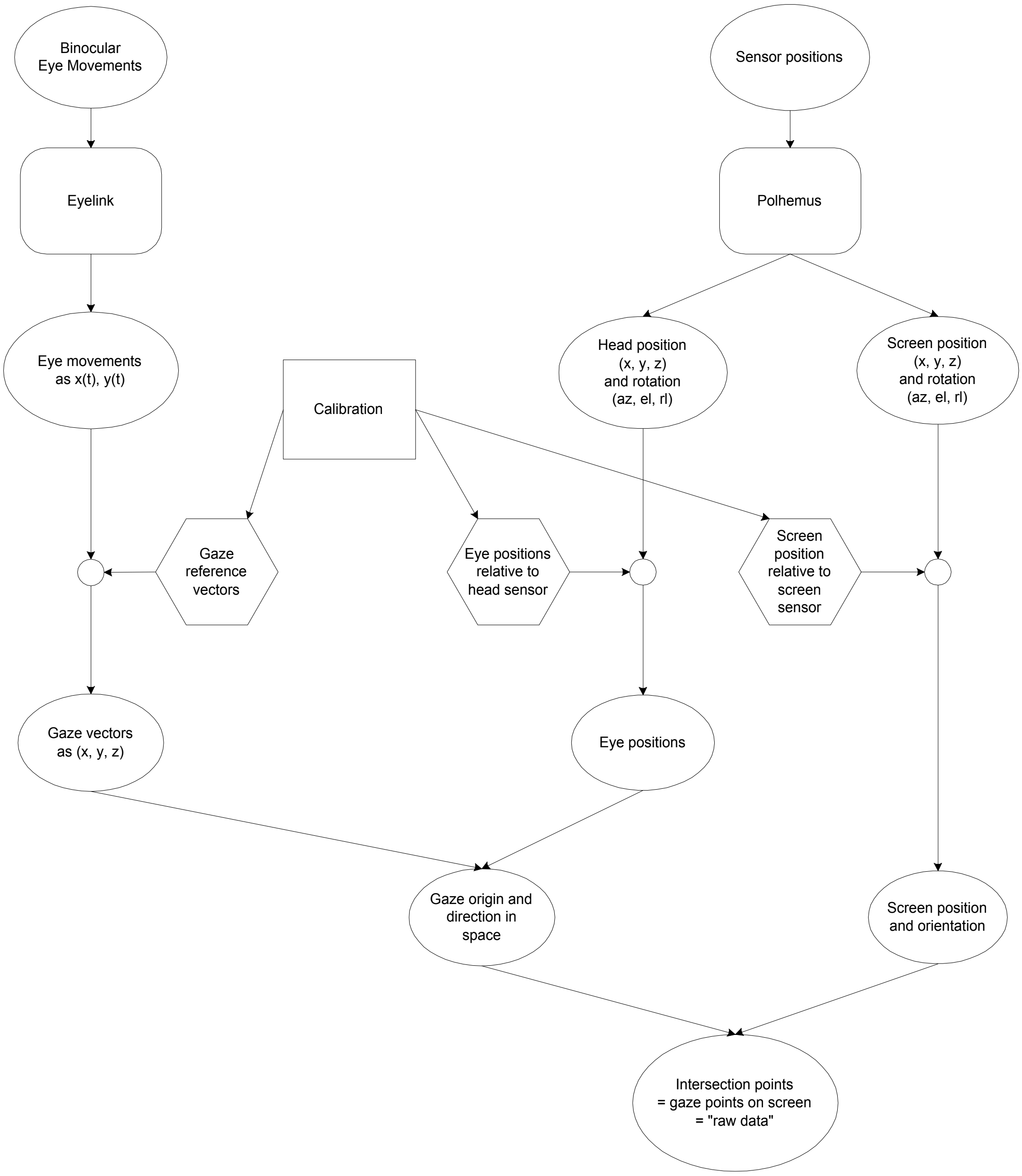


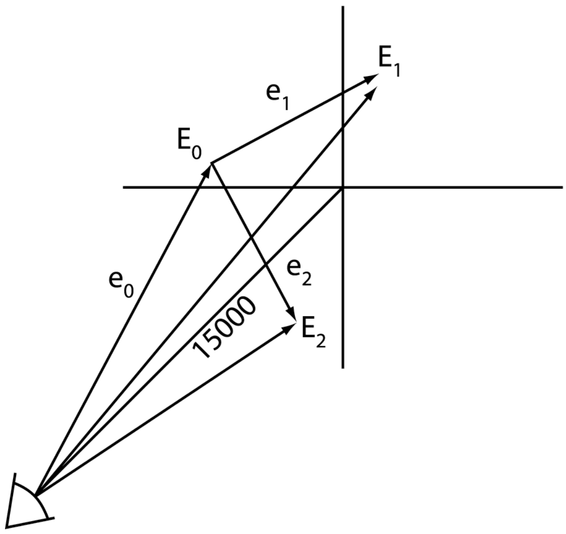
Figure 16 A diagram of the measurement cycle

The resulting data is the "raw data" produced by the system: time-stamped gaze points on the screen surface measured in pixels or centimeters. At a later phase this data can be refined further with the appropriate analysis method extracting for instance dwell times, scan paths, saccades, and fixations, or machine- or human-defined regions of interest.

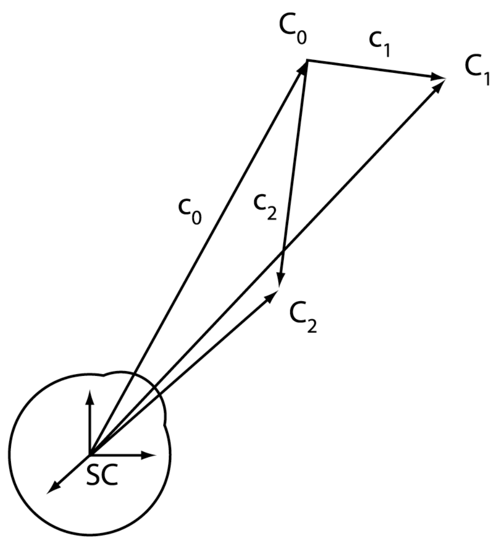
4.2.1 Mapping eye tracker coordinates to real world gaze vectors

The eye movement data obtained from the eye tracker must be mapped to correspond to a real world gaze vector. Eyelink delivers the gaze samples as coordinate pairs in a coordinate frame defined as being orthogonal to the eye and 15000 units in front of it. Originally intended for the measurement of saccade amplitude, this coordinate frame has no absolute position between different calibrations, and the mapping between this and the real world coordinate system has to be calibrated. The measure is mapped to correspond to a real world gaze direction in the coordinate frame of the sighting center of the eye through a linear transformation.

For the translation, three reference vectors are defined in both coordinate frames. The subject is presented with three targets (C_0-C_2) and instructed to fixate these, each in its turn. Ten samples from the Eyelink, and the corresponding real world vectors from the eye sighting center (SC) to the reference target, are averaged, and the resulting vectors (e_0, e_1 and e_2 for Eyelink and c_0, c_1 and c_2 for real world) used as the basis of the transformation.



Eyelink coordinate system



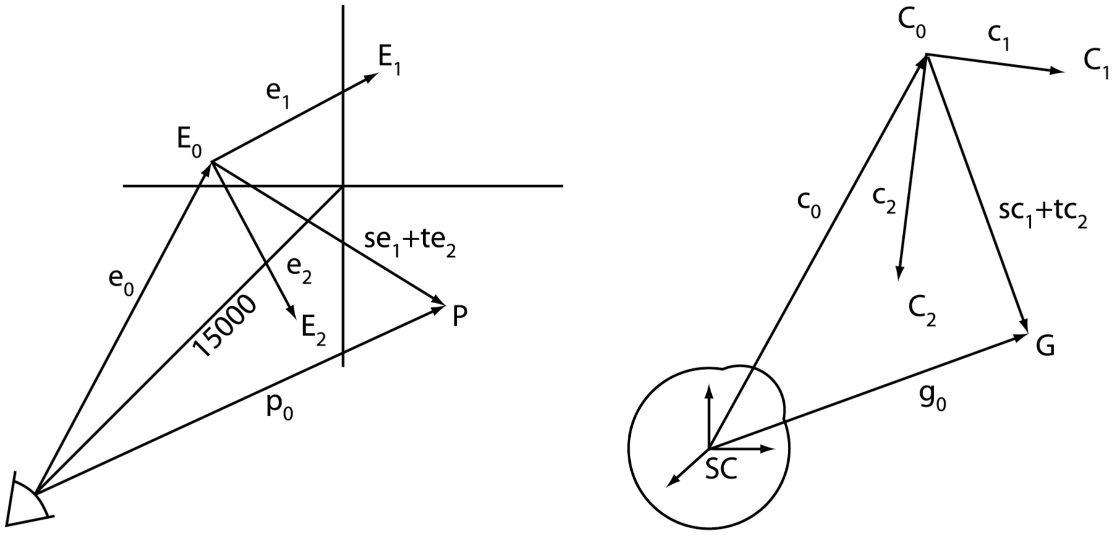
Eye sighting center coordinate system

Now, supplied that the two reference vectors (e_1 and e_2 for Eyelink; c_1 and c_2 for real world) in each coordinate frame are not parallel, any vector p_0 pointing to point P in these frames can be represented as a combination of those vectors multiplied by coefficients s and t. That is, for Eyelink,

$$p_0 = e_0 + se_1 + te_2$$

and in the real world frame

$$g_0 = c_0 + sc_1 + tc_2$$



Eyelink coordinate system Eye sighting center coordinate system

As the mapping between the two coordinate frames is linear, this enables the mapping of any coordinate pair delivered by the Eyelink tracker to be related to a unique gaze vector.

4.2.2 Location of the sighting centers

The sighting center is defined as the point within the eye, through which all the gaze vectors, or line-of-sights, pass. The concept has been founded in previous work (Park & Park, 1933; Epelboim et al, 1995). In Park & Park, this point was found to be approximately 13.5 mm behind the front surface of the cornea along the line-of-sight.

In Epelboim et al. (1995), the sighting center was located through the following procedure: With the subject's head held steady on a bite board, the subject looked through a thin tube with a sighting pinhole at one end, and a LED light located at the other end. The sighting tube was positioned so, that when the subject saw the LED, his/her line-of-sight was coincident with the axis of the sighting tube. Next, the subject closed his/her eye and the tube was moved towards it until it touched the eyelid. The subject then got off the bite board and a positional locator was placed 14 mm (13.5 mm behind the cornea + 0.5 mm allowed for the thickness of the eyelid) in front of the end of the tube nearest to the subject. The procedure was repeated, and the average coordinates obtained used as the location of the sighting center in subsequent calculations.

The method used here is somewhat less labor intensive. The procedure is as follows: The subject points a narrow tube attached to a magnetic sensor towards his/her eye so that the center line of the tube effectively points at the sighting center, that is, so that the subject can see the background light through the narrow tube. The center line of the tube is known in relation to the magnetic sensor. This position is then recorded in relation to the head sensor. Another line-of-sight vector is obtained in a similar manner, after moving the tube approximately 30°-40° of visual angle in relation to the eye (see Figure 17). Setting a large enough angle between the two vectors minimizes the error in calculating the intersection. The procedure is repeated for both eyes.

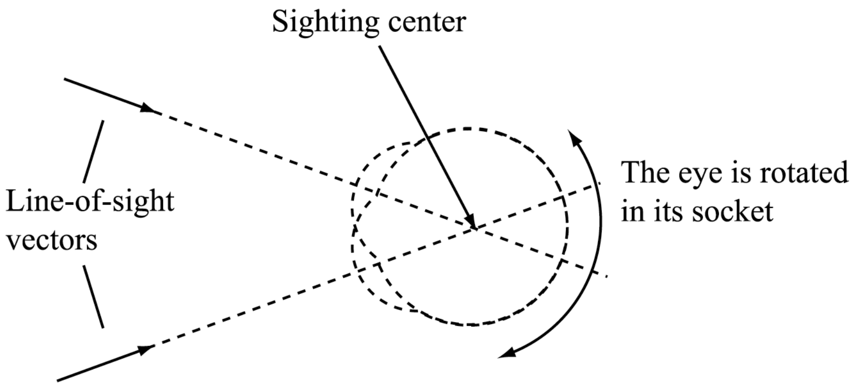


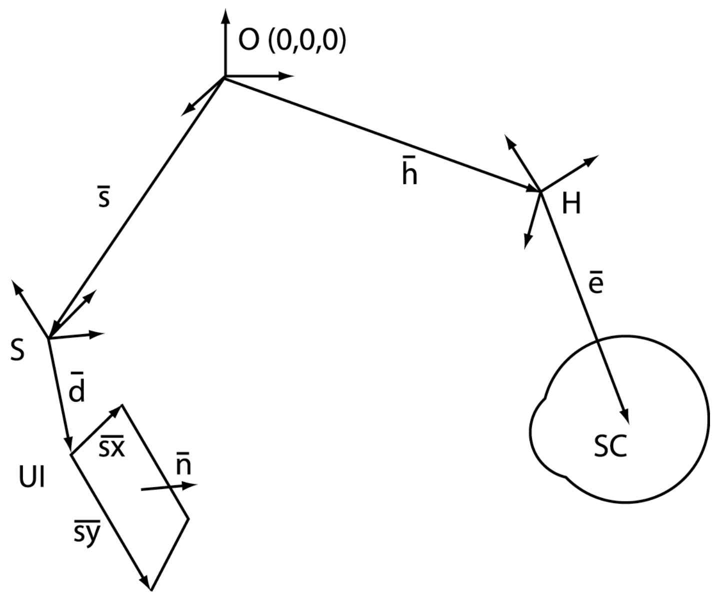
Figure 17 Defining the sighting center of the eye

The sighting center is then defined as the intersection or near intersection point of the two line-of-sight vectors. The near intersection point is the middle point of the shortest distance vector between two vectors. The mathematics of calculating the near intersection point can be found in appendix 3.

The goodness of the obtained location of the sighting center is estimated by the length of the shortest distance vector between the line-of-sight vectors. That is, the measure of how accurately the two vectors pinpoint a single point in space. Although the current aiming equipment is far from optimal, with a little practice by the subject, reproducible values of the three-dimensional location of the sighting center can be measured with an accuracy of less than 1 mm of disparity.

Unlike the method used by Epelboim et al. (1995), the procedure is immune to individual differences in the sizes of eyeballs between the subjects. The procedure could possibly be made more accurate and easier by developing better aims for defining the line-of-sight vectors.

4.2.3 Calculating eye position, and screen position and orientation



After the sighting center of each eye has been defined in relation to the head sensor (see chapter 4.2.2 Location of the sighting centers), the calculation of the position of each eye at a given time is basic 3D geometry.

The vector *h* points from the origin *O* to the head sensor H, whose translation matrix is defined as *H*. The vector *e* points from point H to the sighting center of the eye (SC). The location of the sighting center of the eye can thus be stated as

SC = *h* + *e***H*

The position of the screen can be calculated in a similar manner. That is, the location of the upper left hand corner of the screen

UI = *s* + *d**S

The screen orientation can be described with its normal vector *n*, which in turn is the cross product of vectors *sx* and *sy* defined in the test set-up in relation to the screen sensor

$$n = | sx \text{ x } sy |$$

The final step is to check that the screen normal *n* points at the direction of the user's eyes, and not away from them. This can easily be verified by checking that the dot product of the screen normal and a vector pointing from the upper left hand corner of the screen toward the eye sighting center (*se*, not presented for clarity in the picture above), is positive. That is,

$$| n \text{ o } se | > 0$$

4.3 Technical components

The choice of Eyelink as the eye tracker was guided by the fact that the system should be built on hardware existing in the laboratory. Eyelink is also known as one of the best eye tracking devices on the market. The choice of Polhemus Fastrak as the positional tracker was affected by its suitability to the application, and plans for additional future use in the laboratory.

Both systems place technical limitations on the current prototype, that could possibly be removed or alleviated by using other components. This has been taken into consideration in the design of the system – both trackers have been abstracted in the software, so that changing the underlying equipment is made as easy as possible.

4.3.1 Eyelink

The Eyelink I is a head mounted VOG device manufactured by SR Research. The Eyelink I is described in detail in appendix 2. Also, see chapter 2.3 for a detailed description of eye tracking methodology and equipment.

The system has an optical head movement compensation mechanism for tracking head movement in relation to the stimulus screen. In this work, however, the head movement compensation of the Eyelink system was disabled as redundant.

The Eyelink headband is equipped with two miniature, high speed cameras for tracking the eyes, and a third camera for tracking the stimulus screen. The image processing system processes the pupil and marker position in real time from the images acquired by the cameras, to compute true eye-rotation angles, gaze-position, and gaze-position resolution.

The system provides binocular measurements for horizontal and vertical eye positions and pupil size with a sampling rate of 250 Hz and a latency of 6-12 ms. The system has a linear tracking range of ±30º of horizontal and ±20º of vertical field of view, and delivers the gaze position with an accuracy of 0.5º - 1º of visual angle.

4.3.2 Polhemus Fastrak

The Polhemus Fastrak is a magnetic tracker manufactured by Polhemus Inc. The Polhemus Fastrak has been described in detail in appendix 1 and positional tracking methodology and equipment has been described in chapter 2.5.

The system delivers receiver position (x, y and z) with an accuracy of 0.8 mm RMS, and receiver orientation (azimuth, elevation and roll) with an accuracy of 0.15º RMS. The operational envelope is a hemisphere of 75 cm (30 in.) from the transmitter. The sample latency is reported as 4 ms and the update rate for sensor data is 120 Hz, divided by the number of receivers in use. In the current setup, with two sensors in use simultaneously, the effective sampling rate is thus 60Hz.

4.3.3 Headgear

During the project, it was discovered that the standard Eyelink headband is not stable enough for the current use. The headband slips around when the head is rotated and the cord extending from the backside creates a pull on the headband. This effectively corrupts calibration as the Eyelink eye cameras, and the Polhemus position sensor attached to the headband, move in relation to the eyes.

A better means of fixing the Eyelink eye cameras and the positional sensor in relation to the eyes, had to be developed. The resulting additional headgear is a modified scuba diving mask shown in Figure 18. The mask has a silicone frame that sits firmly and comfortably on top of the face. The Eyelink cameras are attached to the added horizontal bar on top of the frame, and the positional sensor sits on top of a plastic extension.

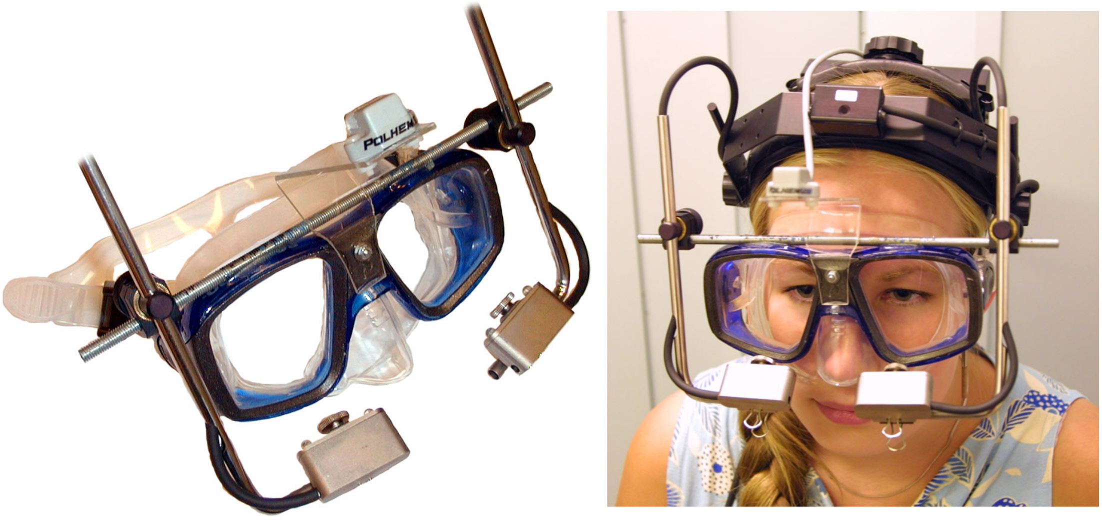


Figure 18 Headgear

However, because of other, ongoing studies conducted in the laboratory with Eyelink, it was not possible to dismantle the Eyelink headband. This resulted in added discomfort to the user, as currently during measurement the user has to wear both the original Eyelink headband, and the scuba diving mask.

4.4 Implementation

The prototype system was implemented as a Delphi[2] software project. Delphi was chosen as the development environment for efficiency and speed of development. Also, the writer has good previous experience with the Delphi Integrated Development Environment (IDE).

The Eyelink is supplied with a C-language Application Programming Interface (API). The development of the prototype system started with translating the Eyelink C API to a Delphi interface. This was made according to the recommendations and guidelines of Project JEDI, a joint endeavor of an international community to develop standard Delphi API libraries. The project homepage can be found at http://www.delphi-jedi.org/.

[2] Delphi is a registered trademark of Borland SoftwareCorporation

Two publicly available Delphi packages were used:

- ❖ Interfacing the Polhemus tracker through its serial interface was implemented using a freeware version of the Turbo Power Async Professional component available at *http://sourceforge.net/projects/tpapro/*
- ❖ The display of 3D data during measurement is implemented with GLScene, an open-source OpenGL based 3D library for Delphi, available at *http://www.glscene.org/*

After defining the requirements for the system, the final program structure was decided. The program follows the Object-Oriented Programming (OOP) approach. The structure is as follows: (see Figure 19; a more detailed description of the program components and their interfaces can be found from Appendix 4)

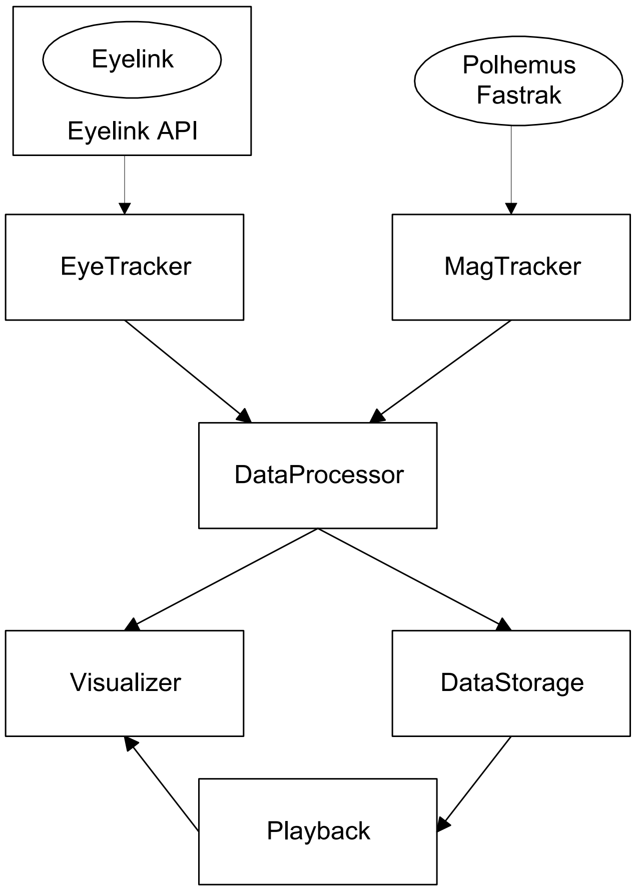


Figure 19 A diagram of the component model

The *EyeTracker* component takes care of the communication with the eye tracker. The component controls the initialization and calibration, starting and stopping the data gathering, and returning the current gaze vectors to the system. The current implementation communicates with the Eyelink through an Ethernet connection, using the Eyelink API. From the viewpoint of other program components, the eye tracker can be considered as a black box delivering eye related gaze vectors, and can be considered as an ordinary positional tracking device.

The *MagTracker* component communicates with the positional tracker. The component collects samples from the tracker to its buffer, and returns the position and orientation of each sensor to the system when requested. The current implementation communicates with the Polhemus Fastrak system through its serial interface.

Both trackers have been abstracted and separated as individual software components, so that replacing them with other available systems or tracking methods only requires changes to these software components.

The *DataProcessor* component is responsible for controlling the trackers, virtually all of the calculations, and passing the measured and calculated samples onward to the DataStorage and Visualizer components.

The *DataStorage* is responsible for storing data. During online measurements, the data is streamed to the computer memory, and after the measurement has stopped, the component stores the data to a binary file on the computer hard disk. The component also includes functions for converting the binary data to ASCII format for exporting to analysis software.

The *Visualizer* is responsible for

- online visualization of the measurement environment as a 3D representation
- visualizing the real time gaze point on the screen surface

The *Playback* component is responsible for streaming the data from a previously recorded session to the Visualizer for viewing and subjective analysis.

4.5 Screenshots & demonstration

This chapter presents the measurement system in operation.

Figure 20 shows the 3D view of the measurement. The subject's eyes are located at the endpoints of the two green vectors on the right hand side. The short vectors pointing from them are the direction of gaze vectors. The screen is represented by the violet surface in the picture. The red vector points to the location of the screen sensor, and the blue vector to the location of the head sensor. The white vector points at the intersection point of the gaze from the right eye and the screen surface. The picture shows the subject looking at the upper left hand corner of the screen.

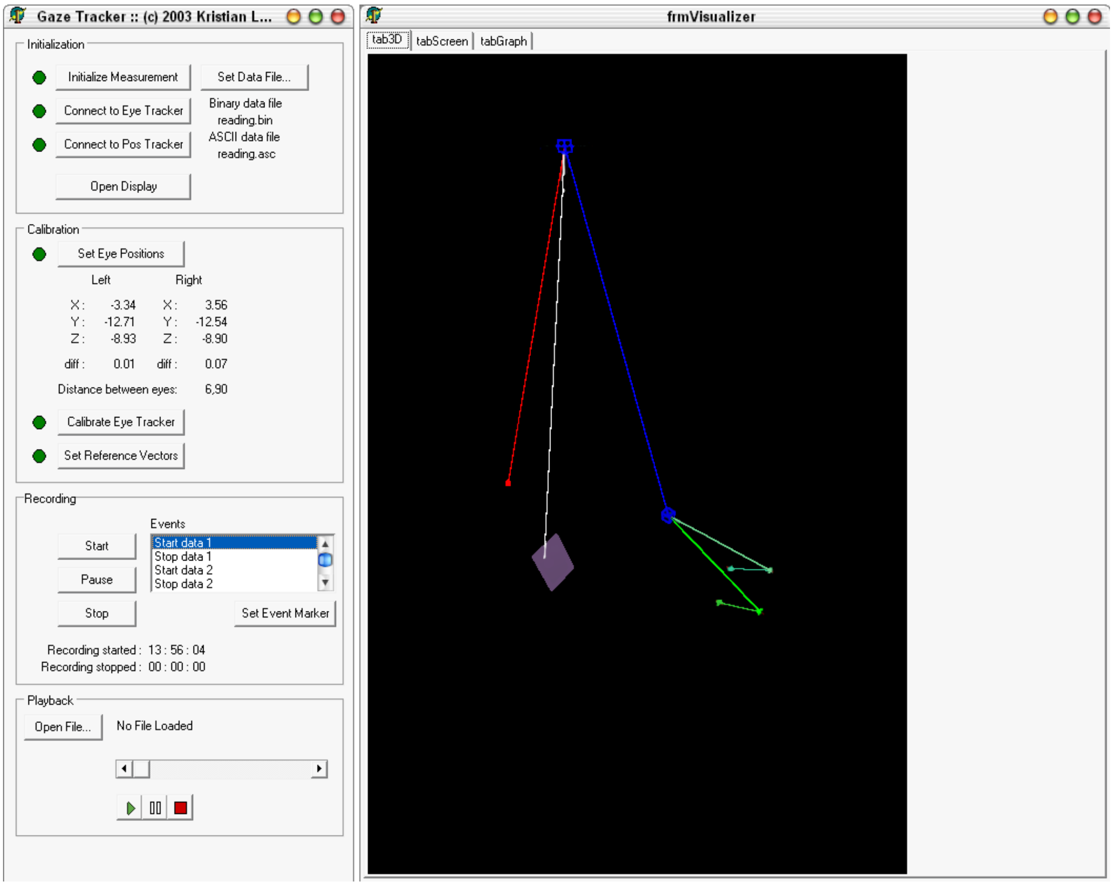


Figure 20 The 3D view shown during measurement

Figure 21 presents the view of the screen during measurement. The subject's gaze is represented by five transparent circles, showing the current gaze point and 4 previous gaze points. The subject is currently reading text on the upper section of the screen.

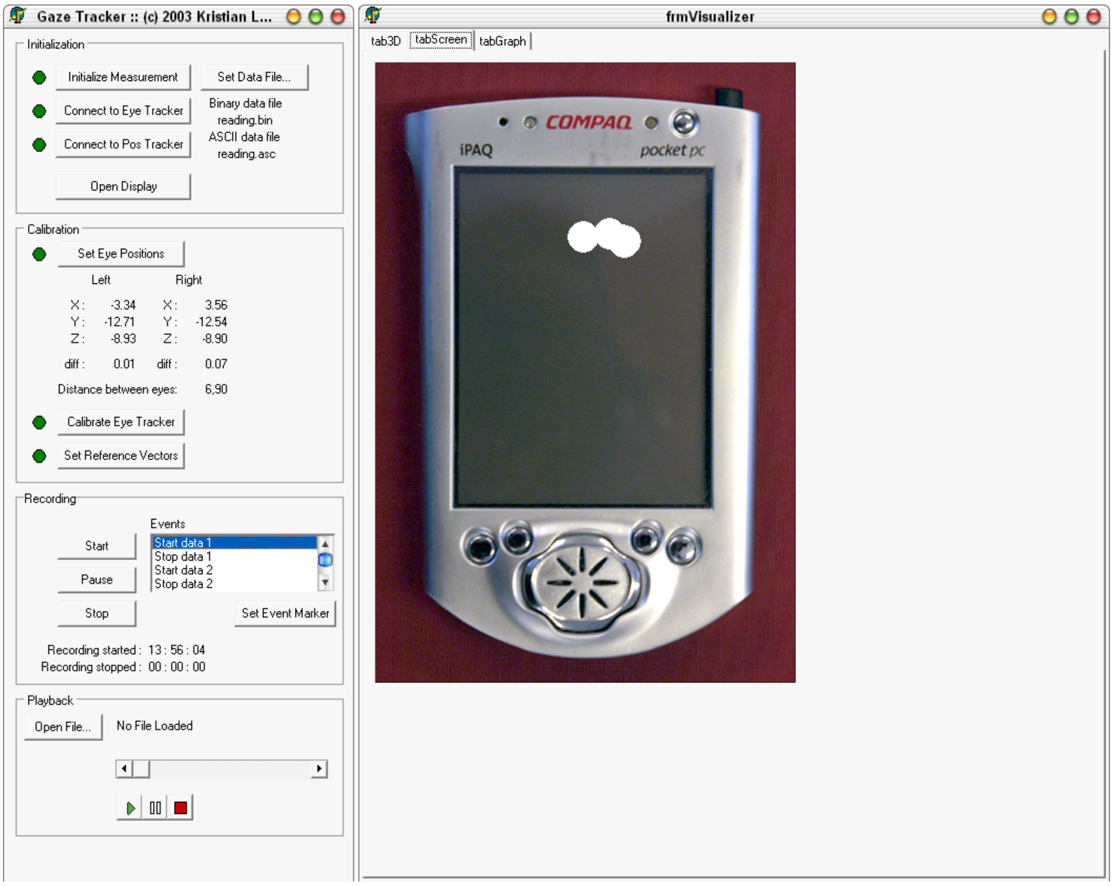


Figure 21 View of the screen during measurement

A demonstration version of the tracker application can be found on the CD-ROM included, in the folder \demo\. Run the executable file *Tracker.exe*. In the demonstration version, the measurement controls are disabled, leaving only the playback option available.

Two example data files are included in the folder \demo\data. The file *fixations.bin* contains a measurement session of a subject fixating two targets on the screen, as described in chapter 5.1. The file *reading.bin* contains a measurement session of a subject reading text from the screen, as described in chapter 5.2.

Load one of the data files by pressing the button *Open File*... The playback can be controlled with the buttons *play*, *pause*, and *stop*. The view between the 3D view and a view of the device can be changed from the tabs (3D, Screen) in the Visualizer window.

5 Prototype evaluation

At the current stage of development, no real experiments utilizing the system have been undertaken. The tests performed here aimed at evaluating the performance and specifications of the capabilities of the system.

Real experiment set-ups would demand at least

- further development of the system for
    - controlling and recording the stimuli presented on the mobile device, and
    - recording user actions on the user interface
- implementing fixation and saccade analysis, and possibly a number of higher level analyses

Though these developmental aspects are far from trivial, they were not determined as a part of this thesis, and will therefore be addressed in future development.

Two tests were performed with three subjects. These tests evaluated

a. fixation accuracy and stability
b. subjective accuracy in a simple reading task

The example figures were produced by plotting the data with Microsoft Excel[3] and Matlab[4], and superimposing the plotted data on an image of the stimulus screen in Adobe Illustrator[5]. This treatment in no way altered the data.

5.1 Fixation accuracy and stability

The test was performed with the subject fixating back and forth between two targets on the screen while holding the mobile device in a natural use position, sitting in a chair and changing his/her position during the test, i.e. moving both the screen and his/her head within the operational envelope of the positional tracker. In the subject-

[3] Excel is a registered trademark of the Microsoft Corporation

[4] Matlab is a registered trademark of the MathWorks Inc.

[5] Illustrator is a registered trademark of the Adobe Systems Inc.

chosen, comfortable posture during the test, the subject held the screen approximately 50 centimeters away from the eyes.

Figure 22 shows an example of the gathered data. The figure presents the positions of gathered gaze samples superimposed on the stimulus (the two fixation targets on screen). The histograms on the sides of the screen present the distribution of x- an y-co-ordinates of the gaze samples. The bars beneath the histograms, centered on the fixation targets, show the magnitude of one degree of visual angle on the screen at a distance of 50 cm.

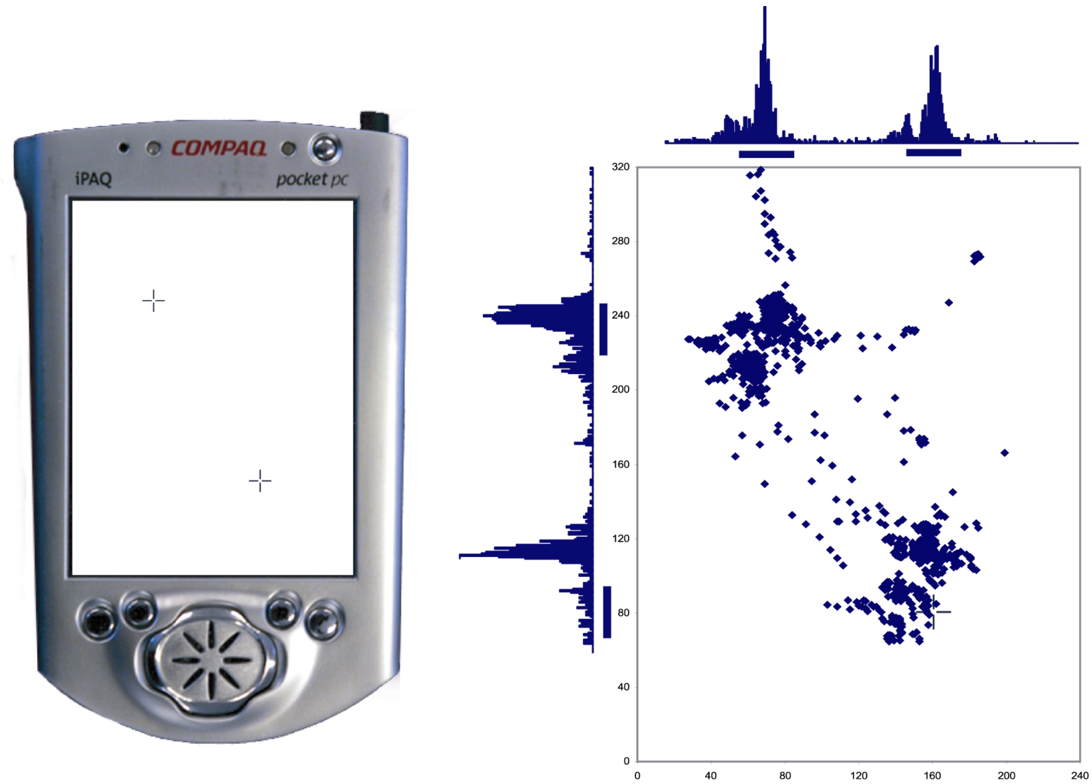


Figure 22 Example data for the fixation test

The device operated on a sampling rate of 30 Hz, gathering binocular gaze samples on the screen. It should be observed, that the data points are gaze samples, not fixations. As saccades take ca. 30-50 ms, the system triggers at least one sample during the saccade, which explains for the samples between the two targets. Natural fixation accuracy of about 1° offers an explanation for the distribution of gaze samples around the fixation targets. The distribution of the data points is such that in the x-direction, 82% and in the y-direction, 74% of the samples were within 1° of the fixation targets.

5.2 Subjective accuracy in a simple reading task

The second test was performed with the subject reading a short piece of text on the screen while, again, sitting in a natural use position and moving both the screen and his/her head within the operational envelope of the positional tracker.

The test aimed at subjectively evaluating the accuracy and validity of the data. Reading was chosen as a well-researched experiment paradigm giving a gaze path distinctive for reading. Reading is much more constrained and context-sensitive than, for instance, visual search in a graphical user interface.

Rayner (1998) provides several facts about eye movements in reading: Reading pattern differences are clear between novices and experts, and complexity of written material can be discerned from eye tracking data. Reading consists of a series of short saccadic eye movements, each spanning about 7-9 letter spaces for average readers. The majority of the words in a text are fixated during reading, although a number of short words may be skipped so that the foveal processing of each word is not necessary. Regressions occur in about 10-15% of saccades.

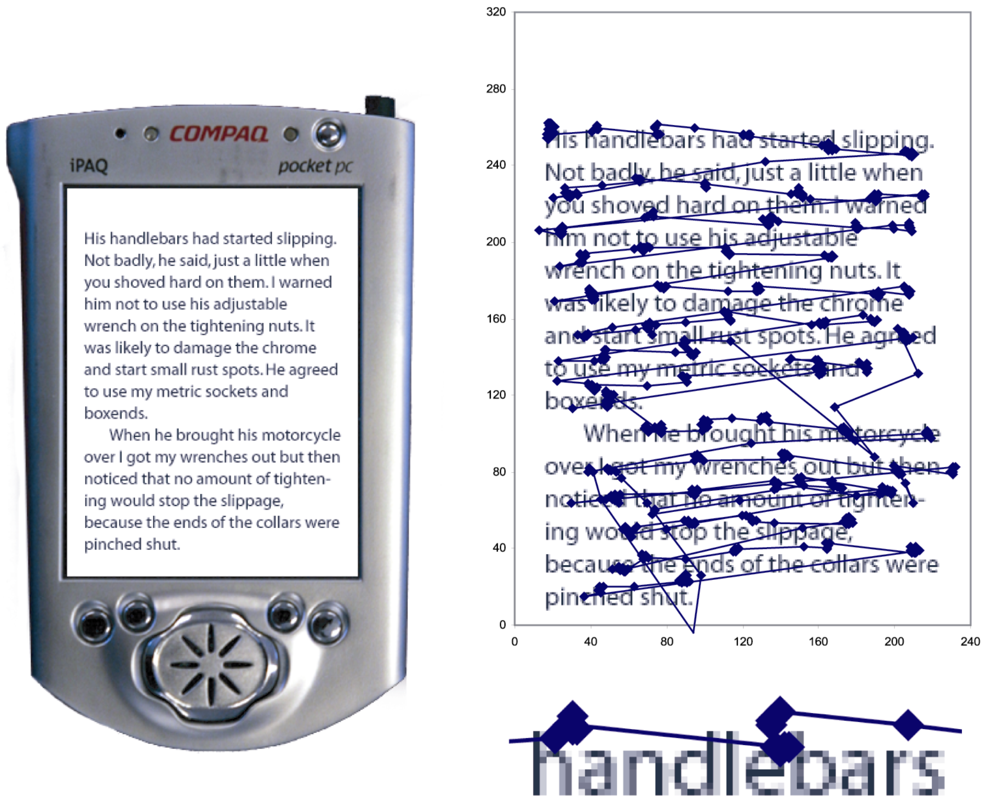


Figure 23 Example data for the reading test

The texts presented were two short excerpts, randomly chosen from the book “Zen and the Art of Motorcycle Maintenance” by Robert M. Pirsig.

Figure 23 shows an example of the gathered reading data. The figure shows a typical gaze path in a reading task with the text lines being fixated relatively accurately. One can even observe that, for instance, processing the second word “handlebars” took two fixations, between letters h and a, and between letters e and b. Zooming in to the data, and with the sampling rate being 30 Hz, by counting the samples (6 in the first fixation and 7 in the second), it can be deduced that reading the word took 13 * 33 ms = 430 ms. The data fits nicely in the description of reading by Rayner, above.

6 Results: Tracker performance

A prototype system was defined and implemented for tracking the point of gaze on a mobile screen or user interface. This chapter gives details of the performance of the currently implemented prototype system.

6.1.1 Linearity

The Eyelink I system provides a linear signal within ±30° horizontal and ±20° vertical gaze angle, which, at least for the time being, restricts the system operation to the same angular field of visual angle. However, for practical purposes this poses only minor problems, as the normal visual working range of a human, and the manner of operation of mobile devices, should naturally bring the screen within that area on most occasions. Taking the screen outside this central area of vision degrades performance significantly. Other than that, presuming that the magnetic field is homogenous and no ferrous objects distort the field, the system introduces no non-linear measures or transformations.

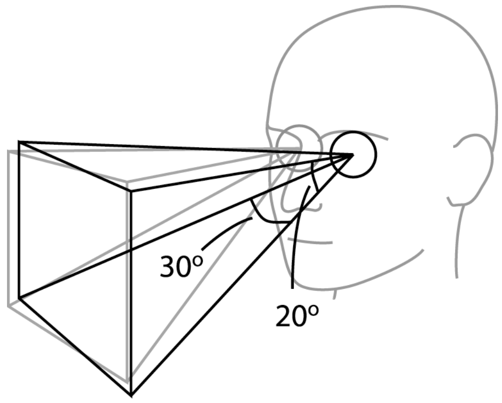


Figure 24 Eyelink linearity angles

6.1.2 Temporal resolution

As the system is software based, the performance heavily depends on the underlying equipment. Eyelink gives a temporal resolution of 250 Hz, whereas the Polhemus equipped with two sensors performs at a level of 60 Hz. As hand movements are significantly slower than eye movements, the current system settles for using the newest samples available from each device.

In post-measurement, offline analysis, the positional data could also be interpolated between samples, due to the relative slowness of hand movements when compared to eye movements, making it possible to resample the data so that the system functions "as fast as possible", with an upper limit of 250 Hz.

Timing tests were performed to define the operational temporal resolution of the system. Currently, operating on a 1.0 GHz Windows XP PC, gathering the data and performing the calculations took 4 ms. This would give the system a temporal resolution of 250Hz. Currently, however, updating the measurement display takes 20-25 ms, giving a total of 30 ms per measurement cycle, resulting in a sampling rate of 30 Hz. This sampling rate was also used in the performed tests. In the future, the slowness of the display system will be addressed. The latency of the system can be calculated as the 4 ms taken by the measurement loop, plus the latency of the Eyelink system, 6-12 ms, equaling 10-16 ms altogether.

6.1.3 Spatial resolution

The objective for spatial resolution was set as 1º of visual angle. The current system seems to accomplish this reasonably well. From the viewpoint of error considerations, see chapter 6.2, the accuracy of the gazepoint can be calculated with an accuracy of about 1º. The preliminary data gathered in the evaluation tests seems to support this.

6.1.4 Calculated data

The system can be configured to output any number of the data values described in Table 4.

Table 4 Data output values

Value	***Comments***
Timestamp	Milliseconds (ms) from the start of the test
Gaze point on screen surface	Binocular gaze point in pixels or centimeters
User defined events	Events such as trial beginnings and ends, button presses etc.
Current head and screen sensor position and rotation	These can be used in comparing gaze with physical movement and rotation of the head and screen
Current screen position and orientation	
Current binocular eye positions	
Current gaze vectors	

6.2 Error considerations

We demand rigidly defined areas of doubt and uncertainty!

-- Douglas Adams

The positional tracking and eye tracking devices introduce a variable amount of random error to the measurement. This error can be estimated by describing the system as a simplified vector chain model, see Figure 25. In the figure, the error components have been exaggerated on purpose. Table 5 presents values for accuracy as reported by the system manufacturers, and typical values for the prototype.

Table 5 Error estimation values

System component	*Measurement*	*Value*
Polhemus Fastrak	Sensor positional accuracy Sensor angular accuracy	d = 0.8 mm RMS α = 0.15º RMS
Eyelink	Gaze angle accuracy	β = 0.5 – 1.0º
Prototype	Eye sighting center position accuracy	sc = 1 mm
Typical values	head-sensor-eye distance screen-sensor-screen distance eye-screen distance	e = 120 mm s = 100 mm g = 500 mm

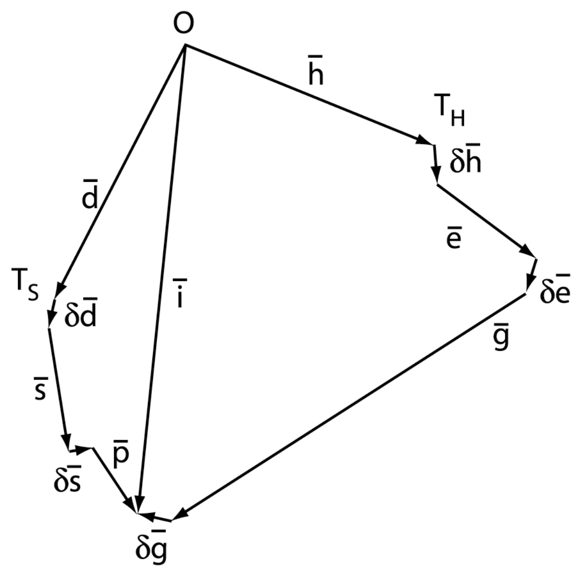


Figure 25 The simplified vector chain model

The intersection point (endpoint of vector *i*, see Figure 25) can be stated as

$$\bar{i} = \bar{d} + \bar{s} \cdot T_S + \bar{p}\text{, and}$$
$$\bar{i} = \bar{h} + \bar{e} \cdot T_H + \bar{g}$$

Differentiating these equations gives

$$\partial \bar{i} = \partial \bar{d} + \partial \bar{s} \cdot T_S + \partial T_S \cdot \bar{s} + \partial \bar{p}\text{, and}$$
$$\partial \bar{i} = \partial \bar{h} + \partial \bar{e} \cdot T_H + \partial T_H \cdot \bar{e} + \partial \bar{g}$$

which can be interpreted as the error generated in the vector chains. The total error can then be described as the sum of these, and by inserting the error estimation values given in Table 5, we get (in millimeters)

$$\Delta \bar{i} = \partial \bar{d} + \partial \bar{s} \cdot T_S + \partial T_S \cdot \bar{s} + \partial \bar{p} + \partial \bar{h} + \partial \bar{e} \cdot T_H + \partial T_H \cdot \bar{e} + \partial \bar{g}$$

$$\Delta \bar{i} = 0.8 + 1 + \tan(0.15) * 100 + 0 + 0.8 + 1 + \tan(0.15) * 120 + \tan(0.5) * 500$$
$$\Delta \bar{i} = 0.8 + 1 + 0.262 + 0.8 + 1 + 0.314 + 4.363$$
$$\Delta \bar{i} = 8.54$$

which, on a viewing distance of 50 cm gives an error of 0.98 degrees of visual angle.

This is the worst case scenario for perpendicular screen orientation, as all of the errors above are calculated as maximum errors, while they should follow the normal distribution. It should be noted that the error introduced by the Eyelink, even when estimated at the lower limit of 0.5º, dominates the error equation.

7 Conclusions and discussion

This chapter presents a summary of what was actually studied and performed in this thesis. The chapter also relates the results to the current status of the field of research. Finally, the chapter presents considerations for future developments for a gaze tracking system applicable to the use of mobile, handheld devices.

7.1 Summary

Tracking eye movements, or gaze position, could offer a valuable addition to the toolkit of usability research and analysis. They provide sensitive and accurate information on the direction of visual attention, information processing, and strategies for visual search and information collection methods of the user during a task, not available through the traditional methods of usability research.

Current eye tracking methods on the market tend to tacitly assume the stabilization of the head and the stimulus display during measurements. However, this results in non-realistic use scenarios and the study of gaze and eye movements in exclusion of the natural head, hand, and body movements.

The number of mobile devices with complex, small screen user interfaces is increasing rapidly, and the future trend of work and play seems to be towards the mobile. However, research of the implications of this development is lagging behind, partly because of the lack of proper research equipment.

This thesis was a development project of a novel research system, aiming to respond to the need of tracking gaze and visual attention while using a mobile device in a free-use environment.

The literature review and the performed questionnaire gave basis for the definition of an eye tracking device for use in usability measurements. This definition was then used to define requirements for the implemented tracker prototype, developed as a proof of concept. The resulting prototype system was then evaluated.

7.2 Discussion

Two objectives were set for this thesis

1. To recognize the needs and define requirements of a system for extending usability studies based on gaze tracking to the realm of mobile computing, and
2. To design and implement a working prototype of such a system.

The defined requirements are presented in chapter 3.2. The *quick and easy set-up* and *accurate calibration* resulting in *good spatial accuracy* was seen as the most important property. *Good spatial resolution* is needed to accurately match gaze position to distinctive portions of the user interface or stimuli. *Robustness and reliability* in practice demand no dropped frames during measurement, and no repeated tests due to device malfunctions. The system should preferably set *no restrictions for the user* during measurements and *not make contact* with the user. The system should provide the experimenter with *extensive data*, although a high sampling rate for use in usability research was not seen as a priority. For subjective evaluation of the ongoing test, the system should provide the experimenter with a *real-time view* of the user's scan path.

The evaluation of the implemented prototype system shows, that the prototype is able to track gaze with a spatial accuracy of about one degree of visual angle. As the system is software based, the temporal resolution depends heavily on the underlying equipment. Currently, operating on a 1.0 GHz Windows XP PC, the measurements and calculations can be performed with a sampling frequency of about 250 Hz. Updating the display currently slows down the process considerably.

Due to the performance issues of the third party eye tracking device used, a linear signal can only be guaranteed within the central viewing range of 30º horizontal and 20º vertical angle. Also, the error analysis shows, that the error equation is dominated by the error generated in the gaze direction measurements of the eye tracking device.

The prototype fulfills the defined requirements acceptably. Due to the techniques used for tracking eye and head movements, making contact with the user was unavoidable. The calibration procedure requires user co-operation, which results in the need for trained subjects. Replacing the current eye tracker with another solution could make the headgear more comfortable for the user.

7.3 Future research

The future development of the system has two immediate goals:

- Developing a component for presenting and controlling the stimuli presented on a handheld device, and recording user interactions, and
- Developing analysis software for fixation and saccade analysis, plus a number of higher level analysis (see, for instance, the list by Karn et al. (1999), in chapter 2.4.1)

The first goal should also include the development of device-specific interfaces for exchanging information with different handheld devices. The second goal includes the non-trivial definition for basic eye movement measures, such as fixations and saccades, still under debate in the research community. The whole industry desperately needs standard definitions!

The system enables the future research of the usability aspects of mobile user interfaces in natural use positions, without restrictions to the users' natural movements. In the near future, the system should be validated, to identify the operational requirements and limits of the system.

Potential future uses for the device include measurements of dynamic vision, e.g. using a mobile phone while walking, and extending measurements from the current implementation, with a screen surface, to measurements of gaze position over three dimensional objects. The use of gaze position as input in an interface should also be of interest in the research of future user interface techniques.

8 References

Abrams R.A., Meyer D.E., Kornblum S. (1989) *Speed and accuracy of saccadic eye movements: Characteristics of impulse variability in the oculomotor system*. Journal of Experimental Psychology: Human Perception and Performance, 15. p. 529-543

Benel D.C.R., Ottens D. Jr., Horst R. (1991). *Use of an eye tracking system in the usability* laboratory, in Proceedings of the Human Factors Society 35th Annual Meeting, vol. 1. Human Factors and Ergonomic Society, Santa Monica. p. 461-465

Berg P., Scherg M. (1991). *Dipole models of eye movements and blinks*. Electroencephalography and Clinical Neurophysiology. Vol. 79. Issue 1. p. 36-44

Bhatnagar D.K. (1993). *Position trackers for head-mounted display systems: A survey*. Technical Report: TR93-010. University of North Carolina. Chapel Hill, USA.

Carpenter R.H.S. (ed.) (1991). *Eye Movements*, in Vision and visual dysfunction, vol. 8, Cronly-Dillon J.R., Editor. Macmillan, London.

Clarkson T. (1989). *Safety aspects in the use of infra-red detection systems*. International Journal of Electronics, 66. p. 929-932

Collewijn H. (1999). *Eye movement recording*, in Vision Research, a practical guide to laboratory methods, Carpenter R., Robson J. G., Editors. 1999 Oxford University Press, New York. p. 245-285.

Crosby M.E., Peterson, W.W. (1991). Using eye movements to classify search strategies. Proceedings of the Human Factors Society 35th Annual Meeting, vol. 2. Human Factors and Ergonomic Society, Santa Monica. p. 1476-1480

Crowe E.C., Narayanan N.H. (2000) *Comparing interfaces based on what users watch and do*. Eye Tracking Research & Applications Symposium 2000 Conference Proceedings. ACM Press.

Dodge R. and Cline T. (1901). *The angle velocity of eye movements*. Psychological Review, vol. 8, p. 145-157.

Dodge R. (1903). *Five types of eye movement in the horizontal meridian plane of the field of regard*. American Journal of Physiology, vol. 8, p. 307-329.

Driesen J.L. (2003). *Primary visual pathway*. Online, available at http://www.driesen.com/primary_visual_pathway.htm. Referenced 25.10.2003.

Duchowski A.T., Medlin E., Cournia N., Gramopadhye A., Melloy B., Nair S. (2002). *3D Eye movement analysis for VR visual inspection training*. Conference proceedings of Eye tracking Research and Applications Forum 2002. ACM Press.

Duchowski A.T. (2003) *Eye tracking methodology*. Springer-Verlag Ltd. London.

Engbert R., Kliegl R. (2003). *Binocular coordination in microsaccades*. In, *The mind's eye: Cognitive and applied aspects of eye movement research.* Hyönä J., Radach R., & Deubel H., editors. Elsevier Science, Amsterdam. s. 103-117

Epelboim J., Steinman R., Kowler E., Edwards M., Pizlo Z., Erkelens C. J., Collewijn H (1995) *The function of visual search and memory in sequential looking tasks*. Vision Research, vol 35. No. 23/24. p. 3401-3422. 1995 Elsevier Science Ltd.

Fenn W. O., Hursh J. B. (1934). *Movements of the eyes when the lids are closed*. American Journal of Physiology, vol. 118. p. 8-14.

Frøkjær E., Hertzum M., Hornbæk K. (2000). *Measuring usability: are effectiveness, efficiency, and satisfaction really correlated?* In, Conference on Human Factors in Computing Systems, CHI 2000 Conference Proceedings. ACM Press. p. 345-352.

van der Geest J.N., Frens M.A. (2001). *Recording eye movements with video-oculography and scleral search coils: a direct comparison of two methods*. Journal of Neuroscience Methods, 114. Elsevier Science Ltd. p. 118 – 195.

Goldberg, M.E, Eggers H.M., Gouras P. (1991) *The Ocular Motor System*, in Principles of neural science, Kandel, E.R., Schwartz, J.H., Jessell, T.M., Editors. 1991, Appleton & Lange. Connecticut. p. 600-677.

Goldberg J.H., Kotval X.P. (1999). Computer interface evaluation using eye movements: methods and constructs. International Journal of Industrial Ergonomics 24. Elsevier Science Ltd. p. 631-645

Goldberg J.H. (2000). *Eye movement-based interface evaluation: What can and cannot be assessed?* In Proceedings of the IEA 2000/HFES 2000 Congress (44th Annual Meeting of the Human Factors and Ergonomics Society) HFES, Santa Monica. p. 625-628.

Goldberg J.H., Wichansky A.M. (2003) *Eye tracking in usability evaluation: A practitioner's guide*, in The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research. Hyönä J., Radach R., Deubel H. Editors. 1st edition. Elsevier Science. Amsterdam. p. 493-516

Gonzalez R., Woods R. (1992). *Digital Image Processing*. 1st ed. Addison Wesley, 1992

Grüsser O.J. (1983). *Vision and Eye movements*, in Human Physiology. Schmidt R.F. and Thews G., editors. Springer-Verlag, Heidelberg. P. 234-273.

Haines D.E., Editor. (1997). *Fundamental Neuroscience*. Churchill Livingstone. p. 265-285

Hallett, P. E. (1986). *Eye movements*, in Handbook of Perception and Human Performance I. Boff K., Kaufman L. & Thomas J., editors. John Wiley & Sons, New York. p. 10.25-10.28

Heller D. (1988). *On the history of eye movement recording*, in *Eye movement research: physiological and psychological aspects*, Lüer G., Lass U., Shallo-Hoffman J., Editors. 1988 C. J. Hogrebe, Toronto. p. 37-51.

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

Jacob R.J.K. (1993). *Eye Movement-Based Human-Computer Interaction Techniques: Toward Non-Command Interfaces*, in Advances in Human-Computer Interaction, Vol. 4, Hartson H.R. and Hix D., editors. Ablex Publishing Co., Norwood, N.J. p. 151-190

Jacob R.J.K, Karn K.S. (2003) *Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises*, in The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research. Hyönä J., Radach R., Deubel H. Editors. 1st edition. Elsevier Science. Amsterdam. p. 573-605

Joyce C.A., Gorodnitsky I.F., King J.F., Kutas M. (2002) *Tracking eye fixations with electro-ocular and electroencephalographical recordings*. Psychophysiology, 39. Cambridge University Press. p. 607-618.

Jung R. (1977). *An appreciation of early work on gaze control in man and visuo-vestibular research before 1940*, in Control of Gaze in Brain Stem Neurons, Baker R., Bertoz A., Editors. 1977 Elsevier, Amsterdam.

Just M.A., Carpenter P.A. (1984). *Using eye fixations to study reading comprehension*, in New Methods in Reading Comprehension Research. Kieras D.E., Just M.A., Editors. Lawrence Erlbaum Associates. Hillsdale, New Jersey. p. 151-182.

Karn K.S., Ellis S., Juliano C. (1999). *The hunt for usability*. Workshop conducted at CHI'99 Conference of the Computer-Human Interaction Special Interest Group. CHI'99 Extended Abstracts. ACM Press, New York.

Kuutti K. (2000). [in Finnish] *Käyttöliittymä- ja käytettävyystutkimuksen haasteet*, in Miten käytettävyys muotoillaan? Keinonen T, Editor. Taideteollinen korkeakoulu, Helsinki. p. 79-91.

Lee C. (1999). *Eye and head coordination in reading: roles of head movement and cognitive control*. Vision Research 39. Elsevier Science Ltd. p. 3761-3768

Noton D., Stark L.W. (1971) *Scanpaths in eye movements during pattern perception.* Science, vol. 17, no.1. p. 308-311.

Park R.S., Park G.E. (1933). *The center of ocular rotation in the horizontal plane*. American Journal of Physiology, 104. p. 545 – 552.

Pelz J.B., Canosa R., Babcock J. (2000). *Extended tasks elicit complex eye movement patterns*. Eye Tracking Research & Applications Symposium 2000 Conference Proceedings. ACM Siggraph.

Pelz J.B., Canosa R. (2001). *Oculomotor behavior and perceptual strategies in complex tasks*. Vision Research 41. Elsevier Science. p. 3587-3596

Polhemus (2000). *3Space Fastrak User's Manual*. Rev. A. Polhemus Inc. Vermont, USA.

Pongs U.M. (1998). *Projekt Bildverarbeitung Purkinje I-IV Reflexmuster*. Online, available at http://www.medizin.uni-tuebingen.de/~jcbarry/bildbea.htm. Referenced 25.10.2003

Rayner, K., Pollatsek, A. (1989) *The Psychology Of Reading*. Prentice Hall, New Jersey.

Robinson D. A. (1963). *A method of measuring eye movement using a scleral search coil in a magnetic field*. IEEE Transactions of Biomedical Electronics, BME-10. p. 137-145.

Salvucci D.D., Goldberg J.H. (2000). *Identifying fixations and saccades in eye-tracking protocols*. Eye Tracking Research & Applications Symposium 2000 Conference Proceedings. ACM Press. p. 71-78

Shackel B. (1960). *Note on mobile eye viewpoint recording*. Journal of the Optical Society of America, vol. 50, number 9. p. 763-768.

Sibert L.E., Jacob R.J.K. (2000). *Evaluation of eye gaze interaction*. Proceedings of CHI'2000 Human Factors in Computing Systems. ACM Press. P. 281-288

Skalar website (2003). Online, available at http://www.skalar.nl. Referenced 10.11.2003.

Smith V.C., Pokorny J. (1975). *Spectral sensitivity of the foveal cone photopigments between 400 and 500 nm*. Vision Research, vol. 15. p. 161-171.

Sobotta. (1989). *Atlas of Human Anatomy*, vol. 1. 11th English Edition. Urban & Schwarzenberg, Munich.

SR Research. (1996). *Eyelink System Documentation*, version 1.1. SR Research Ltd., Canada.

Stanford-Poynter Project. (2000). *Eye tracking online news*. Available online at http://www.poynterextra.org/et/i.htm. Referenced 24.10.2003.

Stark L., Ellis S. R. (1981) *Scanpaths revisited: Cognitive models direct active looking*, in Eye movements: Cognition and visual perception. Fisher D., Monty R., Editors. 1981 Lawrence Erlbaum Associates. Hillsdale, New Jersey. P. 193-226.

Viviani P. (1990). *Eye movements in visual search: Cognitive, perceptual and motor control aspects*, in Eye movements and their role in visual and cognitive processes. Kowler E. Editor. Elsevier Science, Amsterdam. p. 353-394

Wade N.J. (2002) Guest editorial in Perception, vol. 31. p.643-650

Ware C., Mikaelian H.H. (1987). *An evaluation of an eye tracker as a device for computer input*. Proceedings of CHI+GI'87 Human Factors in Computing Systems. ACM Press. p. 183-188

Weedman Molavi D. (1997) *Neuroscience Tutorial*. The Washington University School of Medicine. Online, available at http://thalamus.wustl.edu/course/ Referenced 08.10.2003.

Young L. R., Sheena D. (1975) *Survey of eye movement recording methods*. Behavior Research Methods & Instrumentation, vol. 7. 1975. p. 397-429.

Zülch G., Stowasser S. (2003) *Eye tracking for evaluating industrial human-computer interfaces*, in The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research. Hyönä J., Radach R., Deubel H. Editors. 1st edition. Elsevier Science. Amsterdam. p. 531-553

Appendix 1
Polhemus magnetic tracker

This text is largely based on the Polhemus 3Space Fastrak user's manual (Polhemus Inc., 2000) delivered with the system.

Introduction

The Polhemus Fastrak[6] is an electromagnetic tracker giving accurate measurements in six degrees of freedom, in real-time. These degrees are the X, Y and Z Cartesian coordinate locations of the sensor in relation to the transmitter unit, and the azimuth, elevation and roll angles of the sensor at that location.

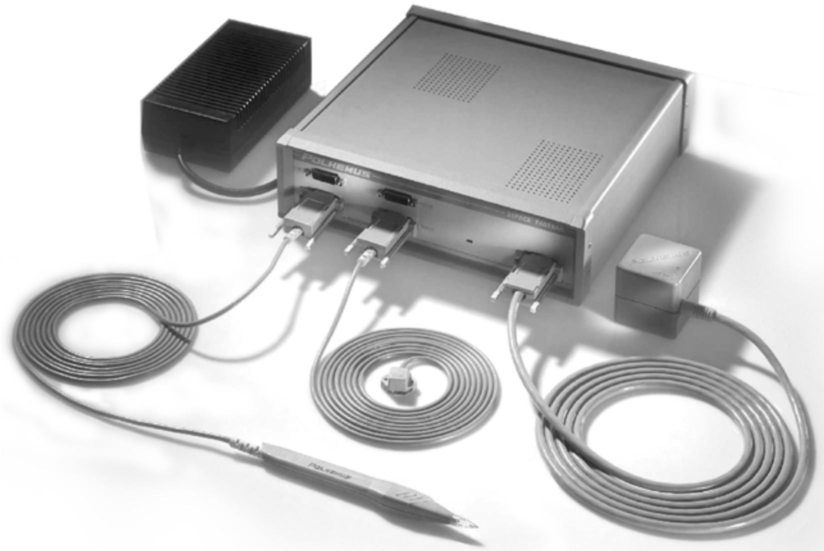


Figure 26 Polhemus 3Space Fastrak System (Polhemus, 2000)

The Polhemus Fastrak is widely used in various applications ranging from biomechanical analysis to virtual reality. A single system is capable of tracking one to four sensors simultaneously. Four systems can be cascaded, so that up to 16 sensors can be tracked with no change in update rate. The Polhemus Fastrak uses patented low-frequency magnetic transducing technology.

[6] Polhemus Fastrak is a registered trademark of Polhemus Inc.

Technical specifications

The system specifications promise accurate positional coverage within a range of 0.75 meters from the transmitter and a range of up to 3 meters with slightly reduced performance. Sample latency is reported as 4 milliseconds and the update rate for sensor data is 120 Hz divided by the number of receivers in use.

With the Fastrak system, Polhemus promises to deliver static accuracy of 0.8 mm RMS for the X, Y and Z positions, and 0.15° RMS for receiver orientation. The spatial resolution is defined as 0.002 mm per centimeter of transmitter and receiver separation. The receivers are all-attitude with no limits in angular coverage.

Large metallic objects located near the transmitter or receiver can adversely affect the operation, as they distort the magnetic field produced by the transmitter.

Theory of operation

The position of a point in a three-dimensional space can be fully described by its relation to any fixed three axis (x,y,z) coordinate system. The orientation of an object at that point can be defined as a direction in relation to that position, which can be fully described by three angles known as *azimuth*, *elevation* and *roll*, also known as yaw, pitch and roll.

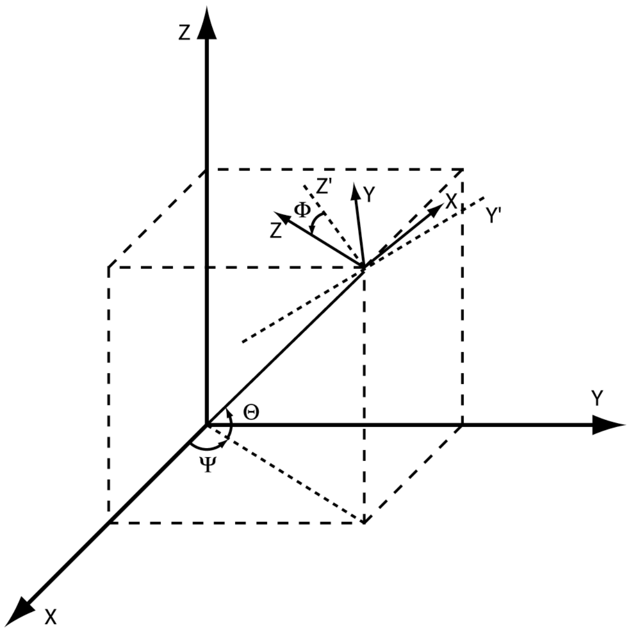


Figure 27 Euler rotation angles

The Fastrak tracking system uses electromagnetic fields to determine the position and orientation of a remote object. The operation is based on generating near field, low frequency magnetic field vectors from a single assembly of three concentric, stationary antennas called a transmitter. These field vectors are then detected with a single assembly of three concentric, remote sensing antennas called a receiver.

The signals induced to the receiver antennas by the transmitted electromagnetic field are input to a mathematical algorithm that computes the receiver's position and orientation relative to the transmitter.

The system uses an alternating current (AC) cycle pattern to generate fields in turn for each of the three concentric coils positioned orthogonal in respect to each other. The coil diameters are kept very small compared to the distance between the transmitter and the receiver, so that each coil can be regarded as a point or infinitesimal dipole.

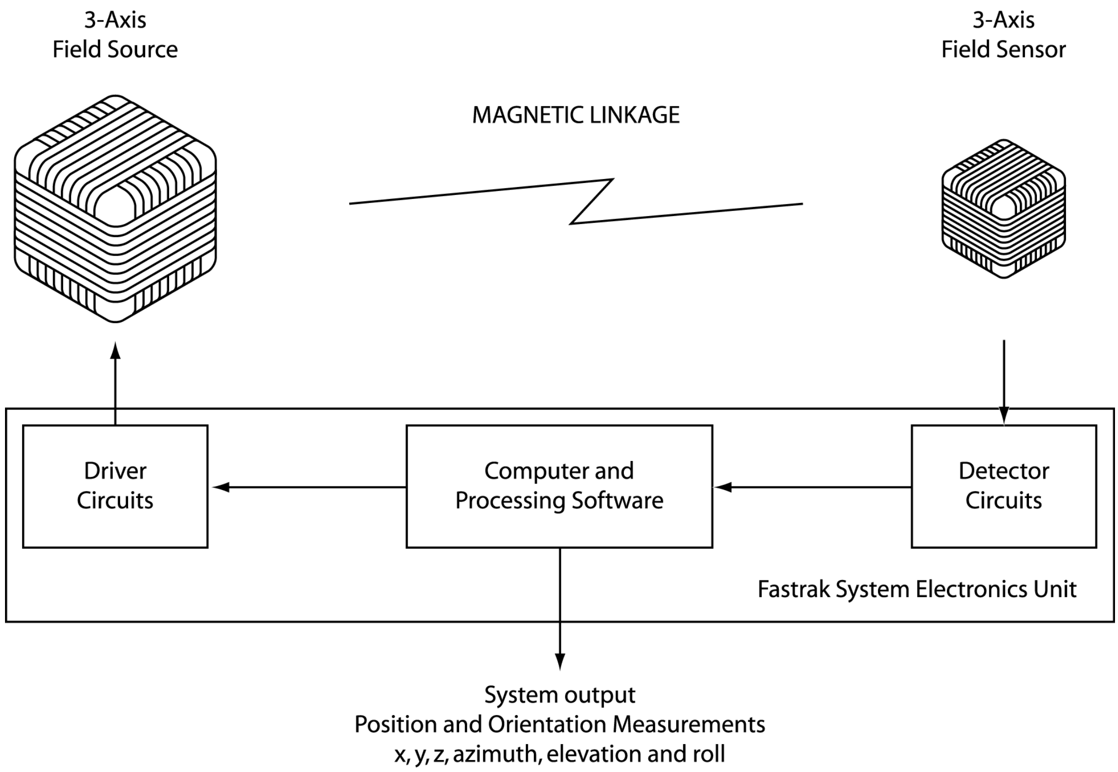


Figure 28 Polhemus Fastrak system block diagram

Exciting a coil antenna with electric current produces a field consisting of a far-field component and a near or induction-field component. The far-field intensity is a function of coil size and excitation frequency, and it decreases with the inverse of the distance (1 / r). The "quasi-static" near field component intensity is not frequency dependent and decreases by the inverse cube of the distance ($1 / r^3$). The near field is not detectable at long distances, but dominates at short distances where the far field is negligible.

Hemisphere symmetry

At any one time, the measurement can utilize only half of the total spatial sphere surrounding the transmitter because of the inversion symmetry of the magnetic fields generated by the transmitter. There are two possible mathematical solutions for the X, Y and Z position coordinates for each set of receiver data processed.

The system is unable to determine which of the two solutions is correct without additional information. Therefore the hemisphere of operation has to be defined. This also means that the receiver unit must be placed accordingly in relation to the measurement setup.

The orientation coordinates do not have a two-solution spherical ambiguity, and are therefore valid throughout the operating sphere centered at the transmitter.

Data output

The Polhemus Fastrak outputs data trough a serial RS232 interface. The device can be configured to output data in binary and ASCII formats.

The sampling rate of the Polhemus Fastrak is 120 Hz divided by the number of sensors used.

Appendix 2
Eyelink I eye tracking system

This text is largely based on the Eyelink I System Documentation (SR Research, 1996) delivered with the system.

Introduction

The Eyelink[7] system is an advanced VOG eye tracking system used worldwide in research fields including psychology, neurology, ophthalmology, ergonomics, human factors, cognition, and reading. Besides its use in analysis, the system permits the design and implementation of eye-guided applications. The system provides binocular measurements for horizontal and vertical eye positions and pupil size.

Technical specifications

The Eyelink system consists of

- a headband with two custom-built, miniature high-speed cameras
- a "Subject PC" for presenting experiments
- an "Operator PC" for controlling experiments and storing data, fitted with
- an Eyelink Image Analysis Board

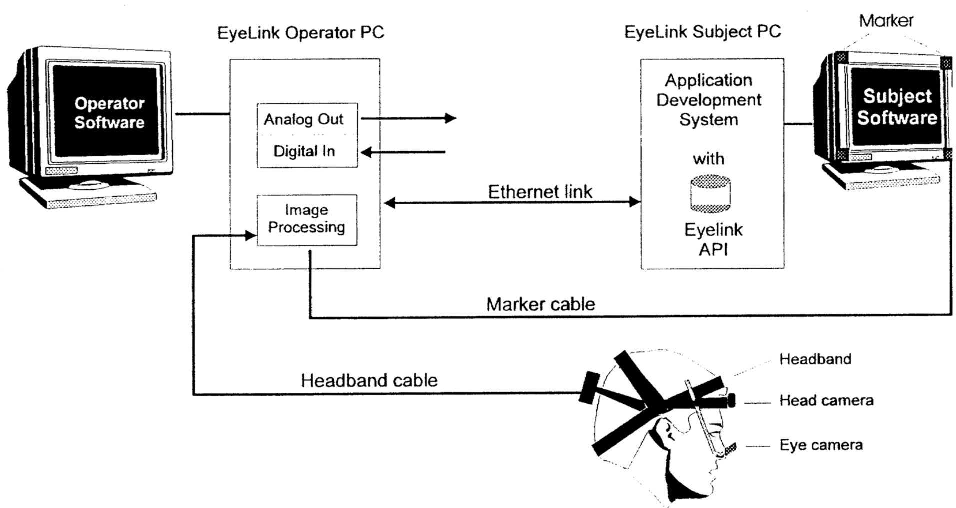


Figure 29 The Eyelink I System (SR Research, 1996)

[7] Eyelink is a registered trademark of SR Research Ltd.

The Eyelink uses IR-based video tracking, and has a sampling rate of 250 Hz for binocular sampling. The system measures

- horizontal and vertical eye-position and pupil size
- eye rotation angles in fixed, head referenced coordinates
- gaze position in display coordinates
- eye movement events (saccades, fixations and blinks)

The data transit delay, from physical eye movement till eye movement data samples are available to a software application through the Ethernet link, varies between 6-12 ms.

The system provides

- eye position tracking with horizontal range of ±30 degrees, and vertical range of ±20 degrees
- gaze position tracking with horizontal range of ±20 degrees, allowing for moderate head motion
- gaze and eye position resolution of 20 arc sec with a noise level of 0.01º RMS
- gaze position accuracy of 0.5 - 1.0 degrees of visual angle, being primarily limited by fixation accuracy of the subject during calibration

The eye is illuminated with two 940 nm IR LEDs per eye with the maximum irradiance level of $1.2mW/^{cm2}$, well within the safety limits for continuous IR irradiation levels that should be kept below $10mW/cm^2$ (Clarkson, 1989)

Principle of operation

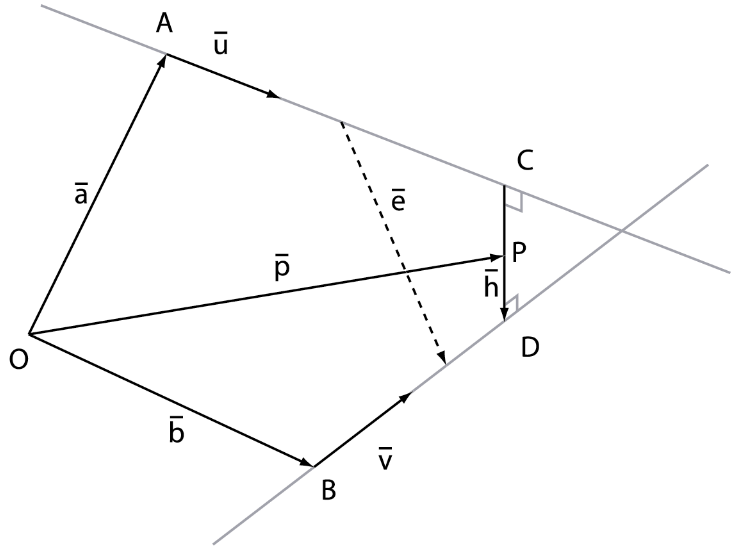
The Eyelink headband is equipped with two miniature, high speed cameras capable of operating on a frame rate of 250 Hz. Each camera has a built in IR illuminator for lighting the camera field of view. The image processing system tracks the position of the pupil in the eye image with a threshold technique, the pupil being darker than its surroundings, when the eye is lit correctly. A third camera tracks four IR LED markers mounted on the visual stimulus display for head motion compensation and true gaze position tracking.

The image processing system processes the pupil and marker position in real time to compute true eye-rotation angles, gaze-position, and gaze-position resolution (effective screen distance). The calculation is based on the relative movement of the pupil center in the eye image, as the eye is rotated. The calibration is usually made with a 9-point calibration sequence, mapping pupil position to relative gaze angles on the calibration screen.

Appendix 3

The near intersection point of two lines in 3D space

The near intersection point is defined as the middle point of the shortest distance vector between the two lines.



Let line 1 be defined by point A and unit vector *u* pointing along the line. Line 2 is defined by point B and unit vector **v** similarly.

The shortest distance vector *h* between the lines is perpendicular to both lines and is given by the vector product

$$\bar{h} = \bar{u} \times \bar{v}$$

Let *e* be a vector connecting lines 1 and 2

$$\bar{e} = \bar{b} + t\,\bar{v} - \bar{a} - s\,\bar{u}$$

If s and t correspond to points of closest approach (C and D) then *e* must be parallel to *h*, and

$$\underbrace{\frac{e_x}{h_x} = \overbrace{\frac{e_y}{h_y}}^{}}_{1^o}\!\!\overbrace{= \frac{e_z}{h_z}}^{2^o}$$

From equation 1° we can solve for s:

$$\frac{tv_x+b_x-a_x-su_x}{h_x}=\frac{tv_y+b_y-a_y-su_y}{h_y}$$

$$(tv_x+b_x-a_x-su_x)h_y=(tv_y+b_y-a_y-su_y)h_x$$

$$tv_x h_y+b_x h_y-a_x h_y-su_x h_y=tv_y h_x+b_y h_x-a_y h_x-su_y h_x$$

$$s(u_y h_x)=(tv_y+b_y-a_y)h_x-(tv_x+b_x-a_x)h_y$$

$$s=\frac{(tv_y+b_y-a_y)h_x-(tv_x+b_x-a_x)h_y}{u_y h_x-u_x h_y}$$

and from equation 2° for t:

$$\frac{tv_y+b_y-a_y-su_y}{h_y}=\frac{tv_z+b_z-a_z-su_z}{h_z}$$

$$tv_y h_z+b_y h_z-a_y h_z-su_y h_z=tv_z h_y+b_z h_y-a_z h_y-su_z h_y$$

$$t(v_y h_z-v_z h_y)+(b_y-a_y)h_z+(a_z-b_z)h_y=s(u_y h_z-u_z h_y)$$

By inserting s into the equation, we finally get

$$t=\frac{((b_z-a_z)h_y+(a_y-b_y)h_z)(u_y h_x-u_x h_y)+((b_x-a_x)h_y-(b_y-a_y)h_x)(u_z h_y-u_y h_z)}{(v_y h_z-v_z h_y)(u_y h_x-u_x h_y)+(v_y h_x-v_x h_y)(u_z h_y-u_y h_z)}$$

or by setting

$k=(u_y h_x-u_x h_y)$, and $m=(u_z h_y-u_y h_z)$

$$t=\frac{((b_z-a_z)h_y+(a_y-b_y)h_z)k+((b_x-a_x)h_y-(b_y-a_y)h_x)m}{(v_y h_z-v_z h_y)k+(v_y h_x-v_x h_y)m}$$

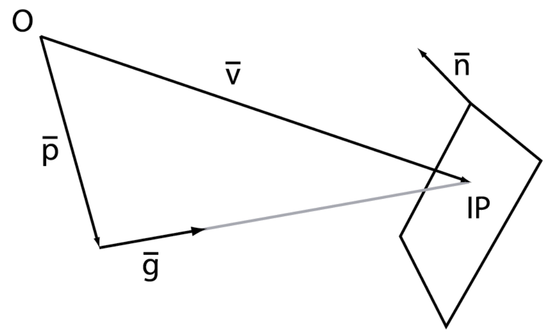
Vector *p* pointing at the near intersection point P is then defined as the middle point of vector *h*

$$\bar{p}=\bar{b}+t\,\bar{v}+\frac{1}{2}\bar{h}$$

The implementation of the algorithm in Delphi code:

```
// function GetVectorNearIntersection( const sp1, sp2, v1, v2
//                                     : TVector;
//                                     ip : PVector ) : Single;
// calculates the near intersection point of two vectors
// params
//   sp1 = starting point of vector 1
//   sp2 = starting point of vector 2
//   v1 = vector 1 'ray'
//   v2 = vector 2 'ray'
//   ip = pointer to a vector variable for holding the
//        intersection point
function GetVectorNearIntersection( const sp1, sp2, v1, v2 : TVector;
                                   ip : PVector ) : Single;
var
  ax, ay, az : Single;
  bx, by, bz : Single;
  cx, cy, cz : Single;
  dx, dy, dz : Single;
  e : TVector;
  ex, ey, ez : Single;
  s, t : Single;
  k, m : Single;
  h : TVector;
  tmp : TVector;
  hx, hy, hz : Single;
begin
  // starting points
  ax := sp1[0]; ay := sp1[1]; az := sp1[2];
  bx := sp2[0]; by := sp2[1]; bz := sp2[2];

  // vectors
  cx := v1[0]; cy := v1[1]; cz := v1[2];
  dx := v2[0]; dy := v2[1]; dz := v2[2];

  // cross product c || d
  h := VectorCrossProduct( v1, v2 );
  hx := h[0]; hy := h[1]; hz := h[2];

  // coefficients from the equation
  k := cy*hx - cx*hy;
  m := cz*hy - cy*hz;

  // ray vector coefficients
  t := ( ((bz-az)*hy + (ay-by)*hz)*k + ((bx-ax)*hy-(by-ay)*hx)*m  ) /
         ( (dy*hz - dz*hy) * k + (dy*hx - dx*hy)*m );

  s := ( (t*dy + by - ay)*hx - (t*dx+bx-ax)*hy ) / ( k );

  // near intersection line
  e := CombineVectors( v2, sp2, t, 1 );
  SubtractVectors( e, sp1, @e );
  SubtractVectors( e, ScaleVector( v1, s ), @e );

  // intersection point ( middle point of near intersection line )
  ip^ := CombineVectors( sp1, v1, 1, s );
  ip^ := CombineVectors( ip^, e, 1, 0.5 );
  Result := VectorLength( e );

end;
```

The intersection point of a vector on a plane



Given a vector

$$\bar{v} = \bar{p} + t\,\bar{g}$$

And a plane

$$Ax + By + Cz + D = 0$$

Where

$\bar{n} = Ax + By + Cz$, and

D is the distance of the plane from origin in the direction defined by $\bar{n}$

Their intersection point IP can be found by solving parameter t from the equation group

$$v_x = p_x + tg_x$$
$$v_y = p_y + tg_y$$
$$v_z = p_z + tg_z$$
$$Ax + By + Cz + D = 0$$

Substituting x, y and z to the plane equation gives

$$A(p_x + tg_x) + B(p_y + tg_y) + C(p_z + tg_z) + D = 0$$

Giving for t

$$t = \frac{-D + Ap_x + Bp_y + Cp_z}{Ag_x + Bg_y + Cg_z}$$, which can be reduced tc

$$t = \frac{-D + (\bar{n} \circ \bar{p})}{\bar{n} \circ \bar{g}}$$

Giving an unambiguous solution for t, supplied that the vector g is not parallel to the plane.

$$\bar{n} \circ \bar{g} \neq 0$$

Now the intersection point can be calculated by substituting t in to the original equation.

The implementation of the algorithm in Delphi code:

```
// function RayCastPlaneIntersect( const rayStart, rayVector :
TVector;
//                                 const planePoint, planeNormal :
TVector;
//                                 intersectPoint : PVector = nil ) :
Boolean;
// calculates the intersection point of a ray vector and a surface
// (  EPSILON : Single = 1e-30; A very small number )
// params
//   rayStart       = starting point of ray vector
//   rayVector      = direction of ray vector
//   planePoint     = any point on the plane
//   planeNormal    = plane normal
//   intersectPoint = a vector variable for holding the intersection
point
// returns
//   true if intersection point found
//   false if not (surface and ray vector are parallel)
function RayCastPlaneIntersect( const rayStart, rayVector : TVector;
                                const planePoint, planeNormal :
TVector;
                                intersectPoint : PVector = nil ) :
Boolean;
var
  sp : TVector;
  t, d : Single;
begin
  d := VectorDotProduct( rayVector, planeNormal );
  Result := ((d>EPSILON) or (d<-EPSILON) );
  if Result and Assigned( intersectPoint ) then
    begin
      SubtractVectors( planePoint, rayStart, @sp ); // check sp & ep
      d := 1 / d;
      t := VectorDotProduct( sp, planeNormal ) * d;
      if ( t>0 ) then
        intersectPoint^ := CombineVectors( rayStart, rayVector, 1, t
)
      else
        Result := False;
    end;
end;
```

Appendix 4

Delphi code of the interface sections of classes TDataProcessor, TEyeTracker, TMagTracker and TPlayback

```
{ -----------------------------------------------------------
  Unit DataProcessor           (c) Kristian Lukander TTL 2003
  Defines and implements the TDataProcessor class
  ---------------------------------------------------------- }
unit DataProcessor;

interface

uses
  Windows, Classes, ExtCtrls, SysUtils, Math,
  EyeTracker, MagTracker, GeneralFunctions, TypeDefinitions,
  Visualizer, DataStorage;

type
  PDataProcessor = ^TDataProcessor;
  TDataProcessor = Class( TObject )
    private
      // trackers
      eyeTracker : TEyeTracker;
      magTracker : TMagTracker;

      // pointers to data sinks ( screen & file )
      disp : PfrmVisualizer;
      storage : TDataStorage;

      // variables for filenames
      binFileName : TFileName;
      ascFileName : TFileName;

      // vectors
      gazeVectorLeft, gazeVectorRight : TVector;
      ipLeft, ipRight : TVector;
      eyePosLeft, eyePosRight : TVector;
      screenPosLeft, screenPosRight : TVector;

      // an event for storing data
      recEvt : TRecEvent;

      // calibration variables
      leftEyePositionRTH, rightEyePositionRTH : TVector;
      gazeRef : Array[0..1] of Array[0..2]of TVector;
      gazeHREF : Array[0..1] of Array[0..2] of TFloatPoint;
      deviceToScreen : TVector;
      screenXVector, screenYVector : TVector;

      // timing test
      qpFrequency : Int64;

      // buffer variables
      scrEvt, headEvt : PPosEvent;
```

```
    smp : PEyeSample;

    // screen geometry                          // 0 -- 1
    screenCorners : Array[0..2] of TVector;    // |
    screenNormal : TVector;                     // 2
    screenCoordSystem : TMatrix;

    // timer
    sampleTimer : TTimer;
  public
    // function pointer for updating the display in diff modes
    displayUpdateFunction : TDisplayUpdateFunction;

    // constructor Create( visForm : PfrmVisualizer );
    // params
    //   visForm : pointer to the visualizer form
    constructor Create( visForm : PfrmVisualizer );
    destructor Destroy;
    procedure Close;

    // procedure SetDisplayUpdateMode( mode : TUpdateMode );
    // params
    //   mode : display update mode (3d, screen)
    procedure SetDisplayUpdateMode( mode : TUpdateMode );
    // display update
    procedure UpdateDisplay3D();
    procedure UpdateDisplayScreenFace();

    // file ops

    // procedure SetDataFileNames( binfn, ascfn : TFileName );
    // params
    //   binfn : the name (w/ full path) of the binary file
    //   ascfn : the name (w/ full path) of the ascii file
    procedure SetDataFileNames( binfn, ascfn : TFileName );
    procedure SaveCurrentDataToFile;
    procedure SaveDataFile;

    // trackers

    // function OpenEyeTrackerConnection : Integer;
    // Opens the connection to the eye tracker
    // returns
    //   1 if successful
    //   0 if not
    function OpenEyeTrackerConnection : Integer;

    // function OpenMagTrackerConnection( Sender : TComponent;
    //                              comPortNum : Integer;
    //                              baudRate : Integer ) : Integer;
    // Opens the connection to the mag tracker
    // params
    //   Sender : Owner of the object
    //   comPortNum : number of COM port tracker is connected to
    //   baudRate : connection speed
    // returns
    //   1 if successful
    //   0 if not
    function OpenMagTrackerConnection( Sender : TComponent;
                                 comPortNum : Integer;
```

```
                              baudRate : Integer ) : Integer;

    // function CalibrateEyeTracker( handle : HWND ) : Integer;
    // Starts the eye tracker calibration procedure
    // params
    //   handle : handle to empty, fullscreen window w/ no borders
    // returns
    //   1 if successful
    //   0 if not
    function CalibrateEyeTracker( handle : HWND ) : Integer;

    // function SetEyePositions( var diffLeft,
    //                           diffRight : Single ) : Integer;
    // Calibrates eye sighting center locations in relation to the
    // head sensor
    // params
    //   diffLeft, diffRight : goodness estimations
    // returns
    //   1 if successful
    //   0 if not
    function SetEyePositions( var diffLeft,
                               diffRight : Single ) : Integer;

    // procedure GetEyePositions( leftPos, rightPos : PVector );
    // returns the positions of the eyes
    // params
    //   pointers to vector variables for the return values
    procedure GetEyePositions( leftPos, rightPos : PVector );

    // function SetReferenceGazeVectors( ) : Integer;
    // A function for setting the two reference vectors
    // returns
    //   1 if successful
    //   0 if not
    function SetReferenceGazeVectors( ) : Integer;

    // procedure GetGazeVectors( leftGaze : PVector = nil;
    //                        rightGaze : PVector = nil ) : Single;
    // gets the current gaze vectors
    // params
    //   leftGaze, rightGaze : pointers to vector variables
    function GetGazeVectors( leftGaze : PVector = nil;
                             rightGaze : PVector = nil ) : Single;

    // user events

    // procedure PutEvent( evt : TRecEvent );
    // Inserts user events to data files
    // params
    //   evt : A TRecEvent filled with appropriate information
    procedure PutEvent( evt : TRecEvent );

    // sample timer onTimer event, called by the timer component
    procedure SampleTimerTimer( Sender : TObject );

    // controlling the timer
    procedure StartRecording;
    procedure StopRecording;

end; // TdataProcessor
```

```
{ --------------------------------------------------------
  Unit MagTracker           (c) Kristian Lukander TTL 2003

  Defines and implements the TMagTracker class, which
  - takes care of communication with Polhemus Fastrak
    using TurboPower AsyncPro ComPort component and
    Polhemus ASCII RS232 interface
  - collects and stores the newest samples to buffers

  -------------------------------------------------------- }
unit MagTracker;

interface

uses
  Classes, TypeDefinitions, GeneralFunctions, AdPort, OoMisc,
SysUtils;

const
  SENSOR_HEAD = 0; SENSOR_SCREEN = 1;
  CONTINUOUS_MODE = 1; STOPPED_MODE = 0;

type
  // pointer types for defining fixed coordinates in relation to the
sensor
  TPointer = (sensorBackCross, pointer );

  PMagTracker = ^TMagTracker;
  TMagTracker = class( TObject )
    private
      // COM port object for communication
      ComPort : TApdComPort;
      // position buffers
      headPos : array[0..1] of TPosEvent;
      screenPos : array[0..1] of TPosEvent;
      // indexes for reading positions
      latestHeadIndex : Byte;
      latestScreenIndex : Byte;
      latestHeadRead : Byte;
      latestScreenRead : Byte;

      // connection open?
      bConnection : Boolean;
      // current mode of measurement as
      // CONTINUOUS_MODE or STOPPED_MODE
      currMode : Integer;

      // pointers to temporary event variables
      headEvt, scrEvt : PPosEvent;

      // pointer coordinates relative to sensor
      pointerTgtVectorStart, pointerTgtVectorEnd : TVector;
      sensorBack : TVector;

      // trigger event called by the COM port object
      procedure ComPortTriggerData( CP : TObject;
                                    TriggerHandle : Word );

      // procedure SetNewData( str : String );
      // procedure for parsing new data from the Polhemus to an event
```

```
    // in the buffer
    // params
    //   str : string to be parsed
    procedure SetNewData( str : String );

  public
    constructor Create();
    destructor Destroy();

    // function Initialize( Sender : TComponent;
    //                      comPortNum : Integer;
    //                      baudRate : Integer ) : Integer;
    //   Creates and initializes the ComPort
    //   Initializes the connection to the Polhemus Fastrak
    //   Sets up communication parameters and data output format
    // params
    //   Sender     = sender component
    //   comPortNum = COM port number
    //   baudRate   = baud rate
    // returns
    //   0 = did not succeed
    //   1 = ok, comport open
    function Initialize( Sender : TComponent; comPortNum : Integer;
                         baudRate : Integer ) : Integer;

    // procedure StartContinuous();
    // starts collecting data samples to the buffer continuously
    // use getnewheaddata() and getnewscreendata() to get the
    // sample from the buffer
    procedure StartContinuous();

    // procedure StopContinuous();
    // stops collectiong data samples from the magtracker
    procedure StopContinuous();

    // procedure GetPointData();
    // gets a single data point to the buffer from the magtracker
    // use getnewheaddata() and getnewscreendata() to get the
    // sample from the buffer
    procedure GetPointData();

    // function GetNewHeadData( data : PPosEvent = nil ) : Boolean;
    // Gets a new datapoint for the head sensor
    // params
    // returns
    function GetNewHeadData( data : PPosEvent = nil ) : Boolean;

    // function GetNewScreenData( data : PPosEvent = nil )
    //                            : Boolean;
    // Gets a new datapoint for the screen sensor
    // params
    // returns
    function GetNewScreenData( data : PPosEvent = nil ) : Boolean;

    // function GetNewHeadData( sensor : Integer;
    //                          data : PPosEvent = nil ) : Boolean;
    // Gets a new datapoint for the given sensor
    // params
    // returns
    function GetNewData( sensor : Integer;
```

```
                      data : PPosEvent = nil ) : Boolean;

    // procedure SendMsg( str : String );
    // sends a string through the serial interface to
    // the magtracker
    // params
    //   str = string to send (with no ending return)
    procedure SendMsg( str : String );

    // procedure CloseConnection();
    // Closes the connection to the tracker
    procedure CloseConnection();

    // function currentMode : Integer;
    // returns the current mode of operation as
    // CONTINUOUS_MODE or STOPPED_MODE
    function currentMode : Integer;

    // procedure GetPointAtEyeVector( sp, ray : PVector );
    // Gets the sighting vector pointing at the eye
    // from the sights. Used in calibrating the eye sighting
    // center locations
    // params
    //   sp : start point
    //   ray : vector direction
    procedure GetPointAtEyeVector( sp, ray : PVector );

    // procedure GetPointerVector( pointer : TPointer;
    //                             vect : PVector );
    // gets a vector pointing at a predefined point in relation
    // to the sensor
    // params
    //   pointer : selection of pointer as TPointer
    //   vect : pointer to a vector variable
    procedure GetPointerVector( pointer : TPointer;
                                vect : PVector);

end; // TmagTracker
```

```
{ --------------------------------------------------------
  Unit EyeTracker           (c) Kristian Lukander TTL 2003

  Defines and implements the TEyeTracker class, which
  - takes care of communication with Eyelink through the
    Eyelink Interface (API)
  - collects and stores the newest samples to buffers

  -------------------------------------------------------- }
unit EyeTracker;

interface

uses
  Windows, Classes, SysUtils, StdCtrls, Math,
  EyeData, EyelinkExptKit2, EyelinkInterface;

const
  E_LEFT = 0; E_RIGHT = 1;  // left and right eye
  N_OF_REF_SAMPLES = 10;    // the number of samples to average, when
defining
                            // reference angles
  F = 15000;                // distance to Eyelink HREF coordinate
system

type
  TEye = ( eyeLeft, eyeRight );
  TEventTypes = ( fixation, saccade, blink );

  PFloatPoint = ^TFloatPoint;
  TFloatPoint = record
    x : Single;
    y : Single;
  end;

  PEyeEvent = ^TEyeEvent;
  TEyeEvent = record
    stime : Cardinal;      // start time
    duration : Cardinal;   // duration
    eType : TEventTypes;   // event type
    eye : TEye;            // eye (left / right)
    gaze : TFloatPoint;    // screen gaze x, y
    angle : TFloatPoint;   // x and y angles
                           // (relative to the reference angle
                           // set with SetReferenceAngle() )
  end;

  PEyeSample = ^TEyeSample;
  TEyeSample = record
    time : Cardinal;                      // 0 = left, 1 = right
    gaze : Array[0..1] of TFloatPoint;    // screen gaze x, y
    angle : Array[0..1] of TFloatPoint;   // x and y angles (relative
                                          // (rel to the ref angle
                                          // set with
                                          // SetReferenceAngle() )
  end;

  PEyeTracker = ^TEyeTracker;
  TEyeTracker = class( TObject )
  private
```

```
    currentEvt : PAllFData;
    currentSmp : PFSample;

    dispinfo : TDisplayInfo;

    myFileName : PChar;
    fileString : String[255];

    referenceAngle : Array[0..1] of TFloatPoint;
    calPoints : Array[0..1] of Array[0..2] of TFloatPoint;

    bEyelinkConnected : Boolean;

  public
    // constructor Create( Owner : TComponent );
    // creates the class,
    // reserves memory for private variables
    constructor Create; //( Owner : TComponent );

    // destructor Destroy; override;
    // destroys the class,
    // frees memory
    destructor Destroy; override;

    // function InitializeConnection : Integer;
    // Initializes the connection to Eyelink
    // returns
    //   1 if success
    //   0 if not
    function InitializeConnection : Integer;

    // procedure CloseConnection;
    // closes the connection to Eyelink
    procedure CloseConnection;

    // function IsConnected : Integer;
    // returns
    //   false  if Eyelink is not connected
    //   true   if Eyelink is Connected
    function IsConnected : Boolean;

    // procedure SetEyelinkParameters( saccadeVelocityThr,
    //                                 saccadeAccelThr : Integer );
    // sets the threshold values, default values should be
    //   velocityThr 35
    //   accelThr    9500
    procedure SetEyelinkParameters( saccadeVelocityThr,
                                    saccadeAccelThr : Integer );

    // function Setup( hWnd : HWND;
    //                 headCompensation : Boolean;
    //                 headSimDist : Integer;
    //                 calFG, calBG : COLORREF;
    //                 calType : Integer ) : Integer;
    // params
    //   hWnd : handle to the calibration window
    //          (empty, full screen with no borders)
    //   headCompensation : true if used, false is simulated
    //   headSimDist : simulated screen distance,
    //                 used only if headCompensation = true
```

```
//   calFG, calBG : foreground and background colors for
//                  the calibration screen
//   calType : type of calibration
//     0 = HV9 (default, 9-point hor+vert)
//     1 = H3  three horizontal targets
//     2 = ...
//   useDataFile : if true, create a .edf -file on Eyelink
//
// returns
//   1 if success
//   0 if not
function Setup( hWnd : HWND;
                headCompensation : Boolean;
                headSimulationDistance : Integer;
                calFG, calBG : COLORREF;
                calType : Integer;
                useDataFile : Boolean ) : Integer;

// procedure Start( fSamples, fEvents, lSamples,
//                  lEvents : Integer );
// starts the recording on eyelink
// params
//   0 = not in use, 1 = used
//    fSamples EL generates samples to a file
//    fEvents  EL generates events to a file
//    lSamples EL passes samples through the link
//    lEvents  EL passes events through the link
procedure Start(fSamples, fEvents, lSamples, lEvents : Integer);

// procedure Stop;
// stops the recording on Eyelink
procedure Stop( handle : HWND );

// function GetFile() : Integer;
// receives the file through the link to a local file
// params
//   fname = the name of the file
// returns
//   file size   if success
//   -1          if error in retrieving file
//   -2          if no filename given and no private filename
function GetFile( fname : PChar ) : Integer;

// function GetNewScreenGazeEvent( evt : PEyeEvent ) : Integer;
// gets the newest screen gaze event
// params
//   evt  pointer to a TEyeEvent
// returns
//   1 if success
//   0 if not
function GetNewScreenGazeEvent( evt : PEyeEvent ) : Integer;

// function GetNewScreenGazeSample( smp : PEyeSample ) : Integer;
// gets the newest screen gaze sample
// params
//   smp  pointer to a TEyeSample
// returns
//   1 if success
//   0 if not
function GetNewScreenGazeSample( smp : PEyeSample ) : Integer;
```

```
// function GetNewHREFAngleEvent( evt : PEyeEvent ) : Integer;
// gets the newest HREF angle event
// the angle is calculated in relation to the reference
// angle set with SetHREFReferenceAngle()
// params
//   evt  pointer to a TEyeEvent
// returns
//   1 if success
//   0 if not
function GetNewHREFAngleEvent( evt : PEyeEvent ) : Integer;

// function GetNewHREFAngleSample( smp : PEyeSample ) : Integer;
// Gets the newest HREF angle sample. The angle is calculated
// in relation to the ref angle set with SetHREFReferenceAngle()
// params
//   smp  pointer to a TEyeSample
// returns
//   1 if success
//   0 if not
function GetNewHREFAngleSample( smp : PEyeSample ) : Integer;

function GetNewHREFSample( smp : PEyeSample ) : Integer;

// function SetHREFReferenceAngle( avg : TEyeSample,
//                                 thr : Single ) : Single;
// sets the ref angle to be used in calculating visual angles.
// params
//   avg = average gaze position ( n samples )
//   thr = threshold for the max deviation during the procedure
// returns
//   the maximum deviation during the procedure, if successful
//   -1 if not
function SetHREFReferenceAngle( avg : PEyeSample;
                                thr : Single ) : Single;

// function SetHREFReferencePoint( pointNr : Integer;
//                                 leftP, rightP : PFloatPoint;
//                                 threshold : Single ) : Single;
// sets the reference points used in coordinate transformations
// params
//   pointNr = calibration point number (0..2)
//   leftP, rightP = pointer to an eye point, if nil, nothing
//   thr = threshold for the max deviation during the procedure
// returns
//   the maximum deviation during the procedure, if successful
//   -1 if not
function SetHREFReferencePoint( pointNr : Integer;
                                leftP, rightP : PFloatPoint;
                                threshold : Single ) : Single;

// procedure TransformHREFRefPointsRelToZeroPoint(
//                               pointNr : Integer;
//                               leftP, rightP :PfloatPoint);
// transforms a reference point in eyelink coordinates to
// a point relative to the zero point
// used in creating the calibration reference vectors
// params
//   pointNr : number of the reference point [0..2]
//   leftP, rightP : binocular point variables (left, right)
```

```
    procedure TransformHREFRefPointsRelToZeroPoint(
                                    pointNr : Integer;
                                    leftP, rightP :PfloatPoint);
  end; //TMagTracker
{ ---------------------------------------------------------
  Unit Playback             (c) Kristian Lukander TTL 2003

  Defines and implements the TPlayback class
  --------------------------------------------------------- }
unit Playback;

interface

uses
  TypeDefinitions, Classes, SysUtils, ExtCtrls, Windows, Visualizer,
Math,
  GeneralFunctions;

type

  PPlayback = ^TPlayback;
  TPlayback = Class( TObject )
  private
    // filestream for reading the file
    fileStream : TFileStream;
    // stream EOF position
    streamEOF : Integer;

    // playback timer
    timer : TTimer;
    startTime : Cardinal;
    startTimeTag : Cardinal;

    // internal timing variable
    qpFreq : Int64;

    // a buffer for currently read data
    buf : TRecEvent;

    // a pointer to the display form
    disp : PfrmVisualizer;

  public
    // function pointer for updating the display in diff modes
    displayUpdateFunction : TDisplayUpdateFunction;

    // constructor Create( visForm : PfrmVisualizer );
    // params
    //   visForm : pointer to the visualizer form
    Constructor Create( visForm : PfrmVisualizer );
    Destructor Destroy;

    // procedure SetDisplayUpdateMode( mode : TUpdateMode );
    // params
    //   mode : display update mode (3d, screen)
    procedure SetDisplayUpdateMode( mode : TUpdateMode );

    // functions for indexting the playback stream

    // function GoToLocation( loc : TStreamLoc;
```

```
  //                        offset : Integer ) : Integer;
  // params
  //   loc : location as (start, end, current)
  //   offset : offset as number of events
  // returns
  //   1 if ok
  //   0 if EOF reached
  function GoToLocation( loc : TStreamLoc;
                         offset : Integer ) : Integer;

  // function GetDataFromFile( evt : PRecEvent ) : Integer;
  // params
  //   evt : event buffer
  // returns
  //   1 if ok
  //   0 if EOF reached
  function GetDataFromFile( evt : PRecEvent ) : Integer;

  // function OpenFile( const filename : PChar ) : Integer;
  // params
  //   filename : The name of the file to open (w/ full path)
  // returns
  //   1 if ok
  //   0 if not
  function OpenFile( const filename : PChar ) : Integer;
  procedure CloseFile();

  // procedure SetPlaybackTimer( interval : Integer );
  // Sets playback timer interval
  // params
  //   interval : time in msecs
  procedure SetPlaybackTimer( interval : Integer );

  // playback timer onTimer event, called by the timer component
  procedure timerTimer( Sender : TObject );

  // function GetMSecTime() : Cardinal;
  // returns the accurate time in msecs from the last computer boot
  // using QueryPerformanceCounter
  function GetMSecTime() : Cardinal;

  // procedures for controlling playback
  procedure Play();
  procedure Pause();
  procedure Stop();

  //display update functions
  procedure UpdateDisplay3D();
  procedure UpdateDisplayScreenFace();

end; //class Tobject
```

Appendix 5
Translated questionnaire

The original questionnaire in Finnish included a description of the planned project, the device, and its principle of operation.

Part one

What kind of features do you think the described device should have from the viewpoint of…

a) The user, that is the person performing the study?

b) The test subject?

c) The measured data?

What kind of future uses would you expect for the device? How could it be developed further?

Part two

Choose the six most important properties of the device, and prioritize them on a scale from 1 to 6 (most important – least important)

1	The strain exerted by the system on the subject should be minimized. - The test session should be made as comfortable as possible	
2	The system should set no restrictions on the subject during measurement. - The user can perform freely during the test	
3	The system should be easy to calibrate - The calibration sequence is as short and easy as possible, both for the test subject and the test director	
4	Calibration accuracy - The calibration of the device is accurate and the measurements produce accurate spatial data	
5	The system should be reliable and robust. - The system does not create error conditions that force repeated trials.	
6	The system should deliver data for all measurable parameters.	
7	The data should be easy to use and analyze - The data is delivered in an easy to use format. No pre-processing should be necessary.	
8	Different experiments should be easy to set up. - For instance changing trial order and timing	
9	Experiment setup should be flexible - The system gives freedom in designing experiment setups, although it may result in added work in the design of all experiments	
10	The user interface of the system should be easy to use	
11	Temporal resolution - The device produces high sampling rates	
12	Spatial resolution - The device produces accurate data	

Part three

What kind of research questions would you prefer to study using quantitative or qualitative measures of data?

Qualitative analysis (subjective evaluation and data scoring from a video, visual tracking of viewing sequences...)	**Quantitative analysis** (numerical analysis, dwell-time analysis, Region-of-interest-analysis...)

Thank you for your answers. Should parts two, or three, of the questionnaire have prompted new ideas, feel free to fill them in to your answers in the first part.