

Eye Movements and Human-Computer Interaction

Päivi Majaranta, Kari-Jouko Räihä, Aulikki Hyrskykari, Oleg Špakov

Contents

| | |
|---|----|
| Contents..... | 1 |
| Abstract..... | 2 |
| 1 Introduction and Learning Objectives..... | 2 |
| 1.1 Human-Computer Interaction..... | 2 |
| 1.2 Gaze-Based Human-Computer Interaction | 4 |
| 2 Historical Annotations..... | 5 |
| 2.1 Early Experimental Eye-Controlled Communication Aids..... | 5 |
| 2.2 First Commercial VOG Systems..... | 6 |
| 2.3 EOG-Systems for Eye-Based Interaction..... | 8 |
| 3 Using the Eyes as an Input Method | 9 |
| 3.1 Eye Tracker as an Input Device | 9 |
| 3.2 Benefits and Challenges of Gaze Input..... | 10 |
| 4 Eye Movement Types and Their Application in HCI | 11 |
| 4.1 Fixations and Dwell Time..... | 11 |
| 4.2 Saccades and Gaze Gestures | 12 |
| 4.3 Smooth Pursuit Eye Movements for Interaction | 14 |
| 4.4 Voluntary Vergence..... | 15 |
| 4.5 Voluntary Pupil Dilation..... | 16 |
| 4.6 Voluntary Blinks and Winks | 16 |
| 4.7 Eye-based Head Gestures..... | 16 |
| 4.8 Summary of Eye Movement Types and Their Application | 17 |
| 5 Design Considerations | 18 |
| 5.1 Controllability and Non-interfering Design..... | 18 |
| 5.2 Feedback and Visibility of the System Status | 20 |
| 5.3 Layout, Color Design, and Typography..... | 22 |
| 5.4 Acceptability, Social Norms, and User Experience..... | 24 |
| 5.5 Customizability and User-Centered Design | 24 |
| 6 Development of Gaze-Responsive Applications and Web Pages | 25 |
| 6.1 Towards Tracker-Independent Applications..... | 25 |
| 6.2 Gaze on the Web..... | 27 |
| 7 Conclusion and Future Directions..... | 28 |
| 8 Key Events and Suggested Readings..... | 29 |
| 9 Questions students should be able to answer | 30 |
| Bibliography..... | 30 |

Abstract

Gaze provides an attractive input channel for human-computer interaction because of its capability to convey the focus of interest. Gaze input allows people with severe disabilities to communicate with eyes alone. The advances in eye tracking technology and its reduced cost make it an increasingly interesting option to be added to the conventional modalities in every day applications. For example, gaze-aware games can enhance the gaming experience by providing timely effects at the right location, knowing exactly where the player is focusing at each moment. However, using gaze both as a viewing organ as well as a control method poses some challenges. In this chapter, we will give an introduction to using gaze as an input method. We will show how to use gaze as an explicit control method and how to exploit it subtly in the background as an additional information channel. We will summarize research on the application of different types of eye movements in interaction and present research-based design guidelines for coping with typical challenges. We will also discuss the role of gaze in multi-modal, pervasive and mobile interfaces and contemplate with ideas for future developments.

Keywords: eye input, gaze interaction, gaze-controlled applications, attentive interfaces

1 Introduction and Learning Objectives

This chapter presents the basics of using gaze in human-computer interaction in real time, either as an explicit input method (intentionally controlled gaze-based applications) or as an additional information channel in the background (gaze-aware, attentive applications). We will discuss how different types of eye movements such as fixations, saccades, smooth pursuit or even vergence can be used in human-computer interaction (HCI). We also give a brief introduction to human-computer interaction and user-centered design. The general design principles for HCI are relevant for any interface, including gaze-based interfaces.

After reading this chapter, the reader should:

- Understand the pros and cons in using the eyes for interaction in real time.
- Recognize common pitfalls involved in using the eyes for interaction and know how to cope with the typical challenges such as inaccuracy and Midas touch.
- Have an idea how human-centered design principles and user interface guidelines could be applied in the context of gaze-based applications.
- Have an overview of the technical design and implementation of gaze-aware applications.

In addition, we give some pointers to further information at the end of the chapter, including links to online examples and advanced information about implementation issues for programming-oriented readers.

1.1 Human-Computer Interaction

Human-computer interaction (HCI) is a field of research that has studied human factors in engineering since the bloom of personal computers in the early 1980s. Nowadays, HCI covers many

subfields that study how people interact with computers and technology in general, incorporating insights from cognitive psychology, linguistics, perception, ergonomics, etc. “Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them” (Hewett et al., 1992). Traditionally, interaction between humans and computers has been done with keyboard and mouse as the input methods and a computer screen as the output device. Today computers are found embedded into everyday appliances and we are constantly surrounded by this technology. The choice of available input and output methods has increased significantly over the years. Vision is one of the modalities that can be used either for perception (perceiving the output) or giving commands to the system (input using gaze). More information about human-computer interaction is available in the free online Encyclopedia of Human-Computer Interaction (Soegaard & Dam, 2012).

General design principles include for example *feedback*, referring both to the immediate response on user’s actions as well as the feedback given on successfully accomplished tasks (Shneiderman, 1998). Other important principles include *visibility* and *consistency* (Norman, 1998). Visibility means the user should be aware of the system’s current status and have some idea what to do next. The term “visibility” should be interpreted broadly here, also covering non-visual ways to make the user aware of the possible functionalities (e.g., spoken instructions). Maintaining consistency means, for instance, that given actions produce the same results, making it possible to learn and predict system behavior within and between systems with similar elements. For more information about general interaction design principles, see e.g. Preece, Rogers, and Sharp (2002).

There are also other key principles that are relevant for this chapter, such as *accessibility*. There should be alternative ways to find information and use a system, to allow people with different needs and abilities to use it. Ensuring that the intended target users can use the system requires a *user-centered approach*. The key principles of *human-centered design* apply also in the design and implementation of gaze-based interfaces. Users can be involved from the very beginning, for example, by applying *participatory design* (Muller & Kuhn, 1993). An iterative design and evaluation process is the key to success. Principles of good usability also

Box 1: Human-Centered Design

Human-centered design (HCD), or *user-centered design* (UCD), considers the needs, wishes and limitations of the user of the product or service during the design process. Six key principles to ensure user-centered design for interactive systems (from ISO standard 9241-210:2010):

- The design is based upon an explicit understanding of users, tasks and environments.
- Users are involved throughout design and development.
- The design is driven and refined by user-centered evaluation.
- The process is iterative.
- The design addresses the whole user experience.
- The design team includes multidisciplinary skills and perspectives.

apply in the design of gaze-based interfaces; for more information, see Heikkilä and Ovaska (2012).

1.2 Gaze-Based Human-Computer Interaction

Gaze is a powerful interaction method. Gaze direction shows to whom a speaker's utterance is targeted at, or what is the object of our interest. Gaze also typically precedes action and gives valuable hints of our intent (Land & Hayhoe, 2001). This makes gaze an attractive alternative input method for human-computer interaction.

Gaze-based HCI is still fairly rare in everyday applications but it has been used in assistive technology for several decades (Majaranta et al., 2012; Majaranta & Bulling, 2014; Istance, 2016). People who are otherwise completely paralyzed may still maintain the control of their eye movements. The control of the eyes is also often preserved in progressive conditions such as amyotrophic lateral sclerosis (ALS) until the late phases of the disease. Access to an eye tracking communication device can significantly improve the quality of life of such users (Caligari, Godi, Guglielmetti, Franchignoni, & Nardone, 2013). Eye tracking also holds the potential for being the easiest and most comfortable alternative means of communication for people with a wide variety of disabilities (Donegan et al., 2009). For example, if manual control or speaking requires considerable effort and induces compulsory movements, a simple "look-to-select" eye tracking application may provide a far more relaxed method for communication and control (Donegan, 2012).

Gaze input holds potential also for mainstream computing. For the general public, gaze could provide an alternative control method, e.g. hands-free pointing and highlighting in presentations (Špakov, Siirtola, Istance, & Rähä, 2016). Gaze can be used to increase immersion in games (Sundstedt, 2011), to control aiming in shooting games (Isokoski, Joos, Špakov, & Martin, 2009), to control the avatar in virtual worlds (Istance, Vickers, & Hyrskykari, 2009), or to provide remote guidance (Akkil, James, Isokoski, & Kangas, 2016).

Information from eye movements can also be used subtly in the background, without disturbing the viewer's natural gaze behavior. The simplest form of *eye-awareness* could be that an application notices the presence of eyes, without the knowledge of the actual gaze direction or target of the gaze. For example, if a cell phone's front camera sees the user's eyes, it can deduce that the user is probably looking at it, even if the cell phone does not know the exact location of the gaze on the screen (Akkil et al., 2015). If information on the gaze direction and scan path (Bischof et al., this volume; Foulsham, this volume) is available, the system "knows" much more about the user's interests and cognitive processes. In gaming, such *gaze-awareness* could, for example, make horror games scarier by letting the player see scary elements in the corner of their eye but make them disappear just before the user can see them sharply (Dechant, Heckner, & Wolff, 2013). The reading experience can be enhanced by offering automatic translation or explanations of difficult terms (Hyrskykari, 2006) or by interactively providing visual and auditory effects to enliven the story as the reading progresses (Biedert, Schwarz, Hees, & Dengel, 2010).

The boundary between explicit gaze input and implicit gaze-aware interfaces is not always clear. People may learn to take advantage of the proactive features of a system and start to use them intentionally to enhance the eye tracking features of an application. For example, if

translations to foreign words are provided automatically based on natural gaze behavior during reading, readers may learn to intentionally prolong their gazes at hard words to get them translated (Hyrskykari, Majaranta, & Rähkä, 2003).

Majaranta and Bulling (2014) introduced a “continuum” of eye tracking applications for human-computer interaction (Figure 1). It represents a simplified categorization of different ways to use eye tracking in HCI; as already discussed, the boundaries of the categories are not clear cut. At one end of the continuum, off-line recordings of unintentional, natural eye movements are used passively for diagnostic applications. For example, eye tracking is a valuable tool for evaluating user interfaces (Thoma and Dodd, this volume). The information gathered from natural gaze behavior can be used for understanding and modeling user behavior. This information can then be used, for example, for activity recognition (Bulling, Ward, Gellersen, & Tröster, 2011), which takes us towards more active application of gaze data. The next step is to exploit the information in real time to infer user intent and predict user behavior. Such gaze-aware, *attentive applications* change their behavior based on the user’s natural gaze behavior and can react more naturally, at the right time (Vertegaal & Shell, 2008). Finally, the eyes can be used for explicit command and control, which requires real-time eye tracking and intentional control of the eyes. The degree of explicit eye control varies, especially when gaze is combined with other modalities (Feit et al., 2017).

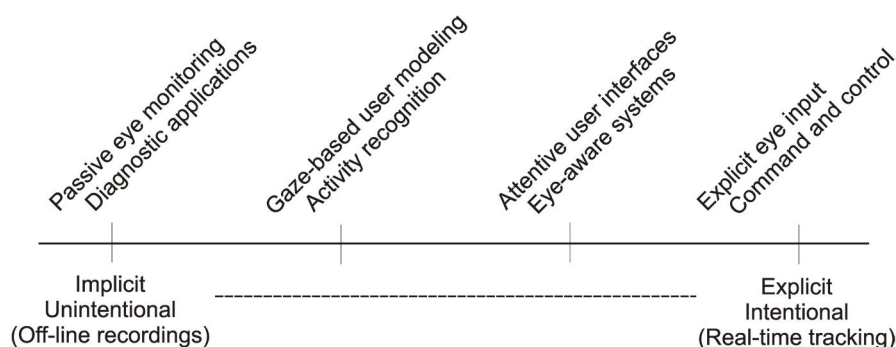


Fig. 1 Continuum of eye tracking applications (slightly modified from Majaranta & Bulling, 2014)

2 Historical Annotations

Eye movements have been used for communication by people with disabilities long before computer-based assistive technology existed (Goossens' & Crain, 1987). The development of computing technology enabled real-time tracking and analysis of eye behavior. This also made it possible to build assistive technology based on eye tracking. In fact, early work on eye movements in HCI initially focused primarily on disabled users (Jacob & Karn, 2003).

2.1 Early Experimental Eye-Controlled Communication Aids

Several eye-controlled electronic communication aids were introduced in the 1970s (Rinard & Rugg, 1976; Foulds & Fincke, 1979). One of the earliest systems, “Eye-Letter-Selector” (Frietman & Tausch, 1978), had phototransistors attached to spectacle frames and a remote electronic control unit with a screen. The system could detect eye movements to left and right,

allowing eyes to be used as simple switches (Ten Kate, Frietman, Stoel, & Willems, 1979). Letters were shown in an electronic box that could give feedback on the current selection. Since only simple left and right commands were available, the system provided a column-row scanning procedure to ease the typing. The user selected the currently highlighted letter by an eye gesture to the right. A glance to the left could be used for cancelling the last action. Using eyes as switches (Grauman, Betke, Lombardi, Gips, & Bradski, 2003) and combining eye movements with scanning are relevant methods also for today's systems (Biswas & Langdon, 2013).

Another early system, EYECOM (Rosen & Durfee, 1978), detected several (rough) gaze directions by the optics mounted in spectacles. The system included eight recognizable directions and an "Enter" in the middle. Eight 3x3 character matrices, including a total of 64 characters, were shown around the central "enter area". A letter was coded by first looking at the direction of the matrix where the correct letter was located and then looking at the direction corresponding with the location of the letter in the matrix (see Figure 2). Feedback for successful selection was given by a brief flash of two LEDs mounted on the spectacles frames. Similar 2-step coding with directional "gaze gestures" have since then been applied in several systems and the technique is still useful for example in low-cost systems that use webcams (Khan, Heynen, & Snuggs, 1999; Bee & André, 2008).

| | | | | | | | | |
|-----------|---|-----|-----------|---|----|-----|---|---|
| A | B | & | . | ? | | J | G | E |
| C | | \$ | ; | | : | % | | F |
| Y | (| SBY | , | ! | | ON |) | K |
| (Group 0) | | | (1) | | | (2) | | |
| N | + | 0 | | | | 2 | * | S |
| < | | 9 | | | | 3 | | > |
| T | - | 8 | | | | 4 | / | R |
| (3) | | | ("ENTER") | | | (4) | | |
| H | [| @ | 7 | 6 | 5 | # |] | Z |
| P | | " | = | | -- | ' | | W |
| O | M | Q | L | I | D | X | V | U |
| (5) | | | (6) | | | (7) | | |

Fig. 2 Illustration of the eight 3x3 character matrix sets (adapted from Rosen & Durfee, 1978). For example, looking first at left and then up would type a character "+" when gaze returns to the center

2.2 First Commercial VOG Systems

In the 1980s, video-based eye tracking or video-oculography (VOG, see Hutton, this volume) sparked off wider interest in eye input, and many systems were developed (Levine, 1981; Yamada & Fukuda, 1987; Hutchinson, White, Martin, Reichert, & Frey, 1989). All of them could be used to enter text and some even included a speech synthesizer to speak out the typed text (Friedman, Kiliany, Dzmura, & Anderson, 1982). To cope with inaccurate tracking, the systems

adopted methods where typing was possible even if only a few objects were shown at a time. For example, the first versions of the ERICA system (originally distributed by Eye Response Technologies Inc., today part of DynaVox Inc.) only included six keys at a time (Hutchinson et al., 1989). The user had to make two selections to enter one letter: first select a group of letters and then the desired letter in the group. Letter and word prediction techniques were developed (Frey, White, & Hutchinson, 1990) to speed up the communication. Another way to speed up communication was to include common sentences in the selection set (Friedman et al., 1982). These techniques are still useful if an accurate or precise enough calibration cannot be obtained (Donegan et al., 2005; Donegan et al., 2009; Donegan, Gill, & Ellis, 2012).

The LC Technologies Eyegaze system (Chapman, 1991) is one of the early commercial systems (Cleveland, 1997) that is still on the market (www.eyegaze.com). It uses direct pointing by gaze, meaning that the user can select the desired target simply by looking directly at it. In addition to facilitating text entry using gaze, the system also implements a set of other useful functions, such as enabling the control of appliances and lights, gameplay, reading electronic documents, and even dialing a telephone by eyes alone.

The potential benefits as well as challenges of using gaze for general human-computer interaction were discussed already in the 1980s. Ware and Mikaelian (1987) showed that pointing by gaze can be very fast – though not as accurate as by using the mouse. Jacob (1990; 1991) discussed the importance of distinguishing normal visual perception from intentional gaze input. Using eye position as a direct substitute for a mouse creates the *Midas touch*¹ problem: “Everywhere you look, another command is activated”. Jacob noted that people’s natural tendency to view the objects of interest longer could be used for implicit input. He compared using “dwell time” (a prolonged fixation) to using a key press for selection, noting that a long dwell time mitigates the speed advantage of eye movements, but dwell can be more convenient for simple information tasks. By careful, iterative interface design, he was able to implement a system where selections were made by gaze using a very short dwell time of 150-250 ms, which actually corresponds with the typical fixation time for visual perception (see Alexander & Martinez-Conde, this volume). Wrong selections could be cancelled easily by looking at the correct object. The problem of Midas touch is a major challenge of gaze input even today. Over the years, a number of solutions have been examined (Velichkovsky, Sprenger, & Unema, 1997); many of them will be discussed later in this chapter.

Another pioneering experiment was conducted by Starker and Bolt (1990), who explored how information from normal viewing could be used to make inferences about the user’s interest and consequently change how the system functioned based on the user’s gaze behavior. Their experimental software implemented a 3D version of the world of *The Little Prince* story (de Saint-Exupéry, 1943). The software analyzed the user’s gaze patterns and changed the order of the narration if the user started to pay more attention to specific objects. All this was done without explicit commands from the user. Gaze is a valuable information channel for non-command (Nielsen, 1993) and proactive interfaces (Hyrskykari et al., 2003).

¹ The phrase “Midas touch” originates from Greek mythology, where King Midas was said to be able to turn everything he touched into gold.

Zhai, Morimoto, and Ihde (1999) demonstrated with their MAGIC (MAnual and Gaze Input Cascaded) pointing how proactive information from normal viewing could be combined with manual control. Gaze was used to warp the mouse cursor near the point the user was looking at (but not exactly, due to tracking inaccuracy; see Figure 3). Manual adjustment with conventional mouse was used to fine-tune the position. This way they got the best of both: the high speed of gaze pointing and the fine accuracy of manual pointing. More information and examples of using gaze as an input method for both command and non-command interfaces can be found in (Hyrskykari, Majaranta, & R  ih  , 2005).

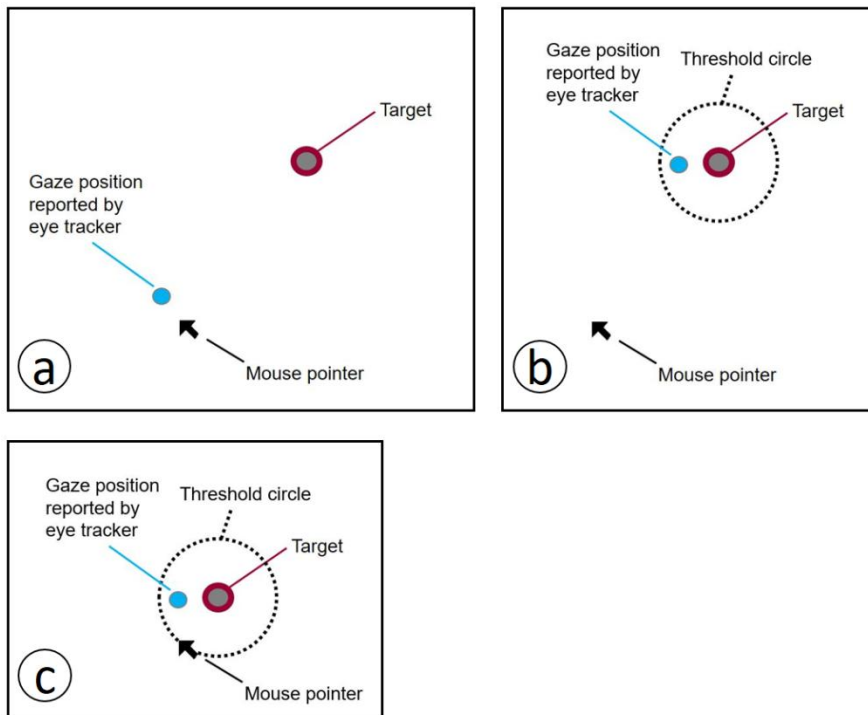


Fig 3 MAGIC pointing. (a) The user wants to move the mouse cursor to the target. Calibration is slightly off. (b) The user looks at the target. Gaze does not land on the target, but close enough, within a predefined threshold. (c) The mouse pointer is automatically warped close to the target, and moving it to the target is then done by using the mouse.

Natural eye movements have also been used implicitly to adapt the level of information shown to the user based on their visual focus. Such foveated rendering of gaze-contingent displays is useful, for example, in virtual reality applications where high resolution details are shown on the user's current area of interest (Duchowski, 1998) or ease visual discomfort of stereo displays (Duchowski et al., 2014). For an overview of related research and application areas, see Reingold et al. (2003) and Duchowski, Cournia, and Murphy (2004). Gaze-contingent displays are located in the implicit end of the continuum of eye tracking applications illustrated in Figure 1. This chapter focuses more on the explicit end, discussing how gaze can be used interactively for explicit control or as an additional channel in eye-aware and attentive interfaces.

2.3 EOG-Systems for Eye-Based Interaction

In addition to video-based systems, also systems based on electro-oculography (EOG, see Hutton, this volume) were used for interactive purposes. For example, Gips, Olivieri, and Tecce (1993) developed an EOG-based system, EagleEyes (www.bc.edu/bc-web/schools/carroll-

school/sites/eagle-eyes.html), which has been and is still being used by people with disabilities for controlling a computer, typing, drawing, playing games etc.

EOG trackers are especially well suited for mobile and outdoor interactions since they require less power (meaning longer battery life) and can cope with ambient light (Bulling, Roggen, & Tröster, 2009; Ishimaru et al., 2015). Since EOG gives information from relative eye rotations, they are especially well suited for steering tasks. For example, they could be used to control a wheelchair (Barea, Boquete, Mazo, & López, 2002) or a robot (Chen & Newman, 2004). Nevertheless, video-based systems have been considered to be better for interactive applications because of their unobtrusiveness and non-invasiveness (Morimoto & Mimica, 2005). However, this might change because the EOG technology has developed so that the electrodes can now be hidden in the nose piece of eye glass frames (Ishimaru et al., 2015) or in ear plugs (Manabe, Fukumoto, & Yagi, 2015).

3 Using the Eyes as an Input Method

Eye tracking technology is introduced in more detail by Hutton (this volume). The same eye tracking hardware that is used for research can also be used for interaction purposes. However, there are some additional requirements.

3.1 Eye Tracker as an Input Device

Using the eye tracker as an input device requires that the data can be transferred to the computer in real time. The gaze data must then be also analyzed in real time, so that the application can react to the user's eye movements. However, the requirements on the quality and details of the data depend on the application. Sometimes exact gaze coordinates are not needed; even information on the presence of eyes may be sufficient, and then also more affordable and less accurate trackers can be adequate.

Both desktop-mounted and head-mounted trackers are used for gaze-based HCI. The exact type and features of the tracker depend on the type of the application and the needs of the user. For example, in laboratory experiments the researcher may require the participant to use a head support or sit very still to preserve best quality in eye movement recordings. The tracker may also be calibrated as often as needed between sessions (see Hutton (this volume) for more information about calibration methods). In real-life scenarios, the user wants to concentrate on the task at hand, without needing to pay attention to the input device (Nielsen, 1993; Jacob, 1995). Non-invasiveness, tolerance to head movements, overall robustness, long-lasting calibration and ease of setup and use are appreciated (Morimoto & Mimica, 2005; Hansen, Hansen, Johansen, & Elvesjö, 2005).

People with disabilities have their own set of requirements for the device as it may be used extensively in their daily activities (Borgestig et al., 2016a). Issues to be considered include portability, maintenance and technical support, good ergonomics, a range of mounting equipment for attaching and positioning the system to be useable from bed or wheelchair, possibility to select which eye to track (e.g., because of divergent squint), customized calibration and ability to adjust the procedure (e.g., because only partial eye movement is retained), tolerance to compulsory body movements or eye nystagmus, ability to connect the system to other assistive systems such as environmental control, and ability to adjust feedback (Holmqvist & Buchholz, 2012; Donegan, Gill, & Ellis, 2012; Donegan et al., 2005).

Our focus is on the design of the software for gaze-based software. Hardware and ergonomics are out of the scope of this chapter.

3.2 Benefits and Challenges of Gaze Input

Using gaze in HCI is attractive, because gaze conveys the *focus of our visual interest*. We look at things we wish to interact with (Land & Furneaux, 1997). Thus, pointing at the objects of interest with gaze is quite natural, easy and effortless. On the other hand, gaze is easily distracted and most of the normal eye movements occur unconsciously. Consequently, persistent voluntary control may feel unnatural and may require some practice, especially for people with severe impairments (Borgestig, Sandqvist, Parsons, Falkmer, & Hemmingsson, 2016).

By following normal gaze behavior, a computer program “knows” more about what is going on in the user’s mind. This has been called the “eye–mind assumption” (Just & Carpenter, 1980). Gaze is *proactive* in nature, giving hints on what we may do next. Such information can be applied in the background, without requiring the user to explicitly change their gaze behavior. However, *inferring relevance* from the gaze is not easy. The problem comes from ambiguous interpretation. As shown already by Yarbus (1967, p. 172), gaze behavior on the same target is very different depending on the task. Methods that apply machine learning to predict interaction intention from gaze behavior have been developed (Bednarik, Vrzakova, & Hradis, 2012; Kandemir & Kaski, 2012). For example, changes in fixation and saccade positions and durations, as well as changes in the pupil dilation, are related to changes in the user’s state. Nevertheless, knowledge of the context, environment, objects, task, as well as additional information from the user’s actions (such as speech or body movement) is highly beneficial in the interpretation (Ajanki et al., 2011; Hyrskykari et al., 2003; Istance & Hyrskykari, 2012).

In both cases, be it voluntary pointing or inferred intention, the big challenge is to avoid false activations, i.e., the Midas touch problem. In voluntary gaze input, unintentional selections can be avoided by adding a separate switch that is used to select the object pointed by gaze (Majaranta, Bates, & Donegan, 2009). However, if the user is only able to move their gaze, other methods like dwell time or gaze gestures are needed (discussed in the next section).

When the eyes point at the focus of the attention, it is called “overt attention”: others can see where the person’s visual focus is targeted at. Since the eyes are always “on” and constantly moving, one should take into account the possibility that the attention may not be focused on the target the eyes happen to point at. Paying attention without moving ones’ eyes is called “covert attention”. For example, a person may pay attention to what is happening in the periphery or outside of their visual field, but does not want to look at it for one reason or the other. In addition, visual attention is actually always slightly ahead of the eye, planning where to move next (Findlay, 2005). Since there is no way to know with absolute certainty which visual information the user is processing at the moment, gaze-aware systems rely on the eye–mind assumption for simplicity. The problem of covert attention is attenuated by informing the user of the system status, by providing feedback on the system’s reactions to the gaze. For a more detailed discussion on covert attention in HCI, see Cantoni and Porta (2014).

Another advantage of gaze is that it is *very fast* (Ware & Mikaelian, 1987). Gaze reaches the object before we move the mouse cursor on the object (Zhai et al., 1999) or before we touch it on the tablet (Weill-Tessier, Turner, & Gellersen, 2016). However, this “look-first-then click” pattern may be more intricate; different people may show different strategies (Smith, Ho, Ark,

& Zhai, 2000) and gaze and cursor are not always aligned (Huang, White, & Buscher, 2012). For example, gaze may already leave for the next target before we click an object, or the user may start to move the mouse towards the known target location already before fixating it (Bieg, Chuang, Fleming, Reiterer, & Bülthoff, 2010). The speed can also be challenging, if the system is supposed to react to the rapid gaze movements fluently, without too much system delay (Kangas et al., 2014a).

A major challenge for gaze input is the *limited accuracy and precision* of the measured point of gaze. Accuracy means how well the measured gaze point matches the real gaze position. Precision, on the other hand, refers to the amount of variation of successive samples within a fixation. There is always some noise in measuring our gaze, even during a fixation (Hansen & Ji, 2010). If the mouse cursor is bound to the eye movement, it will not stay stable on the target. Therefore, many systems apply slight smoothing by averaging consecutive gaze data points. Furthermore, the problem of maintaining good quality gaze data is much more serious outdoors and with mobile systems. Varying light conditions may substantially detract the robustness of the system (Holmqvist et al., 2011).

The active research into eye-based interaction techniques during the last few decades has resulted in a multitude of eye-aware applications and prototypes. Different types of eye movements and their properties can be utilized in these techniques. In the following paragraph we will give examples, categorized by the type of eye movement on which they are based.

4 Eye Movement Types and Their Application in HCI

All of the basic types of eye movements such as fixations, saccades or smooth pursuit can be exploited voluntarily in gaze interaction (Huckauf & Urbina, 2008; Skovsgaard, Rähä, & Tall, 2012; Møllenbach, Hansen, & Lillholm, 2013). We will introduce the basic eye movements and their use for interaction below. However, it should be noted that real interactions are a combination of several types of eye movements.

4.1 Fixations and Dwell Time

Prolonged fixation on a target, referred to as "*dwell time*", is the most used method for making selections by gaze alone. Differentiation from visual inspection is made based on the duration of the dwell time, i.e., selection takes place when the time spent on the target exceeds the predefined threshold. The threshold depends on the task and the user. In any case, the duration should exceed the normal viewing time for the current object in the current context. For example, an experienced gaze typist may adjust the dwell time threshold to be very short to speed up the repetitive task of entering text letter by letter. Dwell times as short as 200-400 ms are possible (Rähä & Ovaska, 2012). Such short dwell times require a fairly accurate tracker (Rähä, 2015). Tasks that require consideration and have higher penalty for false selections, such as problem solving tasks, require longer times, typically 1 second or more (Bednarik, Gowases, & Tukiainen, 2009).

In longitudinal gaze typing experiments (Majaranta, Ahola, & Špakov, 2009; Rähä & Ovaska, 2012), it was found that participants shortened the dwell duration already in the first or second session. The dwell time threshold can also be adjusted automatically, e.g., based on the speed at which the typists leave keys (Špakov & Miniotas, 2004). Sudden or large automatic adjustments can be problematic, because the users learn the rhythm inherent in the dwell

time and may exploit it in their typing (Majaranta, MacKenzie, Aula, & R ih a, 2006). A recent technique (Mott, Williams, Wobbrock, & Morris, 2017) aims at moderate automatic adjustment in order to maintain the rhythm.

Dwell-free methods have gained interest lately (Pedrosa, Pimentel, Wright, & Truong, 2015; Kurauchi, Feng, Joshi, Morimoto, & Betke, 2016). Such fixation-based dwell-free methods assume that the system has enough information about the context and the task to be able to differentiate fixations for perception from intended selections. Dwell-free typing is possible by comparing fixation patterns to known words (Salvucci, 1999). The user fixates on the letters in the correct order but does not need to dwell on them, which can potentially increase the typing speed (Kristensson & Vertanen, 2012). With inaccurate gaze tracking, the method is prone to errors, such as a fixation landing on the neighboring key. Advanced algorithms can be used to handle missing or wrong letters (Liu, Lee, & McKeown, 2016). Dwell-free methods are also used in attentive applications that exploit people’s natural tendency to fixate somewhat longer or more often on objects of interest (R ih a, Hyrskykari, & Majaranta, 2011).

4.2 Saccades and Gaze Gestures

Gaze gestures are eye movements that follow a defined pattern, interpreted as a command (Drewes & Schmidt, 2007). They can be simple, even based on a single stroke (M ollenbach, Hansen, Lillholm, & Gale, 2009) or complex, including several strokes or “legs” (Istance, Hyrskykari, Immonen, Mansikkamaa, & Vickers, 2010). Simple gestures are fast and easy but they may be confused with natural eye movements. Complex gestures are rare in nature but they take time to learn (Huckauf & Urbina, 2008) and can also be harder to execute, which makes them error prone (K opsel, Majaranta, Isokoski, & Huckauf, 2016).

The gestures can be bound to certain locations, such as gestures made by crossing predefined screen areas in a certain order. With such gestures, it is possible to implement quite an extensive set of commands, such as those used for text entry (Wobbrock, Rubinstein, Sawyer, & Duchowski, 2008) or for controlling a computer (Porta & Turina, 2008). Or, the gestures can be off-screen (Isokoski, 2000), which frees the screen for other purposes. Simple gestures that start from the screen and then go off-screen and back by crossing one of the display borders have been used for mode change during gaze-based gaming (Istance, Bates, Hyrskykari, & Vickers, 2008), and controlling a mobile phone (Kangas et al., 2014b) or a smart wrist watch (Akkil et al., 2015). Figure 4 illustrates an on-screen gesture implementation in a game.



Fig. 4 On-screen gesture for gaming. The semi-transparent sectors around the avatar help the user to focus on the gaze-reactive areas. The white numbers and gesture path are for illustration only (Vickers, Istance, & Hyrskykari, 2009).

Location free gestures that are recognized based on changes in the angle of the eye movement are insensitive to spatial accuracy problems and can be done even with uncalibrated trackers (Drewes & Schmidt, 2007). Such relative gestures can easily be tracked with EOG-based trackers, useful in mobile settings (Bulling, Roggen, & Tröster, 2009; Ishimaru et al., 2015).

Rozado, San Agustin, Rodriguez and Varona (2012) defined *gliding gaze gestures* a separate category, meaning gestures where gaze is glided along a predefined path. In their experiment, they found that saccadic gestures outperformed the gliding gestures in user performance and satisfaction. Saccadic gestures were also easier to recognize, with less false positives. Gliding gestures (though not named as such) have been earlier applied by Heikkilä and Rähä (2009) in drawing tasks, where they asked people to draw different shapes by following their outlines by gaze. Curved shapes were especially hard. Drawing a reasonable circle by “gliding” the gaze required many small saccades and was slow.

Continuous pointing gestures is a special sub-category of gaze gestures. What makes them special is the continuous (smooth) eye movement, instead of discrete saccades. Gaze-controlled Dasher (Ward & MacKay, 2002) is a well-known example. With Dasher, one can write with continuous eye movements by pointing at the desired characters. The interface zooms in, by increasing the area of the desired character thus making it easier to select. The character is selected when it crosses the central line, followed by a new set of characters, with the most probable next letters occupying more space thus making them again easier to select (see Figure 5). Continuous gestures are useful in various zooming interfaces and steering tasks, for example, inspecting large images (Adams, Witkowski, & Spence, 2008) or steering an avatar (Vickers, Istance, & Hyrskykari, 2013).

Bulling, & Gellersen, 2013). Location information (or calibration) is not needed, if the mapping is based on the object's movement pattern and speed. With this method, it is possible to select one specific object out of many, by testing which object's movement pattern best matches with the user's eye movements (Figure 6).

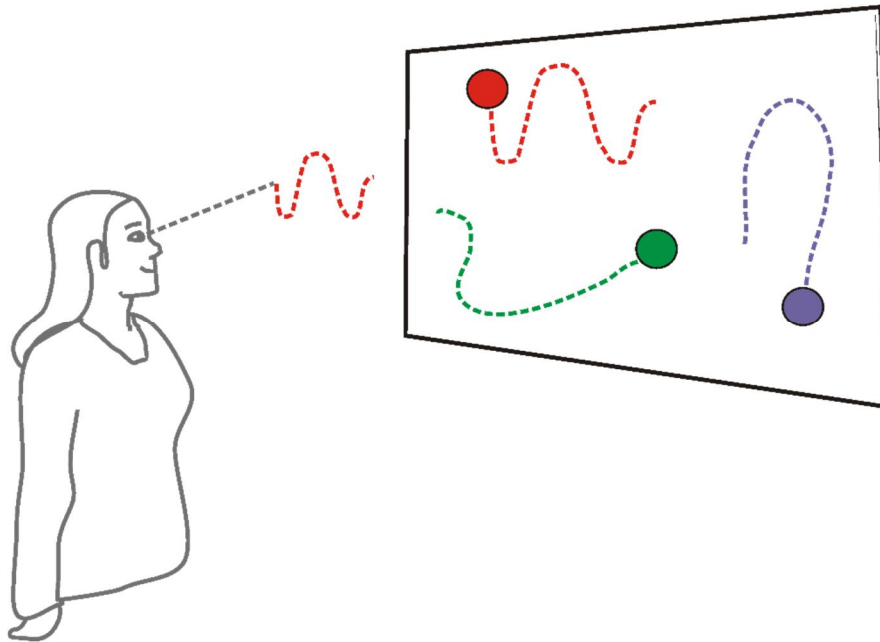


Fig. 6 In smooth pursuit interaction, an object is selected by following it with gaze (re-drawn based on the original by Vidal et al., 2013).

Smooth pursuit interaction is especially suitable for spontaneous interaction in public spaces where random passers-by cannot be expected to go through a calibration phase (Khamis, Alt, & Bulling, 2015; Velloso, Wirth, Weichel, Esteves, & Gellersen, 2016). In addition, it is useful in any interaction situations where calibration is hard to do or hard to maintain, such as for entering pin codes (Cymek et al., 2014) or controlling a smart wrist watch (Esteves, Velloso, Bulling, & Gellersen, 2015). However, smooth pursuit can also be useful for interacting with the conventional user interface widgets (Špakov, Isokoski, Kangas, Akkil, & Majaranta, 2016). Smooth pursuit movement is easy to suppress, thus making it easy to avoid unintentional activations (Schenk, Tiefenbacher, Rigoll, & Dorr, 2016).

4.4 Voluntary Vergence

Convergence (moving eyes inwards to see objects closer) or *divergence* (looking at objects far away) can be used voluntarily for gaze interaction. For example, while focusing on an object of interest, the user moves their gaze behind the object at will, thus using voluntary divergence to select the object of interest (Kudo et al., 2013). Forcing oneself to move the viewpoint forward can be learned but the opposite, moving the viewpoint nearer by simply crossing the eyes, may be easier. Kirst and Bulling (2016) carried out an experiment with voluntary convergence and found it a feasible technique to make selections. All participants were able to use the technique after practice but their experiences varied from “completely awkward” to “surprisingly easy”. Convergence is a fairly robust selection method in 2D interfaces, where it would not happen by accident. One probably would not make exaggerated convergences in 3D interfaces either, but smaller-scaled divergences might occur.

4.5 Voluntary Pupil Dilation

Pupil dilation is one (though unnatural) way to implement eyes only interaction in highly restricted settings. Even if not natural, it is possible to learn to control pupil dilation by physical and psychological self-regulation (Ekman, Poikola, Mäkäräinen, Takala, & Hämäläinen, 2008). However, people vary in their ability to successfully control pupil dilation. Even for those who manage to do it, it is possible only over brief periods (Ehlers, Strauch, Georgi, & Huckauf, 2016). Such “magical” control method could be used, for example, in gaze-based games (Ekman, Poikola, & Mäkäräinen, 2008). However, the technique should be used cautiously as it depends on many external factors, most notably the lighting conditions. For more information on pupil behavior, see Laeng and Alnæs (this volume).

For attentive applications, information from natural pupil size changes may provide useful information of the user’s emotional and cognitive state (Partala & Surakka, 2003).

4.6 Voluntary Blinks and Winks

Similar to using gaze for interaction versus visual inspection, using the blinks for control induces the Midas touch problem. *Intentional blinks* can be separated from natural blinks by using prolonged blinks (Grauman, Betke, Lombardi, Gips, & Bradski, 2003). A blink can be used together with gaze pointing to select the focused item, or as an additional switch that implements a specific function, e.g. a “stop” command (Heikkilä & Rähä, 2012). A blink can also be used as a switch, combined with automatic scanning that moves the focus from item to item (MacKenzie & Ashtiani, 2011). Most commercially available eye-control systems include blinks as one of the optional selection methods (Majaranta et al., 2009b).

In addition to being an active control method, blink detection may also be used in attentive interfaces, for example, to detect alertness (González-Ortega, Antón-Rodríguez, Martínez-Zarzuela, & Díez-Higuera, 2013) or to avoid computer vision syndrome where eyes get exhausted by too much screen time without blinking (Han, Yang, Kim, & Gerla, 2012).

Winks (closing only one eye) should be fairly robust as they require intentional action from the user. On the other hand, producing winks may be hard for some people. If a person can wink, wink with one eye can implement left click and the other be used for right click (Missimer & Betke, 2010).

4.7 Eye-based Head Gestures

Eye-based head gestures detected from data provided by eye trackers may bring a new dimension to the gaze-based interaction. Head gesture detectors utilize eye position in a remote (Špakov & Majaranta, 2012) or head-mounted (Mardanbegi, Hansen & Pederson, 2012) camera view that some eye tracking systems provide along the gaze point. The user views the target (using the eyes as a pointing device) and executes the head gesture while keeping the eyes on the target. A nodding gesture may be an especially convenient method to select the target under visual focus.

4.8 Summary of Eye Movement Types and Their Application

Table 1 summarizes the ways different eye movements have been used in human-computer interaction. Their advantages and challenges are listed together with indications of their usage in gaze-based interaction.

Table 1 Eye movement types in gaze-based interaction.

| Fixations and dwell select | | |
|---|--|---|
| Category | Advantages | Challenges of restrictions |
| Fixed dwell time threshold | Simple to use | Finding the best threshold |
| Adjustable dwell time threshold | Can be adjusted to user and task | Too frequent adjustments can affect the interaction rhythm |
| Dwell-free selections | Potential for fast interaction | Restricted to text entry |
| Saccades and gaze gestures | | |
| Number of strokes | Advantages | Challenges of restrictions |
| Simple gestures (single or few strokes) | Fast | Can be confused with natural eye movements |
| Complex gestures | Don't get confused with natural eye movements | Need learning, can be error prone |
| Continuous pointing | Useful for zooming and steering tasks | Not useful for other tasks |
| Location of strokes | | |
| In a fixed part of the screen | Don't get confused with viewing the screen content | Require shift of attention from target to control |
| Anywhere on the screen | Can be used with a variety of trackers, including EOG-based | Need to be distinguished from natural eye movements |
| Off-screen | Can be used with a variety of display devices | Limited set of commands |
| Less used or newer methods | | |
| Category | Advantages | Challenges of restrictions |
| Smooth pursuit | Does not require calibration; can be used for calibration; suitable for ubiquitous use of eye tracking | Slower than traditional techniques |
| Voluntary vergence | Robust, does not happen by accident | Can be awkward for users |
| Voluntary pupil dilation | | Not possible for every user; only works for brief moments; affected by (changing) lighting conditions |
| Voluntary blinks and winks | Can be easily separated from natural blinks; can be used together with the traditional methods | Can only work as switches |
| Eye-based head gestures | Simple and intuitive selection method; allows keeping the gaze on the target while interacting | Requires ability to move the head (not possible for paralyzed users) |

5 Design Considerations

Above, we saw how different eye movement types can be exploited in human-computer interaction. Here we will discuss some design principles based on the lessons learnt in the research articles reviewed in this chapter.

General principles for good usability can also be applied for gaze-based interfaces. For example, when designing and evaluating gaze-based systems, one might consult the widely accepted usability heuristics (see Nielsen, 1994). These usability heuristics substantially affect the system's ease of use and learnability. Some of these principles are discussed below, with notions specific to the gaze input point of view.

5.1 Controllability and Non-interfering Design

Box 2: Usability heuristics (Nielsen, 1994)

- visibility of the system status,
- match between the system and real world (e.g. speaking the user's language),
- user control and freedom, consistency and standards,
- error prevention,
- recognition rather than recall (e.g. minimize memory load by making actions visible),
- flexibility and efficient use,
- aesthetic and minimalist design,
- help users recognize, diagnose, and recover from errors,
- provide help and documentation

Finding a suitable eye movement type for an application affects how easy it is to control, and how well it enables the user to concentrate on the task instead of the interface. Which types of eye movement work best depends on the task, context, and the user. Dwell select (selection when dwell time exceeds a threshold) is well suited for desktop applications that rely on point and click interactions. However, dwell select is perhaps not the best option for public displays that are used by visitors who are not willing to go through a calibration procedure to get the pointing accuracy required. In such a context, a more feasible option could be the one applied in GazeHorizon (Zhang, Chong, Müller, Bulling, & Gellersen, 2015), which maps uncalibrated horizontal eye movements to sideways scrolling. Keeping the gaze at the center does nothing, allowing the user to look at the object in the center without a risk of Midas touch. Figure 7 illustrates the idea.



Fig 7 GazeHorizon (Zhang et al., 2015) makes any content looked at by the user drift to the center. A hint ("Look here!") is given on both sides of the screen to attract the user to glance on the side. Photo courtesy of Yanxia Zhang, 2015 (c) Yanxia Zhang

Some diseases may restrict the available eye movement types. Involuntary nystagmus may prevent focusing on small targets or inability to move eyes horizontally may restrict the interaction to vertical eye movements alone. According to Donegan (2012), even the most complex cases can be solved and a functional eye control method can be found with persistent user-centered design and iteration.

Perhaps the most important part of being in control of a system based on eye movements is the ability to cope with the Midas touch problem. In gaze-controlled systems, using a combination of gaze pointing and a separate switch for selection is a good solution (Kumar, Paepcke, & Winograd, 2007; Stellmach & Dachselt, 2012). The switch can be anything, whatever is available or preferred by the user: speech, foot button, puff control, frown, smile, blink, nod, etc.

Alternatively, the system can exploit selection methods with multiple steps. A gaze-reactive button could have two separate areas, one for the command and a separate selection area next to it. The selection area could also be rolled out only after a fixation on the object area is detected, as in NeoVisus (Tall, 2008). The idea of separate areas for viewing and selection is also exploited in the context switching method (Morimoto & Amir, 2010). The interface is replicated, for example, so that the keyboard is duplicated. After the user finds the desired key, it can be selected by looking at the same key on the duplicated screen, i.e. by switching context. This enables viewing the whole screen without fear of the Midas touch problem but requires a duplicate amount of screen space.

Gaze gestures with specific patterns can be used to make selections. Gestures can be used independently or combined with dwelling. Istance et al. (2008) experimented with gaze-alone

control of virtual worlds and games, by combining gaze pointing with quick glances outside of the borders of the screen for changing the mode. When the locomotion mode is activated, an avatar on the screen walks to the direction pointed by gaze. When the mouse action mode is activated, the avatar stays in place and the gaze can be used for selecting objects. These and other methods for selection are discussed in more detail by Skovsgaard et al. (2012) in their chapter on computer control by gaze.

One important issue to consider with eye control is the possibility to turn it temporarily off. The user may wish to visually investigate the screen without the fear of false activations. For people with disabilities, it is important that the eye control can then also be turned on again, without a need to call for an assistant – which may be impossible if the user relies on gaze alone. In addition, ability to recalibrate easily is desired. Ideas of how these can be implemented can be found in the report by Donegan et al. (2006).

Since eye movements are partly unconscious and easily distracted, and also because the technology is not always totally robust, an easy way to cancel any gaze-initiated actions is crucial for good usability. Some systems have been implemented so that the system's reaction to gaze is immediate and there is no big penalty for wrong eye movements. For example, in Dasher (Ward & MacKay, 2002) cancelling is part of the continuous control method: selection is done by pointing at the letters on the right side of the screen and cancelling by pointing at the left side of the screen.

In attentive interfaces, poorly implemented proactivity and unwanted assistance may substantially distract and annoy the user. In addition to spatial accuracy (reacting on the correct target), also temporal accuracy is important. The system should recognize when the user needs help. In developing the iDict reading assistant, Hyrskykari, Majaranta, Aaltonen and Riih  (2000) combined knowledge from eye movements with lexical and syntactic analysis. The frequency of the words and word length were used together with time spent on reading to decide if translation was needed. Thus, rare words got translated faster than common words. With grammatical knowledge of the text the system was also able to provide better quality assistance: it could, for instance, translate verbs and nouns correctly.

Wrong interpretations are bound to happen sometimes. The negative effect of the false alarm can be diminished by careful design of the interface. The gaze-evoked action can perhaps be implemented so that the user can ignore it, instead of being required to stop their task and forced to react. For example, Hyrskykari et al. (2003) provided automatic translations to difficult words in a subtle manner that allowed users to continue reading, sometimes even without noticing that a translation had been given.

5.2 Feedback and Visibility of the System Status

Since the eyes are primarily used for perception, the system should clearly inform the user of the system status and the system's response to the eye movements. Proper feedback not only helps in preventing the Midas touch problem but it can also significantly improve user performance and satisfaction (Majaranta, MacKenzie, Aula, & Riih , 2006).

First, the user should know if the system is following the user's gaze correctly. This cannot be achieved if the system does not see the user's gaze in the first place. To help the user in finding

an optimal position for tracking of the eyes, a desktop application could visualize good positioning during calibration (Donegan et al., 2006), or a gaze-aware public screen could give hints on where the passer-by should stand for the remote camera to see their eyes (Alt, Bulling, Gravanis, & Buschek, 2015).

Without any feedback on the focused object, the user would not know if the gaze tracker has correctly recognized the desired target. One option could be showing a cursor that is following the user's gaze. That can, however, be distracting, especially if the calibration is a bit off; it may be hard to prevent oneself from automatically glancing at the moving cursor, which would then move the cursor further as the calibration is off (Jacob, 1991). Thus, in some cases it is better to, for example, highlight the target itself. On the other hand, recent research (Graupner & Pannasch, 2014) has shown that the cursor movement may not actually be as distracting as previously thought and showing the cursor may even improve interaction at least in some cases, such as gaze typing, where seeing a semi-transparent cursor may help the user in understanding why the soft keyboard is not reacting as they expect.

If dwell time is used for commands, the user's gaze on the target only initiates the action. The actual selection happens after the required dwell time threshold has passed. Furthermore, dwell select requires that the user's gaze stays on the target during the whole dwell duration. Thus, the user cannot move their eyes away from the key, for example, to check if the desired letter has already appeared in the text input field. Feedback on the progression of the dwell duration may help the user maintain the focus on the key long enough for it to be selected (Majaranta et al., 2006). It should be noted, however, that it is also possible to implement the dwell time accumulation so that the duration does not reset if gaze data samples land outside of the key (Hansen & Hansen, 2006; Špakov, 2008). This can happen either because of inaccuracies with the eye tracker or if the user unintentionally glances outside of the focused target. Even in this case, it is useful to give feedback on the remaining dwell duration (see an example in Figure 8).

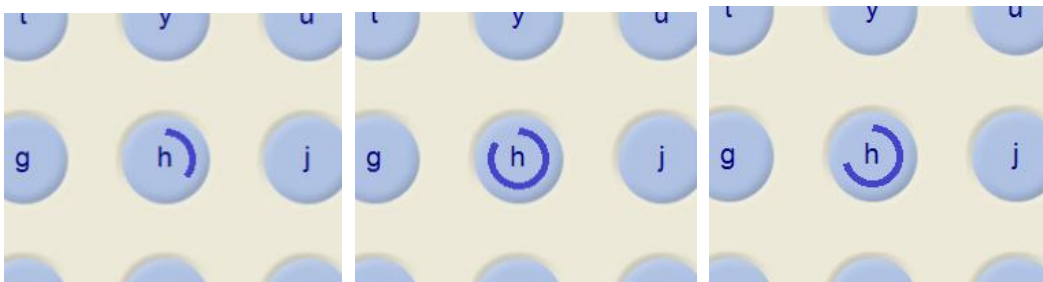


Fig. 8 An animated closing circle indicates the progression of dwell duration in AltTyping (available at <http://www.sis.uta.fi/~csolsp/downloads.php>). If cumulative dwell is used, the circle goes backwards while gaze is out and continues when the gaze returns in to the key.

In addition to feedback on the focused object, clear feedback on the action is important as well. The feedback on selection can be shown on the target itself, especially if the dwelling or smooth pursuit requires the user to look at the target anyways. However, with short dwell times where the user moves the gaze away from the target very fast or when rapid gaze gestures are used, other feedback modalities may be useful. For example, if a person controls a mobile phone by off-screen gaze gestures, haptic feedback on the hand-held phone can inform the user of the successful action (Kangas et al., 2014b). Haptic feedback may also be preferred

in situations where privacy is needed; haptic feedback can only be felt by the person wearing or holding the device. In head-mounted mobile systems, information on the gazed object could be whispered in the user's ear (Eaddy, Blasko, Babcock, & Feiner, 2004). Even with dwell-based systems, users may appreciate additional auditory or haptic feedback since it can be heard or felt even during a blink (Majaranta et al., 2016).

Since eye movements can be very fast, a gaze interface designer should pay attention to the system delay (latency). Eye trackers need to deal with noise in the signal. Even high-end devices cannot avoid the natural jitter of the eyes (Alexander & Martinez-Conde, this volume). The devices may therefore smooth the raw gaze signal, which affects the delay, and may affect the interaction. If the delay between the gaze event and the corresponding feedback is too long, it may confuse the user and hinder performance (Kangas et al., 2014a). When gaze is used for mouse emulation, smoothing the gaze data may help in stabilizing the cursor movement, making it easier to place the mouse cursor over a target and keeping it there. However, if the cursor is used for controlling action games or for fast gaze writing with Dasher, too much smoothing can make the system react too slowly (Donegan et al., 2006). Launching mouse events with gaze gestures may also be problematic because of the fast speed of the saccades. Istance et al. (2010) pointed out that there is a variable delay after the gaze enters a region and before a mouse-over event is generated, which makes the use of Windows events instead of gaze events unreliable.

In attentive applications that are based on natural eye movements one should find a balance between feedback that is not disturbing but that still informs the user of the system status. For example, an application that provides automatic translations depends on good tracking accuracy of the current line of text and the word that is focused. The system could, for example, underline the current line so that the user immediately notices if a wrong line is being tracked, and could easily correct it by pressing the arrow key (Hyrskykari, Majaranta, & Riih , 2003).

5.3 Layout, Color Design, and Typography

The principles of visual usability, aesthetics and minimalistic design are well applicable for gaze-based interfaces. Avoiding unnecessary information is an important part of successful gaze-based user interface design. Limited accuracy and precision mean that the objects should be fairly big to be easily selectable by gaze pointing. How much information can be comfortably fit on the screen depends also, of course, on the screen size. If very small screens are used with eye movements, one should consider using off-screen targets or gestures (see the section of different types of eye movements for examples).

For desktop size screens, it may be useful if the content is located separately, away from the gaze-reactive controls as it allows the user to view the content without needing to worry about accidental activations (Penkar, Lutteroth, & Weber, 2012). Since the accuracy is often better in the center, placing the frequently used objects in the center is useful. Alternatively, easy access to all objects can be facilitated by increasing the size of the objects near the sides of the screen where the accuracy is not as good as in the center (Tuisku, Surakka, Rantanen, Vanhala, & Lekkala, 2013).

One option is to use off-set menus or context sensitive gaze widgets that appear when needed. If a person dwells on a web page link or menu, a gaze-reactive popup menu with fairly

large buttons could appear (Yeoh, Lutteroth, & Weber, 2015). Instead of trying to hit one of the small, closely located menu options, the user can comfortably select one of the big buttons that corresponds to the desired link or menu option (see Figure 9). In addition to such “semantic zooming”, also graphical zooming can be applied (Skovsgaard et al., 2012). In practice, many gaze control systems provide a menu bar for accessing gaze control options, including a magnification option in addition to mouse button functions.

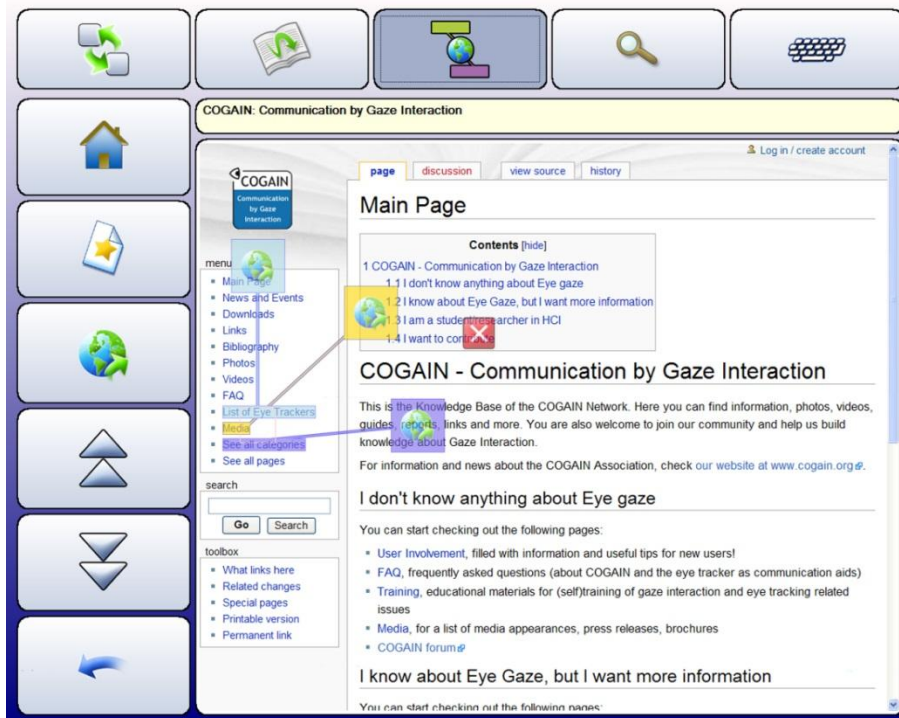


Fig. 9 Screen capture of the “semantic zooming” of the MyTobii browser by Tobii (Note: MyTobii is not available anymore but is replaced with other solutions, see www.tobii.com).

Gaze-based text entry using virtual keyboards is a special case that has been studied a lot. In text entry, not only the size of the keys matters but also the keyboard layout. Direct pointing with dwell select on a full-sized keyboard is more efficient and preferred by people with disabilities compared to a layout with only a few large items combined with word prediction (Pannasch, Malischke, Storch, & Velichkovsky, 2008). The order of the keys (QWERTY vs. alphabetical vs. frequency-based) in the full keyboard layout does not have significant effect on efficiency, though the familiar QWERTY may be preferred (Raupp, 2013). Text entry and on-screen keyboards are also needed with other applications, e.g., filling in web forms. In such cases, the user may want to see at least part of the web page at the same time as entering text. Then either a method based on gaze gestures (Wobbrock, Rubinstein, Sawyer, & Duchowski, 2008) or a scrolling keyboard (Špakov & Majaranta, 2009) that only shows one or a few rows of the keyboard at a time can be useful. Design issues of gaze-based text entry are discussed in more detail by Majaranta (2009).

When designing the layout for a gaze-based interface, it is worth considering how the system can passively help the user focus their gaze. For example, it may be useful to include a visible fixation spot inside a gaze-reactive key (Špakov & Miniotas, 2005). Otherwise, the user may look at the corners of the button instead of focusing on the center. Eye drawing is difficult on a

blank surface. A grid of small dots (preferably with adjustable density) helps the user in focusing their gaze, for example for planning the start and end of a line (Hornof, Cavender, & Hoselton, 2003).

Chitty (2013) proposed design guidelines for eye-controlled technology based on a literature review, surveys, expert interviews and user testing with people with disabilities. Many of the guidelines have already been discussed above (feedback, layout, size). In addition, she proposes guidelines for the color scheme and typography. She found out that, in extended use, dark backgrounds are less straining to the eyes than white backgrounds. Color preferences, however, varied between people. For typography, Chitty recommends using fairly large font sizes and typefaces that are familiar to the user. Also here, the best option is to give the freedom of choice to the user.

5.4 Acceptability, Social Norms, and User Experience

Social acceptability and user experience significantly affect the user's willingness to use the technology. This applies to all users, both those with special needs as well as able bodied people.

Mele and Federici (2012) conducted a literature review of eye tracking technology and found that more emphasis is required on the user experience. In most studies, the focus is on the technology design and users are considered as a measurement object to prove that the technology works. User experience and social norms affect the acceptability and utilization of communication aids. These systems are partly underused because self-consciousness prevents people from using them in public (Hodge, 2007), even though they could be of great benefit. Poor user experience in the beginning may also significantly affect the motivation to use the system. Thus, one should devote enough time during the introductory period for solving problems, learning to use the system and making sure the content is motivating – especially in the case with children (Holmqvist, Thunberg, & Dahlstrand, 2017).

When gaze is used in human-human communication, a screen-based communication system can be obstructive for face-to face-communication (Chitty, 2013). It would be best if the user can maintain eye contact as much as possible also during technology-assisted eye communication.

The ability to maintain gaze contact in human-human interaction is also important for people wearing smart glasses. Akkil et al. (2016) studied user expectations towards everyday gaze interaction with smart glasses. Even though participants were generally positive for the idea of using gaze-aware smart glasses, they were concerned about the social aspects of their use. For example, they worried about looking weird, if they use abnormal eye movements in public. Dwelling on people to get more information about them via the smart glasses was considered invasive; "people are not products". Interaction with objects was seen more positively. If gaze gestures are used, people would want the system to help them remember the gestures and provide clear feedback on their use. Instead of automatically offering help, it was suggested that the glasses could politely ask if the user wants more information about the focused item.

5.5 Customizability and User-Centered Design

Chitty (2013) as well as Donegan et al. (2006, 2009, 2012) emphasize the importance of being able to customize the interface to suit the needs and wishes of the user. Especially considering

people with disabilities, there is no one-size-fits-all solution. There is a great diversity in the needs and user characteristics. One should also keep in mind that many of the diseases are progressive and the condition may vary from day to day. Thus having a one-time setup is not enough.

Successful design requires understanding the users and their needs (Donegan et al., 2009) and expectations (Akkil et al., 2016). It is also important that the actual intended users are involved, otherwise the results may not be valid (Istance, Vickers, & Hyrskykari, 2012). Perhaps the best way to make sure the system really meets the needs of the users is to apply the participatory design method, where the users are part of the design team from the very beginning (Hornof, 2009; Donegan, 2012).

User experience and usability are essential if one wishes that the eye control technology is adapted in everyday use, and not left only as a laboratory experiment or a niche technology only used by those who do not have options.

6 Development of Gaze-Responsive Applications and Web Pages

This section provides a quick introduction and discusses some of the challenges related to the development of gaze-controlled applications to those who wish to implement gaze-based applications of their own. More detailed, up-to-date information and examples can be downloaded from Oleg Špakov's web pages (<http://www.sis.uta.fi/~csolsp/downloads.php>). The examples include source code that can be modified, thus providing a convenient access point to implementation of gaze-responsive applications. The implementation is fairly straightforward but basic programming skills are required.

6.1 Towards Tracker-Independent Applications

As we saw above, during the last decades, tens of prototypes were constructed in the academic institutes, and dozens of commercial systems were released to the market. Most of these systems are supplemented by dedicated software, usually developed by the system manufacturer. Some use a closed protocol for communicating with the hardware, while other systems allow developers to access eye tracking data using a specific API (application programming interface). The variety of APIs and absence of a common (standard) protocol for accessing eye tracking data makes the development and support of gaze-responsive tools and applications more expensive and time-consuming. Several protocols were proposed to standardize the communication between tracking systems and applications (for example, "Open Gaze API" from GazePoint), but none was accepted. This issue remains unsolved even today, when the eye tracking technology steps into its mature phase with cheap and reliable off-the-shelf systems introduced in the mass market.

Until the market players develop a standard protocol, the common method to construct software that supports multiple eye tracking systems is based on using modular architecture. A separate module needs to be developed for each supported system (unless some of them share the same API). While each module utilizes a system-specific API to access eye tracking data, all modules expose the same programming interface to the application core, which then communicates with all eye tracking systems using the same set of methods.

An early example that demonstrates a shift toward a modular architecture is ILAB, a tool for Matlab to analyze recorded gaze data (Gitelman, 2002). This tool does not collect data directly from eye tracking systems; rather, it reads data collected previously and saved in a file. For that ILAB has separate modules to parse gaze data recorded by several eye tracking systems. The parsed data is stored in a single format that is used in all analysis and computations, independently of the system used for collecting it.

OGAMA (Open Gaze And Mouse Analyzer²) is another application with modular architecture for gaze data recording and analysis (Voßkühler, Nordmeier, Kuchinke, & Jacobs, 2008). It was implemented originally to support only one eye tracker but now it supports eye trackers from nine commercial manufacturers and two open-source eye trackers. The application records gaze data directly from an eye tracker, and if the tracker's API allows configuring and calibrating no other tools are needed to work with OGAMA. The application is published as open-source, therefore the modules for data access and communication with eye tracking systems can be reused in software (written in C# programming language).

ETU-Driver (the Eye-Tracking Universal Driver) makes a step forward in facilitating the development of system-independent gaze-responsive applications. It was first developed as a part of iComponent (Špakov, 2008), software similar to OGAMA, and it contained an abstraction layer and modules to work with multiple eye tracking systems transparently from the application core. Later it was extracted into a separate tool to serve as a middleware layer that can be integrated into any software that can embed COM/ActiveX objects (Bates & Špakov, 2006). The tool is distributed with usage examples in several programming languages (C++, C#) that may serve as templates to start developing gaze-responsive applications³.

ETU-Driver supports plugins that extend its capabilities. For example, the "Fixation Detector" plugin consumes gaze points, detects fixations, and streams back fixation start and end events. It may be useful in case an application bases gaze interaction on fixations rather than on gaze points, but the eye tracking system in use does not detect these events internally. Another plugin streams gaze data into the network. There are plugins to transmit gaze data over Bluetooth connection, control mouse cursor by gaze, show notifications when data is not available or of poor quality, etc.

The existing solutions that emulate common gaze tracking API have some shortcomings: a fixed data structure restricts access to system-specific data. The issue may be illustrated in the following example. Normally, an eye tracker outputs a gaze point that consists of 2 or 3 numerical values. The wearable Dikablis eye tracker from Ergoneers (www.ergoneers.com) estimates up to 17 gaze points: one ordinary gaze point that is relative to the scene camera, and up to 16 gaze points relative to 16 detectable markers (a.k.a. 2D barcodes). At the moment, these 16 gaze points cannot be transmitted to the application core using tools like ETU-Driver, thus a solution with more flexible data structure is still missing.

² Available from <http://www.ogama.net/> (July 2017)

³ Available from <http://www.sis.uta.fi/~csolsp/downloads.php>

Most of the eye tracking software and tools have been developed so far for traditional PCs with x86 CPU architecture, primarily running the Microsoft Windows operating system. Exceptions are rare, but evolution of high-level (interpreted) programming languages and rapid expansion of mobile devices with ARM processors already changes the trend. A good example is PyGaze⁴, an open-source eye tracking software that can be used on MS Windows, Linux and Mac OS X operating systems. It is written in Python and currently supports three commercial systems. The tool allows real-time data collection and usage in Python scripts written by users.

6.2 Gaze on the Web

While traditional applications have access to gaze data either using a system API or tools like ETU-Driver, web-based applications running in browsers do not have such access. The share of these applications tends to increase, therefore it is important to have a way to stream gaze points to a browser. Such connectivity is available using WebSockets technology, as first shown by Wassermann et al. (2012), or using eye trackers based on web-camera image analysis directly in the web page code (Papoutsaki et al., 2016). Recently, Kumar et al. (2017) developed a browser based on Chromium that is integrated with an eye tracker and provides access to gaze data from a page code out-of-the-box. With gaze points accessed from JavaScript code, gaze-responsive web pages and applications introduce new opportunities in interaction with online media using gaze.

One perspective direction in bringing gaze to the web is social media. Today's web applications allow people to share their activities, such as collaborative text editing, drawing, and gaming. Sharing the focus of attention in a virtual world may have a strong impact: it is simply faster and more convenient to share the point of attention using gaze rather than the mouse (Qvarfordt, Beymer, & Zhai, 2005). The attention can also be collected through some period of time and shared later (say, while observing static content such as images). Further, gaze data could be accumulated from every observer and, for example, visualized online as heat maps on shared media. This might be especially important in online learning/teaching, which is nowadays rapidly developing.

Dedicated JavaScript libraries available today, such as GazeHook⁵, support the development of gaze-responsive web pages by implicitly executing all necessary basic actions, such as connecting to the web-socket server and converting data received from a WebSocket to JavaScript objects, as well as providing many useful methods to deal with gaze data online. For example, the *gazeTarget.js* library⁶ implements several algorithms to detect user focus on web page elements. It may show gaze pointer and visual feedback when using dwell-time selection algorithms, play "click" sound on object selection, correct gaze point by analyzing small head movements (Špakov, Isokoski, & Majaranta, 2014), etc. The library has several built-in utilities, such as a gaze-controlled keyboard that can be shown when the gaze focuses on a text input

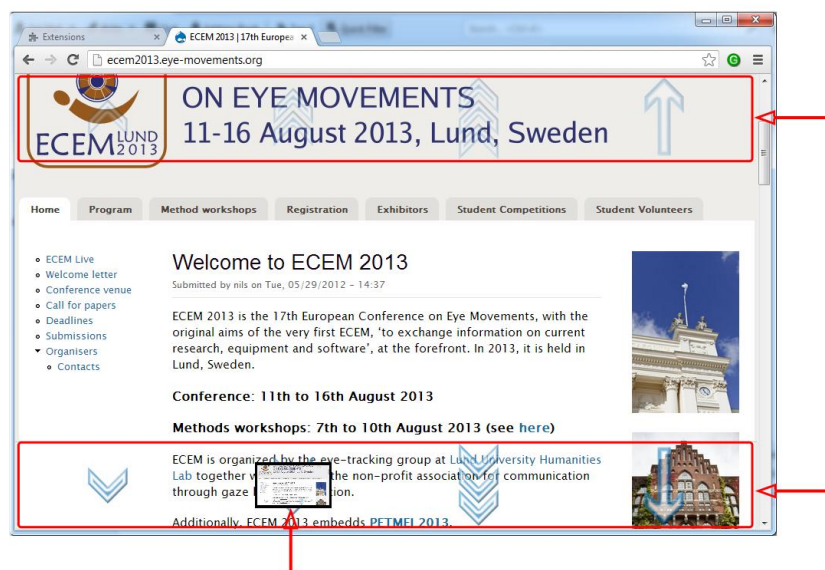
⁴ Available from <http://www.pygaze.org/> (July 2017)

⁵ Available from <http://ux.fiit.stuba.sk/GazeHook> (July 2017)

⁶ Available from <https://github.com/lexasss/etudriver-web> (July 2017)

field, calibration verification tool, automatic page scrolling (similar to the one shown in Figure 10), nodding gesture detector, and free-form head gestures recording and recognition.

Transparent scrolling panels with 4 speed zones shown as icons



Preview of the page shown in the active (focused) zone

Fig. 10 Scrolling panels with semi-transparent areas to indicate varying scrolling speeds and page thumbnail at the focused area for in-place inspection of the current scroll location (© O. Špakov).

Similar functionality is provided by the GazeTK framework (Heil, Renner, & Pfeiffer, 2015): it consists of the component that can handle data from multiple eye trackers and stream it to a browser extension BlickBrowser via WebSocket that enables gaze data access from the page code. The framework implements page auto-scrolling, gaze-enabled main and page menus, smart zoom and link selection, etc. Additionally, it exposes a set of JavaScript functions for developers to add gaze interactivity on custom elements.

7 Conclusion and Future Directions

In this chapter, we have given an introduction to the application of eye movements in human-computer interaction. We have seen how information from natural eye movements can be used subtly in the background as an additional information channel and how to use voluntary gaze input for communication and computer control. We have focused on gaze alone, but one should remember that there are many factors that determine which modality works best for a particular user. Also, many times there is no need to use only gaze. Reviews of using gaze together with other modalities are discussed by Jacob and Stellmach (2016) and extensively by Qvarfordt (2017).

Knowledge of the user, context and task are essential for interpreting the gaze behavior. Non-interfering design should be a guiding principle for attentive interfaces, to avoid distraction and frustration caused by unwanted proactivity. Key issues to consider when designing gaze-based interfaces are the Midas touch problem and feedback. The user should maintain the

feeling of being in control. For example, the system should provide a possibility to cancel erroneous actions. In addition to technical implementation, user experience and usability affect the potential utility and acceptance of the novel technology.

Developments in the technology have significantly increased its availability and usability. Eye trackers have already for a long time been used for daily communication by people with severe disabilities and the number of potential beneficiaries is increasing rapidly as the technology continues to improve (Donegan et al., 2009). The new web technologies along with the ever more accurate web cameras open new possibilities for using eye input with browsers as the platform.

Simple eye-awareness can already been found in commercial mobile phones (Samsung, 2013, p. 30), and some off-the-shelf games already support gaze interaction (Tobii, 2016). At the same time, there is active research on pervasive and mobile eye tracking (Mardanbegi, Jalaliniya, Khamis, & Majaranta, 2016), extending their use to public environments and wearable devices. We hope this chapter serves as an entry point for those who wish to take advantage of this exciting technology in modern interfaces.

8 Key Events and Suggested Readings

The first conference series dedicated to research on eye movements, the European Conference on Eye Movements (ECEM), was launched in 1981 with Rudolf Groner as the driving force. ECEM is organized biannually, and it has grown into the biggest gathering of eye movement researchers. Although the original focus was on psychological studies, applied eye tracking has in the last decade had a growing role in the programme. The abstracts of the papers are published in the *Journal of Eye Movement Research* (<https://www.bop.unibe.ch/jemr>).

The first international conference to explicitly concentrate on eye tracking applications, the ACM Eye Tracking Research and Applications Symposium (ETRA), was established in 2000, with Andrew Duchowski as the General Chair. Since 2000, ETRA has been organized biannually. A large part of the papers related to applied use of eye tracking are also published in general HCI conferences such as the ACM SIGCHI Conference of Human Factors in Computing Systems (CHI). There are also some relevant workshops that have been organized for several years, such as The Scandinavian Workshop on Applied Eye Tracking (SWAET, first organized in 2006) and the International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction (PETMEI, www.petmei.org, first organized in 2011; Bulling, Duchowski, & Majaranta, 2011). In addition, there has been an increasing number of related workshops with varying foci (e.g. on gaze & games and gaze in intelligent user interfaces).

The first book to thoroughly discuss the applied use of gaze, “Eye tracking methodology: Theory and practice”, was published by Andrew Duchowski (2007). The book (now in its third edition) gives an overview of gaze tracking technology and also reviews gaze-based applications. A different point of view is taking in the book by Land and Tatler (2009), who leave the lab and study how eyes behave in real world activities. This is useful knowledge to take into account when designing gaze-based applications. A third key book in this area is “Eye tracking: A comprehensive guide to methods and measures” by Kenneth Holmqvist et al. (2011). This method book is useful to anyone conducting eye tracking experiments, also including interactive applications.

Finally, we wish to mention one more milestone from the history of gaze-based HCI, or *gaze interaction*. The European Commission's funded Network of Excellence on Communication by Gaze Interaction (COGAIN, www.cogain.org), started in 2004. It gathered Europe's leading expertise in eye tracking, involving researchers, eye tracking manufacturers and assistive technology industry, and assistive technology professionals to work together on gaze-based assistive technologies. Users and their representatives were also involved from the very beginning (Bates, Donegan, Istance, Hansen, & Rähkä, 2007). COGAIN eventually involved also members from the USA and Japan. COGAIN worked to better integrate of the eye tracking technology and assistive software, to develop eye tracking standards and to bring down the cost of the systems. Most results achieved during the EU-funded period (2004-2009) are reported in the "COGAIN book" (Majaranta et al., 2012) and others published in the COGAIN website (http://wiki.cogain.org/index.php/COGAIN_Reports). The work continues in the form of an association. It provides a wealth of information on gaze-based assistive technologies, downloadable software, links to open-source eye trackers, and bibliographies on its public web pages, see <http://wiki.cogain.org/>.

For those that need more information on designing interactive software in general, the Interaction Design Foundation provides free electronic books and other online materials related to interaction design, such as The Encyclopedia of Human-Computer Interaction by Mads Soegaard and Rikke Dam (Eds.), see <https://www.interaction-design.org/literature>.

9 Questions students should be able to answer

What is the "Midas touch problem" and what can be done to avoid it?

What kind of challenges are involved in using gaze for mouse emulation?

Why is feedback especially crucial in gaze-based human-computer interaction? (Think, for example, differences in pressing a button with a finger versus using dwell time to select it.)

What are the key design principles to keep in mind when designing visually attentive interfaces?

Bibliography

Adams, N., Witkowski, M., & Spence, R. (2008). The inspection of very large images by eye-gaze control. *Proceedings of the working conference on Advanced visual interfaces (AVI '08)* (pp. 111-118). New York, NY: ACM.

Ajanki, A., Billingham, M., Gamper, H., Järvenpää, T., Kandemir, M., Kaski, S., ... Tossavainen, T. (2011). An augmented reality interface to contextual information. *Virtual Reality*, 15(2), 161-173.

Akkil, D., James, J. M., Isokoski, P., & Kangas, J. (2016). GazeTorch: Enabling gaze awareness in collaborative physical tasks. *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI EA '16)* (pp. 1151-1158). New York, NY: ACM.

- Akkil, D., Kangas, J., Rantala, J., Isokoski, P., Špakov, O., & Raisamo, R. (2015). Glance awareness and gaze interaction in smartwatches. *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI EA '15)* (pp. 1271-1276). New York, NY: ACM.
- Akkil, D., Lucero, A., Kangas, J., Jokela, T., Salmimaa, M., & Raisamo, R. (2016). User expectations of everyday gaze interaction on smartglasses. *Proceedings of the 9th Nordic Conference on Human-Computer Interaction (NordiCHI '16)*. New York, NY: ACM.
- Alt, F., Bulling, A., Gravanis, G., & Buschek, D. (2015). GravitySpot: Guiding users in front of public displays using on-screen visual cues. *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)* (pp. 47-56). New York, NY: ACM.
- Barea, R., Boquete, L., Mazo, M., & López, E. (2002). System for assisted mobility using eye movements based on electrooculography. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 10(4), 209-218.
- Bates, R., Donegan, M., Istance, H. O., Hansen, J. P., & Rähä, K.-J. (2007). Introducing COGAIN: Communication by gaze interaction. *Universal Access in the Information Society*, 6(2), 159-66.
- Bates, R., & Špakov, O. (2006). *Implementation of COGAIN Gaze Tracking Standards*. Communication by Gaze Interaction (COGAIN, IST-2003-511598): Deliverable 2.3. Retrieved from http://wiki.cogain.org/index.php/COGAIN_Reports
- Bednarik, R., Gowases, T., & Tukiainen, M. (2009). Gaze interaction enhances problem solving: Effects of dwell-time based, gaze-augmented, and mouse interaction on problem-solving strategies and user experience. *Journal of Eye Movement Research*, 3(1), 1-10.
- Bednarik, R., Vrzakova, H., & Hradis, M. (2012). What do you want to do next: A novel approach for intent prediction in gaze-based interaction. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)* (pp. 83-90). New York, NY: ACM.
- Bee, N., & André, E. (2008). Writing with your eye: A dwell time free writing system adapted to the nature of human eye gaze. In R. Goebel, J. Siekmann, & W. Wahlster (Eds.), *Perception in Multimodal Dialogue Systems, LNCS 5078* (pp. 111-122). Berlin Heidelberg: Springer.
- Biedert, R. B., Schwarz, S., Hees, J., & Dengel, A. (2010). Text 2.0. *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI EA '10)* (pp. 4003-4008). New York, NY: ACM.
- Bieg, H.-J., Chuang, L. L., Fleming, R. W., Reiterer, H., & Bülthoff, H. H. (2010). Eye and pointer coordination in search and selection tasks. *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '10)* (pp. 89-92). New York, NY: ACM.

- Biswas, P., & Langdon, P. (2013). A new interaction technique involving eye gaze tracker and scanning system. *Proceedings of the 2013 Conference on Eye Tracking South Africa* (pp. 67-70). New York, NY: ACM.
- Borgestig, M., Sandqvist, J., Ahlsten, G., Falkmer, T., Hemmingsson, & Helena. (2016a). Gaze-based assistive technology in daily activities in children with severe physical impairments—An intervention study. *Developmental Neurorehabilitation*. doi:10.3109/17518423.2015.1132281
- Borgestig, M., Sandqvist, J., Parsons, R., Falkmer, T., & Hemmingsson, H. (2016b). Eye gaze performance for children with severe physical impairments using gaze-based assistive technology—A longitudinal study. *Assistive Technology*, 28(2), 93-102.
- Bulling, A., Duchowski, A. T., & Majaranta, P. (2011). PETMEI 2011: The 1st international workshop on pervasive eye tracking and mobile eye-based interaction. *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp '13)* (pp. 627-628). New York, NY: ACM.
- Bulling, A., Roggen, D., & Tröster, G. (2009). Wearable EOG goggles: Eye-based interaction in everyday environments. *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI EA '09)* (pp. 3259-3264). New York, NY: ACM.
- Bulling, A., Ward, J. A., Gellersen, H., & Tröster, G. (2011). Eye movement analysis for activity recognition using electrooculography. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4), 741-753.
- Caligari, M., Godi, M., Guglielmetti, S., Franchignoni, F., & Nardone, A. (2013). Eye tracking communication devices in amyotrophic lateral sclerosis: Impact on disability and quality of life. *Amyotrophic Lateral Sclerosis and Frontotemporal Degeneration*, 14(7-8), 546-552.
- Cantoni, V., & Porta, M. (2014). Eye tracking as a computer input and interaction method. *Proceedings of the 15th International Conference on Computer Systems and Technologies (CompSysTech '14)* (pp. 1-12). New York, NY: ACM.
- Chapman, J. E. (1991). Use of an eye-operated computer system in locked-in syndrome. *Proceedings of the Sixth Annual International Conference on Technology and Persons with Disabilities (CSUN'91)*. Los Angeles, CA.
- Chen, Y., & Newman, W. S. (2004). A human-robot interface based on electrooculography. *Proceedings of 2004 IEEE International Conference on Robotics and Automation (ICRA '04)* (pp. 243-248). IEEE.
- Chitty, N. (2013). *User Fatigue and Eye Controlled Technology*. OCAD University. Toronto, Ontario, Canada: Licensed under the Creative Commons. Retrieved from http://openresearch.ocadu.ca/143/1/Chitty_MRP.pdf
- Cleveland, N. R. (1997). The use of a commercially available eyetracking system by people with severe physical disabilities. *The 9th European Conference on Eye Movements (ECEM '04)*. Ulm, Germany.

- Cymek, D. H., Venjakob, A. C., Ruff, S., Lutz, O. H.-M., Hofmann, S., & Roetting, M. (2014). Entering PIN codes by smooth pursuit eye movements. *Journal of Eye Movement Research*, 7(4), 1.
- De Saint-Exupéry, A. (1943). *The Little Prince*. Reynal & Hitchcock.
- Dechant, M., Heckner, M., & Wolff, C. (2013). Den Schrecken im Blick: Eye Tracking und Survival Horror-Spiele. *Mensch & Computer 2013* (pp. 539-542). Oldenbourg Verlag.
- Donegan, M. (2012). Participatory design: The story of Jayne and other complex cases. In P. Majaranta, H. Aoki, M. Donegan, D. W. Hansen, J. P. Hansen, A. Hyrskykari, & K.-J. Rähä (Eds.), *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies* (pp. 55-61). Hershey, PA: IGI Global.
- Donegan, M. M., Corno, F., Signorile, I., Chió, A., Pasian, V., Vignola, ... Holmqvist, E. (2009). Understanding users and their needs. *Universal Access in the Information Society*, 8(4), 259-275.
- Donegan, M., Gill, L., & Ellis, L. (2012). A client-focused methodology for gaze control assessment, implementation and valuation. In P. Majaranta, H. Aoki, M. Donegan, D. W. Hansen, J. P. Hansen, A. Hyrskykari, & K.-J. Rähä (Eds.), *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies* (pp. 279-286). Hershey, PA: IGI Global.
- Donegan, M., Oosthuizen, L., Bates, R., Daunys, G., Hansen, J., Joos, M., . . . Signorile, I. (2005). *D3.1 User requirements report with observations of difficulties users are experiencing*. Communication by Gaze Interaction (COGAIN), IST-2003-511598: Deliverable 3.1. Retrieved from http://wiki.cogain.org/index.php/COGAIN_Reports
- Donegan, M., Oosthuizen, L., Daunys, G., Istance, H., Bates, R., Signorile, I., . . . Majaranta, P. (2006). *D3.2 Report on features of the different systems and development needs*. Communication by Gaze Interaction (COGAIN), IST-2003-511598: Deliverable 3.2. doi:http://wiki.cogain.org/index.php/COGAIN_Reports
- Drewes, H., & Schmidt, A. (2007). Interacting with the computer using gaze gestures. In *Human-Computer Interaction – INTERACT 2007* (pp. 475-488). Berlin Heidelberg: Springer.
- Duchowski, A. T. (1998). Incorporating the viewer's Point-Of-Regard (POR) in gaze-contingent virtual environments. *The Engineering Reality of Virtual Reality, SPIE Proceedings*, 3295, (pp. 332-343).
- Duchowski, A. T. (2007). *Eye tracking methodology: Theory and practice*. London, UK: Springer.
- Duchowski, A. T., Cournia, N., & Murphy, H. (2004). Gaze-contingent displays: A review. *CyberPsychology & Behavior*, 7(6), 621-634.
- Duchowski, A.T., House, D.H., Gestring, J., Wang, R.I., Krejtz, K., Krejtz, I., ... Bazyluk, B. (2014). Reducing visual discomfort of 3D stereoscopic displays with gaze-contingent depth-of-field. *Proceedings of the acm symposium on applied perception (SAP '14)* (pp. 39-46). New York, NY: ACM.

- Eaddy, M., Blasko, G., Babcock, J., & Feiner, S. (2004). My own private kiosk: Privacy-preserving public displays. *Eighth International Symposium on Wearable Computers (ISWC 2004)* (pp. 132-135). IEEE.
- Ehlers, J., Strauch, C., Georgi, J., & Huckauf, A. (2016). Pupil size changes as an active information channel for biofeedback applications. *Applied Psychophysiology and Biofeedback, 41*(3), 331-339.
- Ekman, I., Poikola, A., & Mäkäräinen, M. (2008a). Invisible Eni: Using gaze and pupil size to control a game. *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI EA '08)* (pp. 3135-3140). New York, NY: ACM.
- Ekman, I., Poikola, A., Mäkäräinen, M., Takala, T., & Hämäläinen, P. (2008b). Voluntary pupil size change as control in eyes only interaction. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '08)* (pp. 115-118). New York, NY: ACM.
- Esteves, A., Velloso, E., Bulling, A., & Gellersen, H. (2015). Orbits: Gaze interaction for smart watches using smooth pursuit eye movements. *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)* (pp. 457-466). New York, NY: ACM.
- Feit, A. M., Williams, S., Toledo, A., Paradiso, A., Kulkarni, H., Kane, S., & Morris, M. R. (2017). Toward everyday gaze input: Accuracy and precision of eye tracking and implications for design. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)* (pp. 1118-1130). New York, NY: ACM.
- Findlay, J. (2005). Covert attention and saccadic eye movements. In L. Itti, G. Rees, & J. K. Tsotsos (Eds.), *Neurobiology of Attention* (Chapter 20, pp. 114-116). Amsterdam: Elsevier.
- Foulds, R. A., & Fincke, R. W. (1979). A computerized line of gaze system for rapid non-vocal individuals. *National Computer Conference 1979 Personal Computing Proceedings*, (p. 81).
- Frey, L., White, K. J., & Hutchinson, T. (1990). Eye-gaze word processing. *IEEE Transactions on Systems, Man, and Cybernetics, 20*, 944-950.
- Friedman, M., Kiliany, G., Dzmura, M., & Anderson, D. (1982). The eyetracker communication system. *Johns Hopkins APL Technical Digest, 3*, 250-252.
- Frietman, E., & Tausch, P. (1978). *An Eye-movement controlled Communication appliance: From Eye-Letter-Selector to Handi-writer*. A Progress and Evaluation Report at request of the Foundation of Multiple Sclerosis.
- Gips, J., Olivieri, P., & Tecce, J. (1993). Direct control of the computer through electrodes placed around the eyes. *Human-computer interaction: Applications and case studies (Proceedings of HCI International '93)* (pp. 630-635). Amsterdam: Elsevier.
- Gitelman, D. (2002). ILAB: A program for post-experimental eye movement analysis. *Behavior Research Methods, Instruments, & Computers, 34*(4), 605-612.

- González-Ortega, D. D.-P., Antón-Rodríguez, M., Martínez-Zarzuela, M., & Díez-Higuera, J. (2013). Real-time vision-based eye state detection for driver alertness monitoring. *Pattern Analysis and Applications*, 16(3), 285-306.
- Goossens', C.; & Crain, S. (1987). Overview of nonelectronic eye-gaze communication techniques. *Augmentative and Alternative Communication*, 3(2), 77-89.
- Grauman, K., Betke, M., Lombardi, J., Gips, J., & Bradski, G. R. (2003). Communication via eye blinks and eyebrow raises: Video-based human-computer interfaces. *Universal Access in the Information Society*, 2(4), 359-373.
- Graupner, S.-T., & Pannasch, S. (2014). Continuous gaze cursor feedback in various tasks: Influence on eye movement behavior, task performance and subjective distraction. *HCI International 2014 - Posters' Extended Abstracts* (pp. 323-329). Cham: Springer.
- Hafed, Z. M., & Clark, J. J. (2002). Microsaccades as an overt measure of covert attention shifts. *Vision Research*, 42(22), 2533-2545.
- Han, S., Yang, S., Kim, J., & Gerla, M. (2012). EyeGuardian: A framework of eye tracking and blink detection for mobile device users. *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications (HotMobile '12)* (Article 6, 6 pages). New York, NY: ACM.
- Hansen, D. W., & Hansen, J. P. (2006). Robustifying eye interaction. *Proceedings of the Computer Vision and Pattern Recognition Workshop (CVPRW '06)*. IEEE.
- Hansen, D. W., & Ji, Q. (2010). In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3), 478-500.
- Hansen, J. P., Hansen, D. W., Johansen, A. S., & Elvesjö, J. (2005). Mainstreaming gaze interaction towards a mass market for the benefit of all. *Proceedings of the 11th International Conference on Human-Computer Interaction (HCI '05)*. Mahwah, NJ: Lawrence Erlbaum.
- Heikkilä, H., & Ovaska, S. (2012). Usability evaluation of gaze interaction. In P. Majaranta, H. Aoki, M. Donegan, D. W. Hansen, J. P. Hansen, A. Hyrskykari, & K.-J. Rähä (Eds.), *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies: Advances in Assistive Technologies* (pp. 255-278). Hershey, PA: IGI Global.
- Heikkilä, H., & Rähä, K.-J. (2009). Speed and accuracy of gaze gestures. *Journal of Eye Movement Research*, 3(2), 1-14.
- Heikkilä, H., & Rähä, K.-J. (2012). Simple gaze gestures and the closure of the eyes as an interaction technique. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)* (pp. 147-154). New York, NY: ACM.
- Heil, D., Renner, P., & Pfeiffer, T. (2015). GazeTK/Blinker: OpenSource framework for gaze-based interaction for the Web. Poster presented at the 18th European Conference on Eye Movements (ECEM '15). Vienna, Austria. More information at <http://gazetk.eyemovementresearch.com/>

- Hewett, T. T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., . . . Verplank, W. (1992). *ACM SIGCHI curricula for human-computer interaction*. New York, NY: ACM.
- Hodge, S. (2007). Why is the potential of augmentative and alternative communication not being realized? Exploring the experiences of people who use communication aids. *Disability & Society, 22*(5), 457-471.
- Holmqvist, E., & Buchholz, M. (2012). A model for gaze control assessments and evaluation. In P. Majaranta, H. Aoki, M. Donegan, D. W. Hansen, J. P. Hansen, A. Hyrskykari, & K.-J. Räihä (Eds.), *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies* (pp. 36-47). Hershey, PA: IGI Global.
- Holmqvist, E., Thunberg, G., & Dahlstrand, M. P. (2017) Gaze-controlled communication technology for children with severe multiple disabilities: Parents and professionals' perception of gains, obstacles, and prerequisites. *Assistive Technology*. DOI: 10.1080/10400435.2017.1307882
- Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., & Van de Weijer, J. (2011). *Eye tracking: A comprehensive guide to methods and measures*. Oxford University Press.
- Hornof, A. J. (2009). Designing with children with severe motor impairments. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)* (pp. 2177-2180). New York, NY: ACM.
- Hornof, A., Cavender, A., & Hoselton, R. (2003). EyeDraw: A system for drawing pictures with eye movements. *Proceedings of the 6th international ACM SIGACCESS conference on Computers and accessibility (Assets '04)* (pp. 86-93). New York, NY: ACM.
- Huang, J., White, R., & Buscher, G. (2012). User see, user point: Gaze and cursor alignment in web search. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)* (pp. 1341-1350). New York, NY: ACM.
- Huckauf, A., & Urbina, M. H. (2008). On object selection in gaze controlled environments. *Journal of Eye Movement Research, 2*(4), 4,1-7.
- Huckauf, A., & Urbina, M. H. (2011). Object selection in gaze controlled systems: What you don't look at is what you get. *ACM Transactions on Applied Perception, 8*(2), Article 13, 14 pages.
- Hutchinson, T., White, K., Martin, W., Reichert, K., & Frey, L. (1989). Human-computer interaction using eye-gaze input. *IEEE Transactions on Systems, Man, and Cybernetics, 19*, 1527-1534.
- Hyrskykari, A. (2006). *Eyes in Attentive Interfaces: Experiences from Creating iDict, a Gaze-Aware Reading Aid*. Dissertation, University of Tampere. Retrieved from <http://urn.fi/urn:isbn:951-44-6643-8>
- Hyrskykari, A., Majaranta, P., Aaltonen, A., & Räihä, K.-J. (2000). Design issues of iDICT: A gaze-assisted translation aid. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '00)* (pp. 9-14). New York, NY: ACM.

- Hyrskykari, A., Majaranta, P., & R ih a, K.-J. (2003). Proactive response to eye movements. In *Human-Computer Interaction – INTERACT 2003* (pp. 129-136). Amsterdam: IOS press.
- Hyrskykari, A., Majaranta, P., & R ih a, K.-J. (2005). From gaze control to attentive interfaces. *Proceedings of Human-Computer Interaction International (HCII 2005), Vol. 2*. Las Vegas, NY.
- Ishimaru, S., Kunze, K., Tanaka, K., Uema, Y., Kise, K., & Inami, M. (2015). Smart eyewear for interaction and activity recognition. *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI EA '15)* (pp. 307-310). New York, NY: ACM.
- Isokoski, P. (2000). Text input methods for eye trackers using off-screen targets. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '00)* (pp. 15-21). New York, NY: ACM.
- Isokoski, P., Joos, M.,  pakov, O., & Martin, B. (2009). Gaze controlled games. *Universal Access in the Information Society*, 8(4), 323-337.
- Istance, H. (2016). *An investigation into gaze-based interaction techniques for people with motor impairments*. Dissertation, Loughborough University. Retrieved from <https://dspace.lboro.ac.uk/2134/23911>
- Istance, H., Bates, R., Hyrskykari, A., & Vickers, S. (2008). Snap clutch, a moded approach to solving the Midas touch problem. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '08)* (pp. 221-228). New York, NY: ACM.
- Istance, H., & Hyrskykari, A. (2012). Gaze-aware systems and attentive applications. In P. Majaranta, H. Aoki, M. Donegan, D. W. Hansen, J. P. Hansen, A. Hyrskykari, & K.-J. R ih a (Eds.), *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies* (pp. 175-195). Hershey, PA: IGI Global.
- Istance, H., Hyrskykari, A., Immonen, L., Mansikkamaa, S., & Vickers, S. (2010). Designing gaze gestures for gaming: An investigation of performance. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '10)* (pp. 323-330). New York, NY: ACM.
- Istance, H., Vickers, S., & Hyrskykari, A. (2009). Gaze-based interaction with massively multiplayer on-line games. *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI EA '09)* (pp. 4381-4386). New York, NY: ACM.
- Istance, H., Vickers, S., & Hyrskykari, A. (2012). The validity of using non-representative users in gaze communication research. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)* (pp. 233-236). New York, NY: ACM.
- Jacob, R. J. (1990). What you look at is what you get: Eye movement-based interaction techniques. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90)* (pp. 11-18). New York, NY: ACM.
- Jacob, R. J. (1991). The use of eye movements in human-computer interaction techniques: What you look at is what you get. *ACM Transactions on Information Systems* 9(2), 152-169.

- Jacob, R. J. (1995). Eye tracking in advanced interface design. In W. Barfield, & T. A. Furness, *Virtual environments and advanced interface design* (pp. 258-288). New York, NY: Oxford University Press, Inc.
- Jacob, R. J., & Karn, K. S. (2003). Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In J. Hyönä, R. Radah, & H. Deubel, *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research* (pp. 573-605). Amsterdam: Elsevier.
- Jacob, R., & Stellmach, S. (2016). What you look at is what you get: Gaze-based user interfaces. *interactions*, 23(5), 62-65.
- Just, M. A., & Carpenter, P. A. (1980). A theory of reading: From eye fixations to comprehension. *Psychological Review*, 87(4), 329-354.
- Kandemir, M., & Kaski, S. (2012). Learning relevance from natural eye movements in pervasive interfaces. *Proceedings of the 14th ACM international conference on Multimodal interaction (ICMI '12)* (pp. 85-92). New York, NY: ACM.
- Kangas, J., Akkil, D., Rantala, J., Isokoski, P., Majaranta, P., & Raisamo, R. (2014a). Delayed haptic feedback to gaze gestures. *Proceedings of EuroHaptics 2014* (pp. 25-31). Berlin Heidelberg: Springer.
- Kangas, J., Akkil, D., Rantala, J., Isokoski, P., Majaranta, P., & Raisamo, R. (2014b). Gaze gestures and haptic feedback in mobile devices. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)* (pp. 435-438). New York, NY: ACM.
- Khamis, M., Alt, F., & Bulling, A. (2015). A field study on spontaneous gaze-based interaction with a public display using pursuits. *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers* (pp. 863-872). New York, NY: ACM.
- Khan, D., Heynen, J., & Snuggs, G. (1999). Eye-controlled computing: The VisionKey experience. *Proceedings of the Fourteenth International Conference on Technology and Persons with Disabilities (CSUN '99)*. Los Angeles, CA. Retrieved from <http://www.eyecan.ca/>
- Kirst, D., & Bulling, A. (2016). On the Verge: Voluntary Convergences for Accurate and Precise Timing of Gaze Input. *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI EA '16)* (pp. 1519-1525). New York, NY: ACM.
- Köpsel, A., Majaranta, P., Isokoski, P., & Huckauf, A. (2016). Effects of auditory, haptic and visual feedback on performing gestures by gaze or by hand. *Behavior & Information Technology*, 35(12), 1044-1062.
- Kristensson, P. O., & Vertanen, K. (2012). The potential of dwell-free eye-typing for fast assistive gaze communication. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)* (pp. 241-244). New York, NY: ACM.
- Kudo, S., Okabe, H., Hachisu, T., Sato, M., Fukushima, S., & Kajimoto, H. (2013). Input method using divergence eye movement. *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI EA '13)* (pp. 1335-1340). New York, NY: ACM.

- Kumar, C., Menges, R., Müller, D., & Staab, S. (2017). Chromium based framework to include gaze interaction in web browser. *Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17 Companion)* (pp. 219-223). International World Wide Web Conferences Steering Committee, Geneva, Switzerland.
- Kumar, M., Paepcke, A., & Winograd, T. (2007). EyePoint: Practical pointing and selection using gaze and keyboard. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)* (pp. 421-430). New York, NY: ACM.
- Kurauchi, A., Feng, W., Joshi, A., Morimoto, C., & Betke, M. (2016). EyeSwipe: Dwell-free text entry using gaze paths. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '16)* (pp. 1952-1956). New York, NY: ACM.
- Land, M. F., & Furneaux, S. (1997). The knowledge base of the oculomotor system. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 352(1358), 1231-1239.
- Land, M., & Hayhoe, M. (2001). In what ways do eye movements contribute to everyday activities? *Vision Research*, 41(25), 3559-3565.
- Laretzaki, G., Plainis, S., Vrettos, I., Chrisoulakis, A., Pallikaris, I., & Bitsios, P. (2011). Threat and trait anxiety affect stability of gaze fixation. *Biological Psychology*, 86(3), 330-336.
- Levine, J. (1981). *An Eye-Controlled Computer*. Yorktown Heights, NY: IBM Thomas J. Watson Research Center.
- Liu, Y., Lee, B.-S., & McKeown, M. J. (2016). Robust eye-based dwell-free typing. *International Journal of Human-Computer Interaction*, 32(9), 682-694.
- Lorenceanu, J. (2012). Cursive writing with smooth pursuit eye movements. *Current Biology*, 22(16), 1506-1509.
- MacKenzie, I. S., & Ashtiani, B. (2011). BlinkWrite: Efficient text entry using eye blinks. *Universal Access in the Information Society*, 10(1), 69-80.
- Majaranta, P. (2009). *Text Entry by Eye Gaze*. Dissertation, University of Tampere. Retrieved from <http://urn.fi/urn:isbn:978-951-44-7787-4>
- Majaranta, P., Ahola, U.-K., & Špakov, O. (2009a). Fast gaze typing with an adjustable dwell time. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)* (pp. 357-360). New York, NY: ACM.
- Majaranta, P., Aoki, H., Donegan, M., Hansen, D. W., Hansen, J. P., Hyrskykari, A., & Rähä, K.-J. (Eds.). (2012). *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies*. Hershey, PA: IGI Global.
- Majaranta, P., Bates, R., & Donegan, M. (2009b). Eye Tracking. In C. Stephanidis (Ed.), *The Universal Access Handbook* (pp. 36:1-20). Boca Raton, FL: CRC Press.

- Majaranta, P., & Bulling, A. (2014). Eye tracking and eye-based human-computer interaction. In S. Fairclough, & K. Gilleade (Eds.), *Advances in Physiological Computing* (pp. 39-65). Springer.
- Majaranta, P., Isokoski, P., Rantala, J., Špakov, O., Akkil, D., Kangas, J., & Raisamo, R. (2016). Haptic feedback in eye typing. *Journal of Eye Movement Research*, 9(1), 1-14.
- Majaranta, P., MacKenzie, I. S., Aula, A., & Rähkä, K. J. (2006). Effects of feedback and dwell time on eye typing speed and accuracy. *Universal Access in the Information Society*, 5(2), 199-208.
- Majaranta, P., & Rähkä, K.-J. (2002). Twenty years of eye typing: Systems and design issues. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '02)* (pp. 15-22). New York, NY: ACM.
- Manabe, H., Fukumoto, M., & Yagi, T. (2015). Conductive rubber electrodes for earphone-based eye gesture input interface. *Personal and Ubiquitous Computing*, 19(1), 143-154.
- Mardanbegi, D., Jalaliniya, S., Khamis, M., & Majaranta, P. (2016). The 6th International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction. *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp '16 Adjunct)*. New York, NY: ACM.
- Mardanbegi, D., Hansen, D. W., & Pederson, T. (2012). Eye-based head gestures. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)* (pp. 139-146). New York, NY: ACM.
- Mele, M. L., & Federici, S. (2012). A psychotechnological review on eye-tracking systems: Towards user experience. *Disability and Rehabilitation: Assistive Technology*, 7(4), 261-281.
- Missimer, E., & Betke, M. (2010). Blink and wink detection for mouse pointer control. *Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments (PETRA '10)* (Article 23, 8 pages). New York, NY: ACM.
- Møllenbach, E., Hansen, J. P., & Lillholm, M. (2013). Eye movements in gaze interaction. *Journal of Eye Movement Research*, 6(2), 1-15.
- Møllenbach, E., Hansen, J. P., Lillholm, M., & Gale, A. G. (2009). Single stroke gaze gestures. *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI EA '09)* (pp. 4555-4560). New York, NY: ACM.
- Morimoto, C. H., & Amir, A. (2010). Context switching for fast key selection in text entry applications. *Proceedings of the 2010 Symposium on Eye Tracking Research & Applications (ETRA '10)* (pp. 271-274). New York, NY: ACM.
- Morimoto, C. H., & Mimica, M. R. (2005). Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding*, 98(1), 4-24.
doi:10.1016/j.cviu.2004.07.010

- Mott, M. E., Williams, S., Wobbrock, J. O., & Morris, M. R. (2017) Improving dwell-based gaze typing with dynamic, cascading dwell times. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '17)* (pp. 2558-2570). New York, NY: ACM.
- Muller, M. J., & Kuhn, S. (1993). Participatory design. *Communications of the ACM*, 36(6), 24-28.
- Nielsen, J. (1993). Noncommand user interfaces. *Communications of the ACM*, 36(4), 83-99.
- Nielsen, J. (1994). Heuristic evaluation. In J. Nielsen, & R. Mack, *Usability Inspection Methods* (pp. 25-62). New York, NY: John Wiley & Sons.
- Norman, D. (1998). *The Design of Everyday Things*. New York, NY: Basic Books.
- Pannasch, S. H., Malischke, S., Storch, A., & Velichkovsky, B. M. (2008). Eye typing in application: A comparison of two systems with ALS patients. *Journal of Eye Movement Research*, 2(8), 6, 1-8.
- Partala, T., & Surakka, V. (2003). Pupil size variation as an indication of affective processing. *International Journal of Human-Computer Studies*, 59(1), 185-198.
- Papoutsaki, A., Sangkloy, P., Laskey, J., Daskalova, N., Huang, J., & Hays, J. (2016). Webgazer: scalable webcam eye tracking using user interactions. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16)* (pp. 3839-3845). AAAI Press.
- Pedrosa, D., Pimentel, M. D., Wright, A., & Truong, K. N. (2015). Filteryedping: Design challenges and user performance of dwell-free eye typing. *ACM Transactions on Accessible Computing*, 6(1), 3 (37 pages).
- Penkar, A. M., Lutteroth, C., & Weber, G. (2012). Designing for the eye: Design parameters for dwell in gaze interaction. *Proceedings of the 24th Australian Computer-Human Interaction Conference (OzCHI '12)* (pp. 479-488). New York, NY: ACM.
- Porta, M., & Turina, M. (2008). Eye-S: A full-screen input modality for pure eye-based communication. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '08)* (pp. 27-34). New York, NY: ACM.
- Preece, J., Rogers, Y., & Sharp, H. (2002). *Interaction Design: Beyond Human-Computer Interaction*. New York, NY: Wiley.
- Qvarfordt, P. (2017). Gaze-informed multimodal interaction. In S. Oviatt, B. Schuller, P. R. Cohen, D. Sonntag, G. Potamianos, & A. Krüger (Eds.), *The Handbook of Multimodal-Multisensor Interfaces* (pp. 365-402). New York, NY: ACM.
- Qvarfordt, P., Beymer, D., & Zhai, S. (2005). RealTourist – A study of augmenting human-human and human-computer dialogue with eye-gaze overlay. In *Human-Computer Interaction – INTERACT 2005* (pp. 767-780). Berlin Heidelberg: Springer.
- Reingold, E. M., Loschky, L. C., McConkie, G. W., & Stampe, D. M. (2003). Gaze-contingent multiresolutional displays: An integrative review. *Human factors*, 45(2), 307-328.

- Räihä, K.-J. (2015). Life in the fast lane: Effect of language and calibration accuracy on the speed of text entry by gaze. In *Human-Computer Interaction – INTERACT 2015* (pp. 402-417). Berlin Heidelberg: Springer.
- Räihä, K.-J., Hyrskykari, A., & Majaranta, P. (2011). Tracking of visual attention and adaptive applications. In C. Roda (Ed.), *Human Attention in Digital Environments* (pp. 166-185). Cambridge University Press.
- Räihä, K.-J., & Ovaska, S. (2012). An exploratory study of eye typing fundamentals: Dwell time, text entry rate, errors, and workload. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)* (pp. 3001-3010). New York, NY: ACM.
- Raupp, S. (2013). *Keyboard layout in eye gaze communication access: Typical vs. ALS*. Communication Sciences and Disorders. East Carolina University. Retrieved from <http://hdl.handle.net/10342/4254>
- Rinard, G., & Rugg, D. (1976). An ocular control device for use by the severely handicapped. *1986 Conference on Systems and Devices for the Disabled*, (pp. 76-79). Boston, MA.
- Rosen, M., & Durfee, W. (1978). Preliminary report on EYECOM, an eye-movement detection and decoding system for non-vocal communication. *Proceedings of the 5th Conference on Systems and Devices for the Disabled*, (pp. 167-171). Houston, TX.
- Rozado, D., San Agustin, J., Rodriguez, F. B., & Varona, B. (2012). Gliding and saccadic gaze gesture recognition in real time. *ACM Transactions on Interactive Intelligent Systems*, 1(2), 10:1-27.
- Salvucci, D. D. (1999). Inferring intent in eye-based interfaces: Tracing eye movements with process models. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)* (pp. 254-261). New York, NY: ACM.
- Samsung. (2013). *User Manual GT-19295*. Samsung Electronics (www.samsung.com).
- Schenk, S., Tiefenbacher, P., Rigoll, G., & Dorr, M. (2016). SPOCK: A smooth pursuit oculomotor control kit. *Extended Abstracts of the Conference on Human Factors in Computing Systems (CHI EA '16)* (pp. 2681-2687). New York, NY: ACM.
- Shneiderman, B. (1998). *Designing the User Interface*. Reading, MA: Addison-Wesley.
- Skovsgaard, H., Räihä, K.-J., & Tall, M. (2012). Computer control by gaze. In P. Majaranta, H. Aoki, M. Donegan, D. W. Hansen, J. P. Hansen, A. Hyrskykari, & K.-J. Räihä (Eds.), *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies* (pp. 78-102). Hershey, PA: IGI Global.
- Smith, B. A., Ho, J., Ark, W., & Zhai, S. (2000). Hand eye coordination patterns in target selection. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '00)* (pp. 117-122). New York, NY: ACM.
- Soegaard, M., & Dam, R. F. (2012). *The Encyclopedia of Human-Computer Interaction*. The Interaction Design Foundation. Retrieved from <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed>

- Špakov, O. (2008). *iComponent – Device-Independent Platform for Analyzing Eye Movement Data and Developing Eye-Based Applications*. Dissertation, University of Tampere. Retrieved from <http://urn.fi/urn:isbn:978-951-44-7321-0>
- Špakov, O. (2011). Comparison of gaze-to-objects mapping algorithms. *Proceeding of The 1st conference on Novel Gaze-Controlled Applications (NGCA'11)* (Article 6, 8 pages). New York, NY: ACM.
- Špakov, O. (2012). Comparison of eye movement filters used in HCI. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)* (pp. 281-284). New York, NY: ACM.
- Špakov, O., Isokoski, P., Kangas, J., Akkil, D., & Majaranta, P. (2016). PursuitAdjuster: An exploration into the design space of smooth pursuit-based widgets. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '16)* (pp. 287-290). New York, NY: ACM.
- Špakov, O., Isokoski, P., & Majaranta, P. (2014). Look and lean: Accurate head-assisted eye pointing. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '14)* (pp. 35-42). New York, NY: ACM.
- Špakov, O., & Majaranta, P. (2009). Scrollable keyboards for casual eye typing. *PsychNology Journal*, 7(2), 159-173.
- Špakov, O., & Majaranta, P. (2012). Enhanced gaze interaction using simple head gestures. *Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp '12)* (pp. 705-710). New York, NY: ACM.
- Špakov, O., & Miniotas, D. (2004). On-line adjustment of dwell time for target selection by gaze. *Proceedings of Nordic Conference on Human-Computer Interaction (NordiCHI '04)* (pp. 203-206). New York, NY: ACM.
- Špakov, O., & Miniotas, D. (2005). EyeChess: A tutorial for endgames with gaze-controlled pieces. *Proceedings of COGAIN 2005* (pp. 16-18). Communication by Gaze Interaction (COGAIN).
- Starker, I., & Bolt, R. A. (1990). A gaze-responsive self-disclosing display. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90)* (pp. 3-10). New York, NY: ACM.
- Stellmach, S., & Dachsel, R. (2012). Look & touch: gaze-supported target acquisition. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)* (pp. 2981-2990). New York, NY: ACM.
- Sundstedt, V. (2011). *Gazing at Games: An Introduction to Eye Tracking Control*. Morgan & Claypool Publishers.
- Tall, M. (2008). *NeoVisus: Gaze Interaction Interface Components*. M.Sc. thesis, Lund University. Retrieved from https://www.researchgate.net/publication/235007493_Neovisus_Gaze_Interaction_Interface_Components

- Tchalenko, J. (2001). Free-eye drawing. *Point: Art and Design Research Journal*, 11, 36-41.
- Ten Kate, J., Frietman, E., Stoel, F., & Willems, W. (1979). Eye-controlled communication aids. *Medical Progress through Technology*, 8(1), 1-21.
- Tobii. (2016, June 28). *Apps*. Retrieved from Tobii tech / experience: <http://www.tobii.com/xperience/apps/>
- Tuisku, O., Surakka, V., Rantanen, V., Vanhala, T., & Lekkala, J. (2013). Text entry by gazing and smiling. *Advances in Human-Computer Interaction*, Article ID 218084, 13 pages.
- Velichkovsky, B., Sprenger, A., & Unema, P. (1997). Towards gaze-mediated interaction: Collecting solutions of the "Midas touch" problem. In *Human-Computer Interaction – INTERACT'97* (pp. 509-516). New York, NY: Springer.
- Velloso, E., Wirth, M., Weichel, C., Esteves, A., & Gellersen, H. (2016). AmbiGaze: Direct control of ambient devices by gaze. *Proceedings of the 2016 ACM Conference on Designing Interactive Systems (DIS '16)* (pp. 812-817). New York, NY: ACM.
- Vertegaal, R., & Shell, J. S. (2008). Attentive user interfaces: The surveillance and sousveillance of gaze-aware objects. *Social Science Information*, 47(3), 275-298.
- Vickers, S., Istance, H., & Hyrskykari, A. (2009). Selecting commands in 3D game environments by gaze. *Proceedings of 5th Conference on Communication by Gaze Interaction (COGAIN)* (pp. 55-59). The Technical University of Denmark, Lyngby, Denmark.
- Vickers, S., Istance, H., & Hyrskykari, A. (2013). Performing locomotion tasks in immersive computer games with an adapted eye-tracking interface. *ACM Transactions on Accessible Computing (TACCESS)*, 5(1), 2:1-33.
- Vidal, M., Bulling, A., & Gellersen, H. (2013). Pursuits: Spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing* (pp. 439-448). New York, NY: ACM.
- Voßkühler, A., Nordmeier, V., Kuchinke, L., & Jacobs, A. (2008). OGAMA – Open Gaze And Mouse Analyzer: Open source software designed to analyze eye and mouse movements in slideshow study designs. *Behavior Research Methods*, 40(4), 1150-1162.
- Ward, D. J., & MacKay, D. J. (2002). Fast hands-free writing by gaze direction. *Nature*, 414, 838.
- Ware, C., & Mikaelian, H. H. (1987). An evaluation of an eye tracker as a device for computer input. *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface (CHI '87)* (pp. 183-188). New York, NY: ACM.
- Weill-Tessier, P., Turner, J., & Gellersen, H. (2016). How do you look at what you touch?: A study of touch interaction and gaze correlation on tablets. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '16)* (pp. 329-330). New York, NY: ACM.

- Wobbrock, J. O., Rubinstein, J., Sawyer, M. W., & Duchowski, A. T. (2008). Longitudinal evaluation of discrete consecutive gaze gestures for text entry. *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '08)* (pp. 11-18). New York, NY: ACM.
- Yamada, M., & Fukuda, T. (1987). Eye word processor (EWP) and peripheral controller for the ALS patient. *IEEE Proceedings A – Physical Science, Measurement and Instrumentation, Management and Education –Reviews*, 134(4), 328-330.
- Yarbus, A. L. (1967). *Eye Movements and Vision*. New York, NY: Plenum Press.
- Yeoh, K. N., Lutteroth, C., & Weber, G. (2015). Eyes and keys: An evaluation of click alternatives combining gaze and keyboard. In *Human-Computer Interaction – INTERACT 2015* (pp. 367-383). Berlin Heidelberg: Springer.
- Zhai, S., Morimoto, C., & Ihde, S. (1999). Manual and gaze input cascaded (MAGIC) pointing. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)* (pp. 246-253). New York, NY: ACM.
- Zhang, Y., Bulling, A., & Gellersen, H. (2013). Sideways: A gaze interface for spontaneous interaction with situated displays. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)* (pp. 851-860). New York, NY: ACM.
- Zhang, Y., Chong, M. K., Müller, J., Bulling, A., & Gellersen, H. (2015). Eye tracking for public displays in the wild. *Personal and Ubiquitous Computing*, 19(5), 967-981.
- Zhang, Y., Müller, J., Chong, M. K., Bulling, A., & Gellersen, H. (2014). GazeHorizon: Enabling passers-by to interact with public displays by gaze. *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14)* (pp. 559-563). New York, NY: ACM.