Gaze Controlled Applications and Optical-See-Through Displays -

General Conditions for Gaze Driven Companion Technologies

von

Mario Humberto Urbina Cazenave

Dissertation eingereicht an der
Fakultät Medien

zum Erreichen des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)

an der
Bauhaus-Universität Weimar

Gutachter:
Prof. Dr. Anke Huckauf
Universität Ulm

Zweitgutachter:
Prof. Dr. Michael Weber
Universität Ulm

Tag der Einreichung: 18. April 2012

Abstract

Gaze based human-computer-interaction has been a research topic for over a quarter century. Since then, the main scenario for gaze interaction has been helping handicapped people to communicate an interact with their environment. With the rapid development of mobile and wearable display technologies, a new application field for gaze interaction has appeared, opening new research questions.

This thesis investigates the feasibility of mobile gaze based interaction, studying deeply the use of pie menus as a generic and robust widget for gaze interaction as well as visual and perceptual issues on head mounted (wearable) optical see-through displays.

It reviews conventional gaze-based selection methods and investigates in detail the use of pie menus for gaze control. It studies and discusses layout issues, selection methods and applications. Results show that pie menus can allocate up to six items in width and multiple depth layers, allowing a fast and accurate navigation through hierarchical levels by using or combining multiple selection methods. Based on these results, several text entry methods based on pie menus are proposed. Character-by-character text entry, text entry with bigrams and with text entry with bigrams derived by word prediction, as well as possible selection methods, are examined in a longitudinal study. Data showed large advantages of the bigram entry methods over single character text entry in speed and accuracy. Participants preferred the novel selection method based on saccades (selecting by borders) over the conventional and well established dwell time method.

On the one hand, pie menus showed to be a feasible and robust widget, which may enable the efficient use of mobile eye tracking systems that may not be accurate enough for controlling elements on conventional interface. On the other hand, visual perception on mobile displays technologies need to be examined in order to deduce if the mentioned results can be transported to mobile devices.

Optical see-through devices enable observers to see additional information embedded in real environments. There is already some evidence of increasing visual load on the respective systems. We investigated visual performance on participants with a visual search tasks and dual tasks presenting visual stimuli on the optical see-through device, only on a computer screen, and simultaneously on both devices. Results showed that switching between the presentation devices (i.e. perceiving information simultaneously from both devices) produced costs in visual performance. The implications of these costs and of further perceptual and technical factors for mobile gaze-based interaction are discussed and solutions are proposed.

Deutsche Zusammenfassung

Blickbasierte Mensch-Computer-Interaktion ist seit einem viertel Jahrhundert ein relevantes Forschungsthema. Der überwiegende Einsatz von Blicksteuerung beschränkte sich darauf, das Menschen mit Behinderungen kommunizieren können. In dieser Form, können z.B. ALS Patienten allein durch ihren Blickbewegungen Texte schreiben, Rollstühle bewegen und Bedürfnisse mitteilen. Durch die rasante Entwicklung von mobilen Endgeräten und tragbaren Displaytechnologien, öffnete sich ein neues Anwendungsfeld und damit neue Forschungsfragen.

Im Rahmen dieser Dissertation wurden grundlegende Interaktionsmöglichkeiten mittels Blicksteuerung entwickelt und erforscht, die das gesamte Potential von Blickbewegungen als Eingabemöglchkeit ausnutzen. Blicksteuerung charakterisiert sich dadurch, dass sie die schnellste motorische Bewegung ist und unwillkürlich gesteuert wird. Sie bildet damit Aufmerksamkeitsprozesse ab. So kann der Blick nicht nur als Mittel zur Eingabe dienlich sein, sondern er verrät auch etwas über die Intentionen und Motive des Nutzers. Dies für die Rechnersteuerung zu nutzen, kann die Eingabe mittels Blicken überaus einfach und effizient machen und zwar nicht nur für motorisch beeinträchtigte, sondern auch für gesunde Nutzer.

Diese These erforscht die Machbarkeit von mobiler Blicksteuerung. Sie untersucht im Detail den Einsatz von Pie Menüs als generisches und robustes Widget für Blicksteuerung, sowie visuelle und wahrnehmungspsychologische Aspekte bei der Nutzung von mobilen optischen Datenbrillen. Diese Arbeit fasst konventionelle blickbasierte Interaktionsmethoden zusammen und untersucht im Detail die Verwendung von Pie Menüs für Blicksteuerung. Es erforscht und diskutiert Layout-Probleme, Auswahl, Methoden und Anwendungen von Pie Menüs. Die Ergebnisse zeigen, dass Pie Menüs bis zu sechs Elemente in Breite und Tiefe, in mehreren Schichten zuordnen können, so dass eine schnelle und präzise Navigation durch die hierarchischen Ebenen gewährleistet ist. Durch die Nutzung oder die Kombination mehrerer Selektionsmethoden, kann eine effiziente und effektive Interaktion gewährleistet werden. Gestützt von diesen Ergebnissen, wurden diverse auf Pie Menüs basierte Texteingabesysteme entwickelt. Diese Texteingabesysteme bauen auf der Eingabe von Einzelbuchstaben, Bigrammen, und vorhersgesagten Wörter. Die genannten Systeme sowie zwei Selektionsmethoden der blickbasierten Interaktion wurden untersucht. Die Ergebnisse zeigen signifikante Unterschiede bei der Geschwindigkeit und Genauigkeit der Texteingabe zugunsten des auf Bigrammen basierten Texteingabesystems, im direkten Vergleich zur Methode der einzelnen Buchstabeneingabe. Die Probanden präferierten, die neue aus Sakkaden basierte Selektionsmethode, über die konventionelle und gut etablierte Schwellzeit (dwell time) Methode.

Pie Menüs erwiesen sich als ein praktikabel und robustes Widget, dass die effiziente Nutzung von mobilen Eye-Tracking-Systemen und Displays, auch bei geringer Genauigkeit, ermöglichen kann. Nichts desto trotz, muss die visuelle Wahrnehmung in mobilen op-

tischen Datenbrillen untersucht werden, um die Übertragbarkeit der bereits benannten Befunde für Datenbrillen sicher zu stellen.

Ziel der AR-Ausgabegeräte ist es, die reale Umgebung mit virtueller Information anzureichern. Die Vorstellung dabei ist, dass die virtuelle Information sich in die reale Umgebung einfügt, d.h., dass Betrachter die virtuelle und reale Information zu einem Bild integrieren. Aus psychologischer Perspektive ist es einerseits plausibel, dass Informationen in raum-zeitlicher Nachbarschaft zusammenzufügen sind. Andererseits kann eine vollständige Integration nur dann erfolgen, wenn die Darbietung als einheitlich wahrgenommen werden kann. Dagegen sprechen allerdings zwei grundlegende Punkte; zum einen das Selbstleuchten der virtuellen Information, zum anderen deren Größen- und Entfernungshinweise. Das Selbstleuchten der per AR-Gerät eingeblendeten Information ist deshalb problematisch, weil dieses Merkmal bei realen Objekten kaum vorhanden ist. Insofern sollte eine vollständige Integration der Information von dem AR-Gerät und anderer Information höchstens dann erfolgen können, wenn es sich bei der realen Information um Reize auf einem Computermonitor handelt, die ebenfalls selbstleuchtend sind. Für andere reale Objekte sollte die Leuchtstärke der eingeblendeten Information allein ein wesentliches Unterscheidungsmerkmal darstellen. Ein weiteres wichtiges Unterscheidungsmerkmal ist die Größeninformation, die einen bedeutenden Anteil an der Entfernungsschätzung hat: In unserer realen Welt werden Objekte, die sich vom Betrachter entfernen, auf zunehmend kleinere retinale Areale projiziert. Gleich große Objekte müssen also mit zunehmender Betrachtungsdistanz kleiner abgebildet sein. Bei der AR-Technologie werden nun Objekte, wie bei einem Nachbild, in konstanter retinaler Größe projiziert, unabhängig davon, wo in der Tiefe sie gerade lokalisiert werden. Da die Objekte üblicherweise auf bzw. vor dem nächsten Hintergrund wahrgenommen werden, sind Größeninformationen der virtuellen Objekte nicht zur Entfernungswahrnehmung zu verwenden. Sie führen sogar teilweise zu widersprüchlichen Tiefenhinweisen, wenn die Distanz zu einem Hintergrund vergrößert wird, die Reize aber die gleichen retinalen Areale beanspruchen und somit bei größer werdender Entfernung als größer wahrgenommen werden.

Für diese These wurden drei Versuchsanordnungen entwickelt, mit denen jeweils bestimmte Aspekte der gleichzeitigen Wahrnehmung von Information auf einem AR-Gerät und realen Umwelt detailliert untersucht wurden. Die Ergebnisse zeigten, dass die gleichzeitige Wahrnehmung von Information aus realen und virtuellen Medien mit kosten in der Sehleistung verbunden ist. Weitere Untersuchungen zeigten, dass das visuelle System bei der gleichzeitigen Darbietung von virtuellen und realen Reizen ständig die Einstellung von Vergenz und Akkomodation ändern muss. Dies könnte die visuelle Beanspruchung erklären, die in zahlreichen Studien beobachtet wurde. Die Auswirkungen der genannten Wechselkosten, Wahrnehmungs- und technischen Faktoren für mobile blickbasierte Interaktion werden hier diskutiert und Lösungen werden vorgeschlagen.

Contents

List of Figures

List of Tables

1. Introduction

1.1. Introduction

Gaze-based interaction is without doubt one of the most exciting interaction methods with computer devices. It allows the user to interact with the computer remotely, by just looking at targets shown in a display. To make this possible a camera system (eye tracker) grabs the movements of the eyes and a special software maps the gaze information into computer coordinates. In this way, gaze offers a unique opportunity for interaction and communication for motor handicapped people, such as ALS patients [66]. For them, gaze interaction provides means to type text, express needs or even control wheelchairs with the gaze [71].

Eye movements are mostly implicitly controlled and provide information about our attention, current interest point, and in some cases, even about some cognitive and psychological processes [11]. Zhai [91] describes gaze control as an attentive input method, which can be in the position to predict future areas of interests on a scene. In this way the user can be guided towards possible next steps, wrong selections can be avoided or even intelligent user assistance can be provided.

One of the main advantages of eye gaze control is its pointing speed. The eye movements are the fastest motor movements that a person can perform with a rotation speed of up to 700°/s [15]. Furthermore, a pointing task implies looking to an object before being able to point at it. This means, that gaze interaction offers the potential to be the fastest input method for human-computer-interaction. Furthermore, gaze control provides an alternative interaction method for people who can not use their hands to interact with a computer for example commissioners who have their hands occupied, factory workers using protection gloves, pilots, surgeons, and the already mentioned handicapped population. Moreover gaze interaction can reduce the fatigue and potential repetitive stress injuries caused by not ergonomic computer interfaces [93].

The interaction with a graphical user interface (GUI) describes two tasks, namely pointing and selecting an interactive element. The pointing task refers to moving and placing the cursor on the computer screen towards the intended item (referred also as move operation [90]). After finishing the pointing task, which mostly has a visual feedback, known as mouse over effect, the selection task takes place. Selecting an item implies an explicit action (for example a click on a mouse or a tap on a trackpad, referred also as push operation [90]) to confirm the selection of the element.

The vision of a (non intrusive) ubiquitous personal computing and entertainment system is getting real. Smart-phones make possible to interact with hundreds of computer applications, like web-browsers, email, organizers, entertainment, etc., everywhere. The

next generation of mobile infotainment systems are expected to be less intrusive than mobile smart-phones, using semi-transparent glasses as displays and eye movements for interaction [40] (for detailed information visit the iSTAR project[1] and the Interaction via gaze project of Nokia Research Center[2]). In this way, the user is always wearing the display getting the desired information on-demand. Gaze-based interaction presents for this developments the only interaction mechanism plausible to guarantee a high user experience.

1.2. Motivation

The call for paper for the "Eye Tracking Research & Applications Symposium" 2012 (which is the main conference for eye tracking research) goes under the theme "Mobile Eye Tracking":

> *"Mobile devices are becoming more powerful every day. Embedding eye tracking and gaze-based applications in mobile devices raises new challenges and opportunities to many aspects of eye tracking research..."*

This two sentences describe the motivation of this thesis. On the one hand, it is indispensable to pose a robust interaction widget and selection methods for gaze-based interaction. On the other hand, the research results acquired on "conventional" displays can not been transposed into mobile displays per-se. This makes absolutely necessary to understand how visual perception on mobile displays works, before being able to draw conclusions about this topic.

Despite the promising qualities that gaze interaction poses, replacing "What You See Is What You Get" by "What You Look At Is What You Get" is still being a real challenge. A fast and accurate direct manipulation and interaction with widgets through eye movements is much more difficult as it seems to be at the first glance. There are several facts that makes gaze computer interaction a difficult task. First, the lack of an explicit, intuitive and accurate selecting mechanism: For instance, selecting an item with a computer mouse is a quite intuitive and straight forward task. The user moves the mouse with his/her hand and the movement of the mouse is mapped to the cursor on the screen. The buttons on the mouse suggest pressing them for selection. For gaze interaction only the pointing task happens almost automatically, mapping the gaze coordinates to the cursor, but there is no explicit and intuitive selection mechanism. For example, blinking or dwelling with the gaze over an item for a certain time may also happen unintentionally. This may lead to the so-called Midas-Touch-Problem, related to the greek mythos of king Midas, who could transform everything he touched into

[1] iSTAR project: http://www.istar-project.org (last visited on 18/05/2011)
[2] Nokia Research Center – Interaction via gaze: http://research.nokia.com/page/4861 (last visited on 18/05/2011)

gold. Moreover, the eye is mainly a perceptual organ and users are not trained (on their daily activities) to interact with any device using eye movements. Second, due to jitter movements and the one degree size of the fovea, the gaze is not accurate enough to interact with standard GUI widgets, which are, at a normal interaction distance of 50 cm, mostly smaller than the mentioned resolution. Third, the practical accuracy of the eye tracking systems and the quality of the calibration decays over the time, making the pointing task more difficult. These challenges have motivated researchers over the past quarter century to investigate towards possible solutions to these three problems. A goal of this thesis is to present and study an interaction widget based on pie menus, which can provide an intuitive, effective and efficient solution for gaze-based interaction. Therefore, this thesis investigates the design space of gaze controlled pie menus, presents and discuss possible applications, interacting and selection methods.

As already pointed out, mobile displays (specially new technologies like see-through-displays) poses different characteristics to conventional displays (like CRTs or LCDs), which may hinder or constrain the transfer of well established gaze-based interaction concepts to mobile devices. For that reason, this thesis studies perceptual issues in a mobile environment, using a optical-see-though head-mounted-display, and compares the vergence point of the gaze axis with conventional and a see-through display. Finally it draws conclusions and suggest future research for this field.

1.3. Outline

This thesis provides a theoretical introduction to gaze-based interaction and recapitulates the research done on the selection task (chapter 2). Chapter 3.3 focuses on question which selection method can be used with pie menus. The design criteria for gaze controlled pie menus is analyzed and discussed in chapter 3. Chapter 4 investigates the transferability of gaze controlled pie menus to diverse applications in different contexts. A longitudinal study using pie menus for text entry with different assisting techniques is presented in Chapter 5. Chapter 6 describes a scenario for mobile gaze-based interaction and the corresponding perceptual issues to be considered for its development. This thesis is closed with a general conclusion on how this research helps the development of mobile gaze interaction and discuss possible future research topics in this field.

The main research questions addressed in this thesis concern:

- Selection method for pie menus (Chapter 3.3)
- Design criteria for pie menus (Chapter 3)
- Transferability of gaze controlled pie menus (Chapter 4)

- Learning effects in adaptive text entry systems(Chapter 5)
- Analyzing perceptual issues on Head Mounted Displays for mobile gaze interaction (Chapter 6)

which are discussed in detail during this work.

1.4. Publications

This thesis is based on the following articles, published in journals or conference proceedings:

Journal articles:

Huckauf, A. & Urbina, M. H. (2008). On object selection in gaze controlled environments. Human-Computer-Interaction. Journal of Eye Movement Research, 2(4):4, 1-7.

Urbina, M. H, Huckauf, A. & Majaranta, P. (submitted). Pie Menus at a Glance: Gaze-Computer Interaction with Pie Menus. ACM Transactions on Interactive Intelligent Systems.

Conference Proceedings:

Urbina, M. H. & Huckauf, A. (2010). Pies with EYEs: The Limits of Hierarchical Pie Menus in Gaze Control. In Proceedings of the 2010 Symposium on Eye Tracking Research & Applications (Austin, Texas, March 22 - 24, 2010). ETRA '10. ACM, New York, NY, 93-96.

Urbina, M. H. & Huckauf, A. (2010). Alternatives to Single Character Entry and Dwell Time Selection on Eye Typing. In Proceedings of the 2010 Symposium on Eye Tracking Research & Applications (Austin, Texas, March 22 - 24, 2010). ETRA '10. ACM, New York, NY, 315-322.

Huckauf, A., Urbina, M. H, Böckelmann, I., Schega, L., Doil, F., Mecke, R. & Tümler, J. (2010). Perceptual Issues in Optical-See-Through Displays. In Proceedings of Applied Perception in Graphics and Visualization, Los Angeles, U.S.A, 41-48.

Urbina, M. H., Lorenz, M. & Huckauf, A. (2009). Selecting with gaze controlled pie menus. In Proceedings of the 5th Conference on Communication

by Gaze Interaction - COGAIN 2009, 25-29.

Huckauf, A., Urbina, M. H, Böckelmann, I., Schega, L., Doil, F., Mecke, R. & Tümler, J. (2009). Besonderheiten der Wahrnehmung bei AR-basierten Ausgabegeräten. In 12. IFF-WISSENSCHAFTSTAGE, Digitales Engineering zum Planen, Testen und Betreiben technischer Systeme 6. Fachtagung zur Virtual Reality, Magdeburg. S. 377-383.

Huckauf, A. & Urbina, M. H. (2008). Gazing with pEYEs: towards a universal input for various applications. In Proceedings of the 2008 Symposium on Eye Tracking Research & Applications (Savannah, Georgia, March 26 - 28, 2008). ETRA '08. ACM, New York, NY, 51-54.

Huckauf, A., Urbina, M., Doil, F., Tümler, J. & Mecke, R. (2008). Distribution of Visual Attention with Head-worn Displays. Applied Perception in Graphics and Visualization, Los Angeles, U.S.A., 198-199.

Parts of the mentioned articles are reproduced as they are, other have been updated and revised. I am the first author (or have a significant influence) of all mentioned articles. All these papers are (co-)authored by Anke Huckauf. Her contribution varies from writing single sections over general review to supervising the research.

Further publications, which are not addressed in detail in this thesis are:

Journal articles:

Huckauf, A. & Urbina, M. H. Object selection in gaze controlled systems: What you don't look at is what you get. ACM Transactions of Applied Perception. 8 (February 2011), 13:1–13:14.

Conference Proceedings:

D. Hamacher, S. Erfurth, M. Urbina & L. Schega (2010). Nutzerzentrierte Prüfung des Low-Costs Head-Mountes-Displays Nikon Media Port UP300x für den Einsatz in mobilen Augmented-Reality-Systemen. Arbeitsmed. Sozialmed. Umweltmed., 45 (6), 370-371.

Tümler J., Böckelmann I., Schega L., Hamacher D., Sarius S., Urbina M., Huckauf A., Mecke R. & Grubert J. (2010). Mobile Augmented Reality in der industriellen Anwendung: Erweiterte Nutzerstudie zum kontinuierlichen Einsatz an einem Referenzarbeitsplatz. In 13. IFF-WISSENSCHAFTSTAGE, Digitales Engineering zum Planen, Testen und Betreiben technischer Systeme 7. Fachtagung zur Virtual Reality, Magdeburg.

Grubert, J., Hamacher, D., Mecke, R., Böckelmann, I., Schega, L., Huckauf, A., Urbina, M. H, Schenk, M., Doil, F. & Tümler, J. (im Druck). Extended Investigations of User-Related Issues in Mobile Industrial AR. In Proceedings of the Ninth IEEE International Symposium on Mixed and Augmented Reality ISMAR 2010.

Tümler, J., Roggentin, A., Mecke, R., Doil, F., Huckauf, A., Urbina, M.H., Pfister, E., & Böckelmann, I. (2008). Subjektive Beanspruchung beim Einsatz mobiler Augmented Reality Systeme (Subjective load in wearing mobile augmented reality systems). ErgoMed, 5, 130-141.

2. Human Computer Interaction with Gaze Movements

This chapter is based on the publication:

Huckauf, A., and Urbina, M. H. On object selection in gaze controlled environments. In Journal of Eye Movement Research (2008), vol. 2 of 4, pp. 1-7. [33]

and it has been updated and adapted to fit the scope of this thesis.

2.1. Introduction

The two main challenges using gaze control are, on one side, the selection methods as the eye is used for both acquiring information and making commands, and on the other side, the low accuracy of the tracking systems that makes it difficult to select small interface items. Jacob [38] described the difficulty of selecting an item with the gaze as the "Midas Touch" problem. Due to the lack of an explicit selection mechanism, like a mouse-click, unintended selections are committed. The best established selection method is dwelling (fixating) the eye gaze on the intended target for a certain duration (commonly between 200-1000 ms). However, since fixations are naturally used for visual perception, the dwelling threshold must be either adaptive or from beginning very well chosen considering the needs and expertise of the user [51]. The low accuracy of the eye tracking systems, combined with the jitter movements of the eyes, makes gaze interaction hard to perform. State-of-the-art commercial eye trackers have a theoretical precision of $0.5°/s$, which can be reached with a very accurate calibration and which decays with time. Thus, interfaces for gaze interaction have high requirements for usability that cannot be ignored.

This thesis focuses on both challenges, studying and discussing alternative selection mechanisms to the established dwell time and analyzing in detail the use of pie menus for gaze interaction, reviewing their design, selection methods, learning effects and applicable tasks.

For the better understanding of this thesis and the used terminologies the next chapter provides a brief introduction in eye movements (for a detailed overview, related to eye tracking and gaze interaction, please see [15]). This chapter also gives an overview of some research done and possible solutions for the pointing and selection task.

2.2. Understanding Gaze Movements

The perhaps best known eye "movements" are *fixations*. Fixations are used to stabilize the retina while looking to an object of interest. During fixations, the eyes are not completely steady. They are performing movements called *microsaccades* or *tremors*, which are very small drift movements from fixation point, responsible for the continuous stimulation of the photosensitive cells. Without these movements, the cells would not be able to generate output (having blindness as consequence). Normally, micro saccades have an amplitude of no more than 0.2 degrees.

Saccades are sudden ballistic eye movements that occur between fixations and are responsible for the switching between points of interest. A saccade has an amplitude range between 1 and 40 degrees, taking 30 to 120 ms [15, 39]. A saccade can be voluntary or reflexive driven. A typical saccade traverse about 15 to 20 degrees. Since saccades are ballistic movements they must be "programmed" in the periphery and its direction and amplitude can not be modified after its start. A saccade is normally started with a delay of about 200 ms after the presentation of a stimulus and need a reload offset of also 200 ms before starting a new saccade. In addition to microsaccades and saccades .

Smooth pursuit movements are much slower than saccades and can only be done by following a moving object (can not be performed voluntarily without a visual stimulus).

2.3. Pointing with Gaze

Eliciting an item on a graphical user interface implies two basic functions, pointing and selecting. Pointing with gaze is a straight forward and very intuitive action. The gaze position is registered with an eye tracker, mapping it to the computer cursor. This means, the user only need to look to the item in order to point at it. Glancing to an object is the fastest pointing method on HCI, not only because saccades are faster than hand movements, but also because before being able to point accurately to an item, the user might look at it. The crucial fact that need to be considered when designing a graphical user interface (GUI) that allows accurate pointing is, as already mentioned, the low accuracy of the eye tracking calibration and the jitter movements.

The easiest way to ensure accurate pointing is to use large items [89], largely separated from each other, having as natural consequence the possible allocation of only few items (due to the limited size of the screen) or a deep hierarchical order of items, implying multiple selections to choose one item.

Miniotas and colleagues [58] proposed to expand the targets by an invisible frame. In this way, the user can still trying to focus the center of the visible target, which is

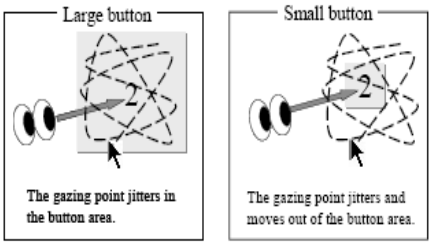

Figure 1: Pointing issues caused by jitter can be solved with large items [90].

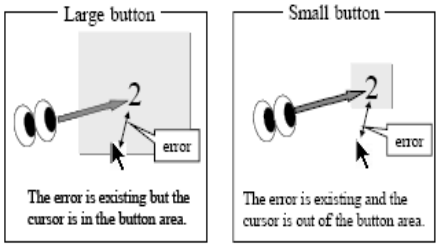

Figure 2: Accuracy problems can be solved with large items [90].

selected if the gaze remains within the invisible surrounding frame for a certain time. They described an increase on effectivity and efficiency using the expanded areas, but also pointed out that even the interface may look tidy, it can not allocate many items, since the distance between the items might consider the invisible frame.

2.3.1. Zooming Interfaces

Bates and Istance [6] explored the usability of zooming interfaces with various pointing devices, like mouse, head and eye movements. They found out that the size of the target has a significant influence on users' performance and the use of discrete zooming-in levels on graphical interfaces improves considerably the efficiency of gaze control.

Ashmore and colleagues [3] examined the use of a fisheye lens for eye pointing (see Figure 3). The fisheye perspective was only turned on during fixation, and off during saccades, which means, that only on fixations the region near the fixation's point was magnified. Their results showed a benefit in terms of usability using the described fisheye implementation compared to an omnipresent fisheye (always turned on) and interaction without zooming.

Positive results using zooming interfaces achieved also Fono and Vertegaal [18], reporting

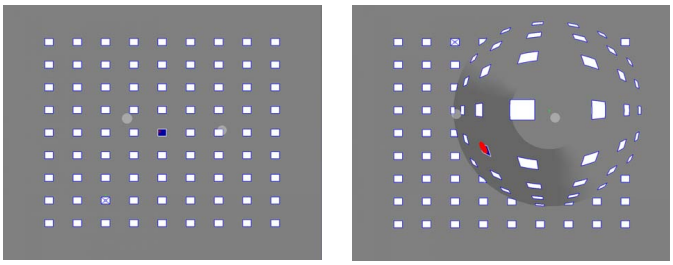

Figure 3: Item selection without magnification (left) and with the fisheye lens [3].

up to 30% faster interaction times using zooming windows over static windows.

Two further zooming interfaces for text entry, namely *Dasher* [86] and *StarGazer* [25] are going to be presented and discussed in section 2.4.6.

2.3.2. Context dependent pointing

MAGIC (Manual And Gaze Input Cascaded) is the name of a innovative pointing technique presented by Zhai and colleagues [93]. The main idea was to warp the cursor to a target that the user is looking at, saving movement amplitude and therefore reducing selection times. MAGIC described two interaction approaches, one liberal and one conservative (in terms of target identification and cursor placement). In the liberal approach, the cursor is moved near or over each target the user fixates. If the user wants to select the target, he/she needs to complete the movement manually. The conservative approach wraps the cursor to the gaze coordinates on the screen only after the user moves the cursor with the hand. Using the MAGIC pointing approaches was shown to reduce the physical effort and pointing time compared with manual pointing and presented better accuracy than traditional gaze pointing.

MacKenzie and Zhang [50] presented a typing system that uses character prediction in order to correct drifts and inaccuracy from the tracked signal within a specified range. For example, after entering “th”, it is highly possible that the following character be an “e”. If the registered gaze position is over the letter “d”, the prediction algorithm corrects the gaze position, warping it to the letter “e” (see Figure 4)

Zhang and colleagues [94] developed a force field metaphor in order to attract the cursor is close to a target and warping it to the center of the target, reducing the effect of jitter movements and low accuracy of gaze pointing. They reported an increased pointing speed as well as reduced error rates compared to the baseline without force field.

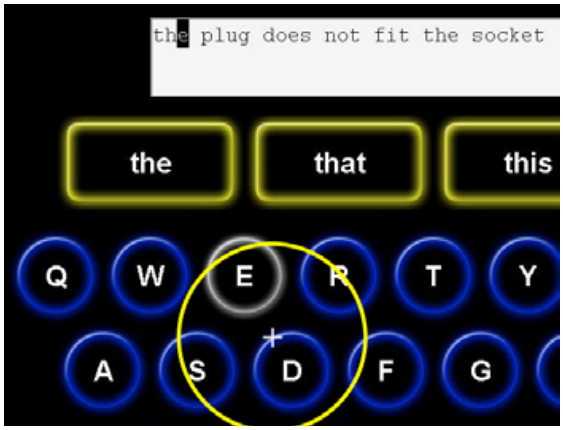

Figure 4: Visualization of the fixation algorithm [50].

2.4. Selection with Gaze

The eye is primary a perceptual organ and it is mostly not trained for control tasks [91]. Moreover, most eye moments happen unintentionally. The lack of an explicit and intentional selection metaphor, which is efficient, effective, and transferable is still being a string motivation for research. This section presents a wide overview of most selections methods for gaze control introduced on the last quarter century.

2.4.1. Blinking for Clicking

One immediately obvious solution to select objects via gaze is using blinks. Blinks happen about ten times per minute [12]. These frequent automatic blinks must be distinguished from the intentional blinks for object selection. Typically, an involuntary blink lasts less than 140ms [4]. Therefore, this threshold can be used to distinguish intentional from non intentional blinks for selection tasks. The main challenge that implies using blinks for gaze-based selection, is that the vergence changes with closed eyes so that the current fixation position is lost after a blink. A reliable work arround to this problem is using a scanning keyboard. Here, letters are organized into a matrix or groups. The system moves the focus automatically by scanning the alphabet item by item (e,g [4, 73]). An item is selected by blinking when the desired item is on focus.

Ashtiani and MacKenzie [4] proposed a text entry system, *BlickWrite2* (see Figure 5), which is based on a scanning keyboard and three time intervals for blink selection. A selection was triggered by a blink's length between 140 ms and 540 ms. Blinks of 540 ms to 1200 ms were classified as "jump blinks" and blinks longer than 1200 ms as "deletion

blinks".

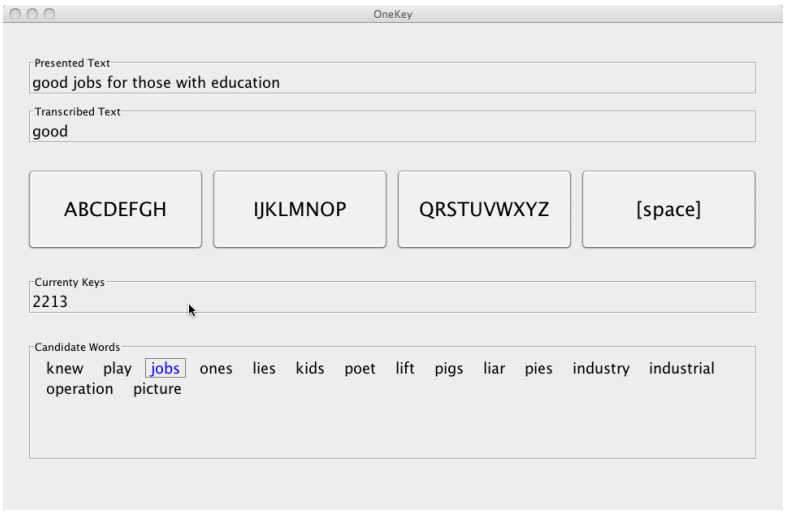

Figure 5: Text entry with BlickWrite2. The word "jobs" is on focus.

A jump blink, switches the scanning focus from the text entry buttons to predicted candidate words, and vice-versa. In a user evaluation, 12 participants achieved an average text entry rate of 5.3 wpm.

More explicit than blink selection is the use of winks. However, not everybody is able to wink. In addition, with head-mounted cameras, winks may lead to movements of face and forehead, which may displace the cameras and thus, have an influence on calibration.

Unfortunately, a detailed practical comparison between blinking and other selection techniques is still missing.

2.4.2. Dwell Time

Dwell time (DT) is probably the most used selection method for gaze interaction. Here, as mentioned above, fixations are used for selection, which last typically from 200ms to 1000ms. DTs can be found in the majority of gaze controlled applications. For example, in eye typing applications, they are used to select and enter letters (e.g., [57, 26, 53]). In eye drawing, the start and end points, as well as various formatting tools, are selected using DTs [29]. With prolonged fixations, applications are started, files are managed and even menu items can be selected [42]. However, selections with DT imply several disadvantages. Figure 6 points out, the exact problematic of using dwelling times. This figure shows an sing located in the "Ilm Park" in Weimar, on which is written in German

"Hebe Deinen Blick und verweile", which means *"Raise your gaze and dwell"*. It points to exceptional beautiful views, and motivates the sightseers to enjoy the view. Fixating or dwelling on a certain point with the gaze is the natural way to gain visual information.

Figure 6: Text entry with BlickWrite2.

This process needs to be clearly discriminated from the selection information. The only practicable method is setting an optimal and situation specific threshold. In order to avoid the Midas Touch problem, a longer DT can be used to trigger selections. In return, this has the disadvantage of hampering the user's performance. DT also limits the achievable speed of system control: whereas eye movements and thus gaze control can be extremely fast, dwelling on a certain object reduces or even destroys this advantage. Selection with DTs provides a limited space for improvement after a considerable amount of training. Moreover, gazing at an object for a long time requires effort, and, in case the gaze leaves the objects' area, refixations.

Huckauf and Urbina [34, 44] and also Špakov and Miniotas [85] proposed the on-line adjustment of dwelling times, according to the correctness of selections, with the goal to find a good trade of between efficiency and affectivity. The problem here is that the system must be able to distinguish between intended selections and errors.

2.4.3. Antisaccades

Huckauf and Urbina [34] examined the usability of antisaccades [44] as a selection method. Antisaccades are an explicit eye movement that have been extensively investigated in cognitive psychology (e.g. [24], [44] and [63]). In antisaccade tasks, an observer fixates on an object in the center of a screen and is presented with an eccentrically located stimulus. The participant is required to make a saccade in the direction opposite to the stimulus, using the amplitude given by the distance between fixation and the stimulus. That is, a stimulus presented 1° to the right of the current fixation requires a saccade of 1° to the left. The explicitness of antisaccades relies on the inhibition of reflexive response to the stimulus, which attracts the attention of the observer, and performing a saccade in the opposite direction. Thus, the landing position of an antisaccade is determined by a stimulus that is distant from the landing position. Antisaccades have been found to be more error-prone and slower than prosaccades [17]. However, with training, antisaccades can be as accurate and fast as prosaccades [17]. Thus, by training subjects to use antisaccades, an object of interest might quickly be selected.

In antisaccade selection, the eye moves to the target of interest. Then, the target is copied at a second location. The eyes are free to investigate the target and its copy without temporal constraints. The procedure of selection by antisaccades is depicted in Figure 7. Here, an object of interest is first fixated. On fixation, the object is highlighted and a copy of it is presented to one side (second step in Figure 7). If either the original object or its copy is gazed upon, nothing happens. As soon as gaze shifts in the direction opposite of the copied object, the target is selected, which is indicated by changing the color of the object to blue, the disappearance of the copy (step 3 in Figure 7), and a click sound. Results showed, although achieving shorter task completion times, the effectiveness of anti-saccades was below that of adaptive DTs. The application of antisaccades is restricted to certain situations, since it requires space on the screen for the target, the stimulus and an unoccupied area to perform the anti-saccade. Nevertheless, it could be deduced that pure (anti)saccade selection may be used as an alternative selection method to DT [30].

2.4.4. On- and Off-Screen Buttons

Ware and Mikaelian [87] suggested using buttons, which were placed on or off the screen in order to select objects. An object is selected by a fixation and a subsequent saccade towards the button. In their user study, these keys were used effectively for selection. Nevertheless, screen buttons require gazing through various objects on the screen, which are situated between the targeted object and the button. This may interfere with focussing on the relevant object and distracts attention from the area of interest. Taken together, Ware and Mikaelian [87] have shown that on- and off-screen buttons require

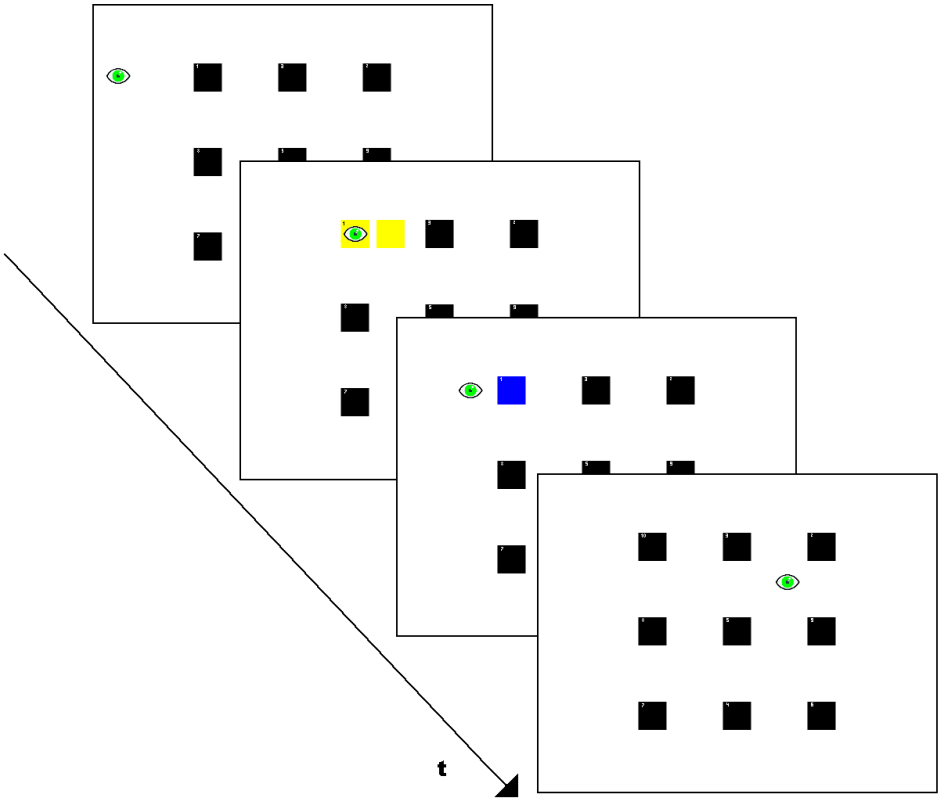

Figure 7: Nine black squares labelled with small numbers were to be selected by gaze using antisaccades. Fixated objects are marked by the yellow color and the stimulus appears (second step). As soon as the eye points to the area directly opposed to the stimulus, the object is selected (indicated by the blue square, third step), and the eyes proceed to another square.

additional dwell times on the objects in order to work effectively. A similar selection method was also developed for eye typing [81] (*Iwrite*). Here, the letters were positioned along the border of the screen. The area between the letters and the border served as selection button. With this alignment, the previously mentioned unintended selections were avoided and no dwell time was required, triggering the selection by entering in the selection area.

Isokoski [35] proposed the use of off-screen targets and various schemes for decoding target hit sequences into text. Off-screen targets help to avoid the Midas' touch problem and conserve display area. However, the number and location of the off-screen targets is a major usability issue. They discussed the use of Morse code, a so-called Minimal Device Independent Text Input Method (MDITIM), QuikWriting, and Cirrin-like target arrangements. They reported as the major motivation that off-screen buttons could

solve accuracy problems of gaze-based interaction but having at the same time as cost the lack of visual feedback on selection.

2.4.5. Context Switching

With a similar approach to on-screen buttons, Morimoto and Amir [62] introduced a selection method based on two separated regions, called contexts. The user needs to focus an item for a short time, and then switch the context (perform a saccade to the other context). Comparing it with screen buttons, context switching saves one saccade per selection (on text entry). Since both contexts contain exactly the same items (letters), the user does not need to gaze back to the key board (see Figure 8).

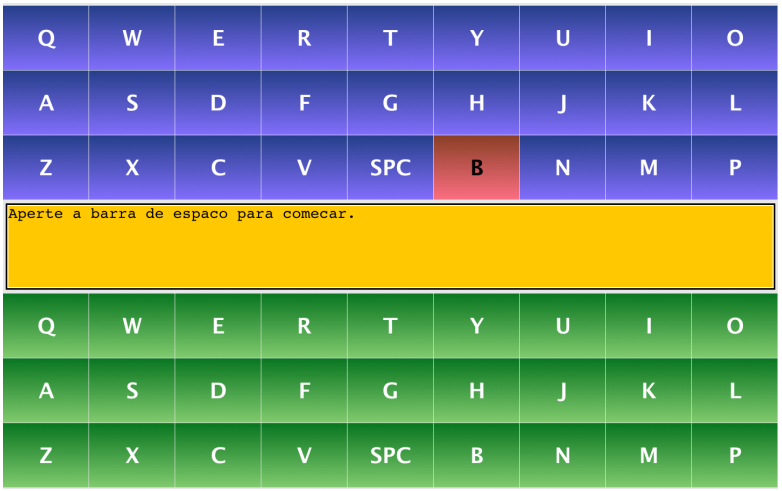

Figure 8: Keyboard using the context switching paradigm.

2.4.6. Interaction with Pursuit Eye Movements

It is not rare that the accuracy (i.e. calibration) of the eye tracker is below the level the user would hope for. As a consequence, this may cause unintended selections or make the interaction practically impossible. The introduction of eye trackers built with off-the-shelf components, like web cameras [2], makes even clearer the need for systems that can be used with a low accuracy.

Two text entry systems have been developed that are based on pursuing objects for selection, namely *Dasher* and *StarGazer*, moving characters are followed by the user's gaze with pursuit movements, triggering their selection.

In *Dasher* [86], letters are presented vertically in alphabetical order on the right side of the screen. In order to select a letter, the user points (gazes) at the desired letter. The interface zooms in, and the letter moves towards the centre. The letter is selected when it crosses a vertical line located in the middle of the window (see Figure 9).

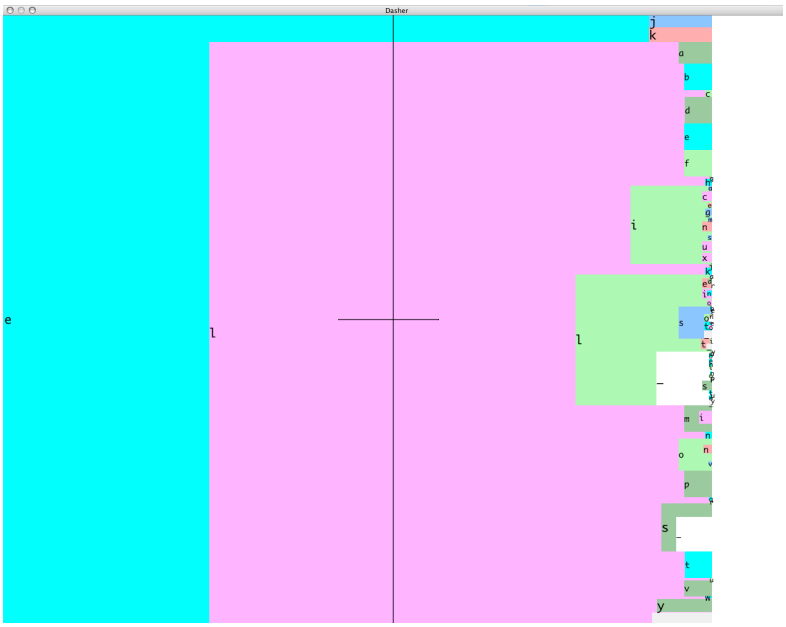

Figure 9: Text entry with Dasher.

Gazing to the right from the vertical line leads to a selection of the target. Gazing left leads to the deletion of written text, causing the opposite visual effect. The distance between the gazing point and the vertical line controls the writing and deleting speed. This means looking to the middle of the vertical line ceases any action. The cooperation of vertical navigation and horizontal selection leads to a smooth motion of the letters. Considering the difference of text entry speeds of 4.5 wpm without text prediction [81] and around 20 wpm [79] with word prediction, it is clear that a character or word calculation algorithm is needed to make gaze based text entry more fluent. A word completion algorithm increases probable items and brings them into the centre of the screen. After the user and the word completion algorithm are sufficiently trained, typing with Dasher becomes fast and fluid, feels comfortable and even fun [79].

StarGazer [25] is like Dasher, a zooming interface. However, the zooming occurs along the z-axis, while the x- and y-axis are used for panning. The letters are ordered alphabetically in a circular form. The display pans towards the direction of the gaze while zooming is continuous, comparable with skydiving. Letters are selected when the desired letter has been zoomed enough. After selection, all letters appear again in the circular order. Hansen et al. [25] reported text entry speeds up to 8 words per minute.

In both Dasher and StarGazer, the user follows the letters with pursuit eye movements, which makes these interfaces very suitable for gaze interaction. Even though the navigation and zooming for every single character selection takes on average at least as long as a normal dwell time, their zooming technique allows low accuracy eye tracking.

2.5. Multimodal interaction

As alternative method for object selection, the addition of other modalities has been suggested. For example, Zhai and colleagues [93] as well as Kumar and colleagues [45] and Yamato and colleagues combined gaze control with manual reactions (key strokes). Surakka and colleagues [74] suggested frowning to assist gaze control. Kaur and colleagues [43] as well as Miniotas and colleagues [59] complement gaze control with speech. Furthermore electromyography (EMG) has been applied with remarkable results [56]. Due to the explicit nature of the mentioned selection modalities, these systems were shown to provide of fast and efficient control. However, they afford an additional device. This reduces some of the advantages of gaze control; namely the idea of saving channel capacity and having ones hands free. Hence, using additional modalities must be restricted to certain settings of tasks and users.

2.6. Gaze Gestures

In the past few years, several alternatives to DTs have been developed and investigated (e.g. [81]). Gaze gestures have shown to be a reliable alternative to dwell time based selection. Gaze gestures comprise a sequence of saccades (or fixations) on certain positions in order to select an item. Since the gestures are based on eye movements, they are theoretically very fast and allow to inspect the scene in a secure way. Gaze gestures have a tremendous potential for gaze interaction. In order to take advantage of them, detailed research needs to be conducted. Drewes and Schmidt [13] investigated the feasibility of gaze gestures for human computer interaction, studying gestures of different complexities and directions with different backgrounds (homogeneous, inhomogeneous and with helping points). They found out that the time needed to perform a gesture depends mostly on the number of strokes used to perform the gesture (referred as segments), taking about 550 ms to complete a stroke. They also found out that is easier to perform large scaled gestures than shorter scaled gestures (see Figure 10).

Møllenbach and colleagues [61] investigated the effects of amplitude and direction of single gaze gestures (gestures consisting of a single stroke). They found out that horizontal saccades were significantly faster than vertical, and could replicate the results of Drewes and Schmidt [13] showing that short gaze gestures were performed significantly faster than larger gestures. Heikkilä and Räihä [27] explored multiple gestures for drawing

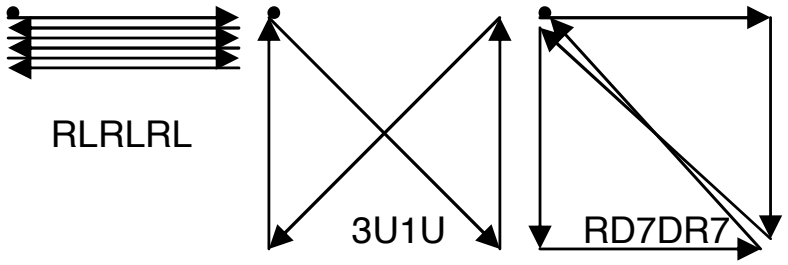

Figure 10: Example for multi-stroke gestures [13].

applications reporting longer stroke completion times than Drewes and Schmidt. In addition, Istance and colleagues [37] took a indepth look into gaze gestures in the context of gaming. They investigated gestures consisting of two or three strokes and discovered the horizontal gestures (e.g. from left to right and back) to be faster than vertical gestures.

The main drawback of the gestures based interaction is the large number of gestures required for a basic alphabet. The gestures should distinguish clearly from each other, increasing the number of strokes per gesture (and in this way implicitly the selection time) and also the mental load needed to recall every single gesture. This drawback can be avoided using pie menus.

2.7. Pie Menus on Manual Control

Pie menus have already shown to be powerful menus for mouse or stylus control [46, 47]. They are two-dimensional, circular menus, containing menu items displayed as pie-formed slices [28]. Selection is done pointing to the desired item and confirming by a mouse click or a stylus tap (see Figure 11a). One of the main advantages of pie menus is that interaction is very easy to learn. A pie menu presents items always in the same position, so users can match predetermined gestures with their corresponding actions. This fact can be fully exploded by experts users, who do not need to search for menu items, as they are able to "mark ahead" without waiting for the menu to pop up. Due this marking technique pie menus are also known as marking menus (see Figure 11b).

Finding a trade-off between user interfaces for novice and expert users as well as accurate pointing and selecting methods are the main challenges in the design of a gaze controlled interface, as it is less conventional and utilized than input controlled by hand. Therefore in this thesis pie menus are transferred to gaze control. The first step is to explore the design space, selection metaphors, arrangement of items and use cases of pie menus in

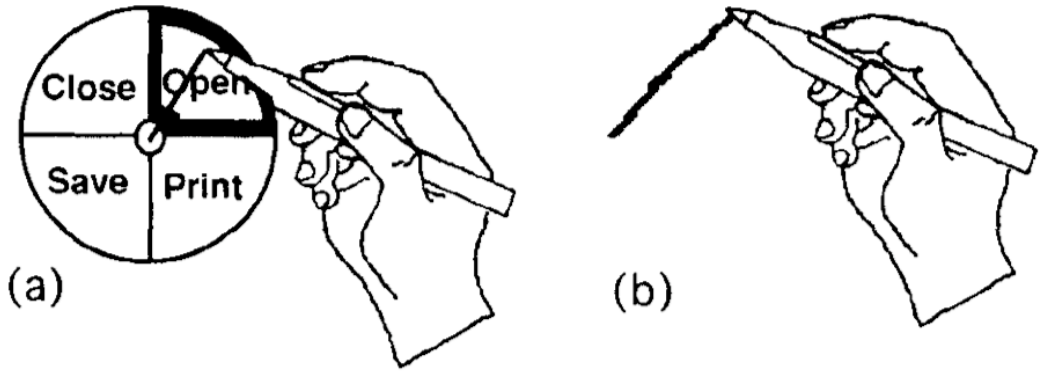

Figure 11: Marking Menus. a) Item selection. b) Selection by "marking ahead" [47].

gaze interaction.

3. The Design of Pie Menus for Gaze Controlled Environments

This chapter is based on the publications:

Urbina, M. H., and Huckauf, A. Selecting with gaze-controlled pie menus. In *Proceedings of the 5th International Conference on Communication by Gaze Interaction* (COGAIN 2009) (2009), pp. 25-29.

and

Urbina, M. H., Lorenz, M., and Huckauf, A. Pies with eyes: the limits of hierarchical pie menus in gaze control. In ETRA '10: *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (New York, NY, USA, 2010), ACM, pp. 93-96.

3.1. Introduction

Since the early 90's it is known that pie menus are promising tools, which could replace in some cases pull-down menus [28]. Pie menus are two-dimensional circles, centered on the mouse cursor. Their elements are ordered like pie slices. This means, all elements are at the same distance from the cursor which accelerates selections relative to pull-down menus, where a user needs to find the desired item by iterating downwards through the menu. Moreover, pie menus allow a fluent transition from novice to expert usage, since menus' items are always placed at the same position. Hence, novice users can search for the desired item whereas expert users know the exact item position, just needing to perform a certain movement trajectory without needing to look at the menu (see Figure 11b). This way of selection is also called "marking ahead" [46].

One important feature responsible for the good usability of pie menus is that the slices increase in size towards their outer border. This facilitates accurate slice selection, even in conditions of low spatial accuracy of the input device - what can enhance performance especially in gaze input. However, although they have lot of advantages over pull-down menus, pie menus never became popular. This can be attributed mainly to the exceptionally high familiarity of users with pull-down menus [92]. Nevertheless, for gaze input, pie menus have already been shown to be a reliable alternative to pull-down menus [42]. As shown earlier, they can be effectively and efficiently used even for tasks like eye typing [81].

This chapter presents related work done on pie menus, to later focus on selection methods

for pie menus and the study of their optimal design.

3.2. Related Work

After the first presentation of pie menus for text entry in gaze-controlled interfaces in a prior work to this thesis [81], numerous researchers adopted and applied pie menus to their interfaces. This section review and discusses their approaches.

3.2.1. Text Editing with Pie Menus

Majaranta and colleagues [54] suggest pie menus as a tool for gaze based text editing. Having the editing commands in a pie-like pop-up menu that is located on top of the text offers several advantages. First, pie menus can help in preserving precious screen space. Conventional eye typing systems often use virtual keyboards, based on QWERTY or alphabetical layout. Since the keyboard itself allocates most of the screen estate, there is not much space left for editing commands (such as copy, paste, bold, and underline). The editing commands are therefore often hidden in the virtual keyboard's menu structure. With pie menus, the commands can be located right where the user needs them, near the text to be edited. This also reduces the need to swap the gaze between the virtual keyboard and the target text.

Majaranta and colleagues suggest having a hole in the middle of the pie so that the text to be edited can be seen "through" the pie. Thus the commands are located in the individual pie slices around the text. The pie menu can be shown right on the point of the user's focus by fixating (using DT) on the piece of text that needs editing. Showing the pie, and more importantly, placing the cursor exactly on the desired location may be difficult because of the inaccuracy problems involved in tracking gaze position. Therefore, adding arrow keys into the pie menu itself could to facilitate easy navigation in the text.

Majaranta and colleagues tested the idea in a feasibility study with 13 novice participants. They compared the (dynamic) pie menu with a (static) menu that was located below the text in simple formatting and editing tasks. There was a trend toward faster task completion times when using the pie menu for simple formatting tasks (such as bolding), whereas the conventional menu seemed to be better suited for more complex tasks that required several steps (e.g. moving text by first selecting it and then performing cut and paste). The participants' opinions varied towards the dynamic versus static pie, though the opinions were a bit more positive towards the static pie. This may be partly due to technical and usability problems that arose during the test. This study was the first step towards more user-friendly text editing by gaze, further research and

development is required to exploit the full potential of pie menus for gaze based text editing.

3.2.2. Snap Clutch

Istance and colleagues [36] propose "Snap Clutch", a general tool that can be used in various applications to implement keyboard and mouse events by gaze alone. Snap Clutch transfers the commands to the underlying application as if they came from the keyboard or mouse. Snap Clutch also implements means to change the mode of interaction in order to avoid Midas Touch problem in gaze-controlled online games. Istance and colleagues investigated three modes to interact securely with 3D games and investigated the user performance on tasks such as locomotion, object manipulation and application control [84]. The objet manipulation task consisted on changing a slide from a presentation or to request a web page to be displayed in the online game "Second Life" (Figure 12). To achieve this, the user needs to trigger a pie menu and select the correct item on it. For the application control, the task consisted on changing the appearance of the avatar selecting the passing actions through a pie menu. They did not report any issues with pie menus (but found issues with for e.g. dialog boxes) showing the adaptability of pie menus on complex tasks and different contexts.

Figure 12: The pie menu allows different actions performed to an object (image courtesy of Dr. Howell Istance).

3.2.3. Customized Widgets

Tall [76] proposed gaze interaction interface components which are, in theory, not affected by the Midas-Touch problem. One of these components was based on pie menus, which was triggered after fixating at the desired item. The interaction options popped up with the pie menu, and the selection was done by making a short saccade to this options (options were visualized by icons see Figure 13).

The usability of the "Radial Pie Menus" was tested in real world oriented tasks, like

Figure 13: The Radial Saccade Pie Menu. Upon gaze entering the component a opaque ellipse expands from underneath the button. Four icons appears on the ellipse. A fixation starts the activation process which is indicated by a glowing border. Both the expansion time and activation time can be configured. The number of icons used is optional between 1-4 (image courtesy of Martin Tall) [76].

playing music or viewing pictures, using three different activation times (10 ms, 300 ms and 500 ms). Results showed a low variability on the selection time and that a long dwell time for activation hampers the performance of the users. The selection time achieved with the short activation time is in line with the item selection times reported by Urbina and colleagues [83] for pie menu with four items, validating their findings.

All the implementations of pie menus reviewed in this section use dwell time (in different length) to select a target (a pie slice). On their original work, Urbina and Huckauf [81] used a dwell time free selection metaphor, based on saccades. This opens the question which selection method should be used to allow efficient and effective selections.

3.3. The Problem of Selecting a Pie Slice by Gaze

In manually controlled pie menus, a slice is selected via pressing a key (for mouse input) or via tapping (for stylus input). In gaze-controlled pie menus, selection is usually performed via dwell times (e.g.,[42, 36]). That is, a fixation longer than a critical duration on a certain object leads to its selection. In favor of dwell time selection, one might assume that for users it is relatively easy to determinate their own point of gaze - although eye movements are usually unconscious. However, this method requires an optimal setting of the threshold: dwell times that are too short will produce numerous unintended selections, whereas those too long require the user to hold the gaze unnaturally long on a certain location, thus slowing performance [38]. For this reason, an alternative method of selecting a slice on pie menus was suggested [33]. Here, a selection border is used for selection (see Figure 14). Whenever the gaze crosses the line between the inner pie and its outer frame (i.e., the selection border), the respective slice is selected. That is, selections can be performed via a fixation within the selection frame (behind the selection border) or via any saccade crossing the selection border. Although this method turned out to be effectively usable [81], its advantages and disadvantages relative to the standard selection by dwell time are unclear. Providing a detailed comparison between selection

by dwell time and by selection border is the aim of the present study. The main idea of the concept of the selection border is twofold: for novice users, the selection borders do not constrain the fixation's duration within the menu. That is, novices can investigate the menu as long as they need to. Experienced users, however, might simply perform slightly longer saccades when navigating through the menu in order to directly select a certain slice. Nevertheless, how effective and efficient this selection method can be realized by the users is unknown, as is the learning rate.

In a first attempt to provide alternative selection methods to dwell time for gaze interaction, Urbina and Huckauf [81] presented three novel eye typing applications. *pEYEdit*, based on pie menus, showed outstanding results, accomplishing all usability criteria. The reported results showed that text entry could be performed at a reasonable speed and accurately using pie menus. Either way, it remained open if the text entry performance achieved was due the pie menus, the selection metaphor used (gazing to the border of he pies), or a combination of both.

In order to deduce which selection methods is better suited to be used with pie menus, the usability of dwell time and saccade based selection was compared.

The mentioned selection methods were compared (by dwell time and by selection border) with novice users, who repeated the same task five times. The task was chosen from the original work from Kurtenbach and Buxton [46] in which they evaluated users' performance for manually controlled pie menus. In the current study, after training, performance in a marking ahead trial was additionally examined (i.e., without presenting the pie menus or any visual help on the screen, beside the tasks and the starting point). The purpose of this "blind" trial was to investigate whether users after short training can indeed make advantage of the marking ahead option of pie menus in gaze control (i.e. use pie menus to learn intuitively gaze gestures). One might assume that a selection method based on fixations (i.e., selection by dwell time) may require longer selection times, but allows for more precision and spatial accuracy by determining the fixation points on marking ahead trials. Or, with other words, whereas the selection border allows for faster selection times, it may be less accurate and thereby less suited for marking ahead.

3.3.1. Methods

Stimulus Pie menus consisting of four slices were presented in three hierarchical layers. The slices were labelled with four corresponding orientations (N - north, O - East ["Ost" in German], S - south, W - west). The pie menus had a radius of 180 pixels (about 7 degrees of visual angle, see Figure 14) in addition to an outer frame of 20 pixels used to remark the selection border.

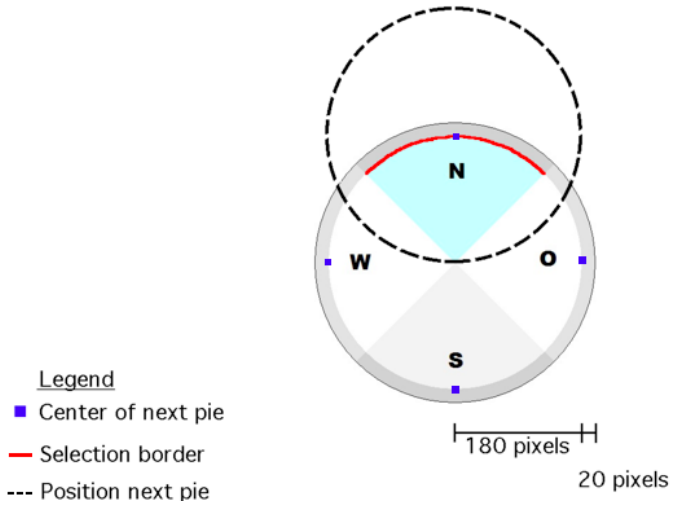

Figure 14: The pie menu implementation in detail. Selection border (in red), position of the upcoming pie (dashed line) and its centre (blue square). All items listed in the legend are highlighted in this figure only for visualization.

Equipment The pie menus were presented on a 21" Sony GDM-F520 CRT display with a resolution of 1280x960 at a frame rate of 75Hz. The experiment ran on a Dell-PC with Pentium 4 processor at 2,6 GHz and 1024 Mb DDR-Ram under Windows XP. The experiment was programmed with Matlab using the Psychtoolbox 2.54 and the EyeLink Toolbox. The eye tracking device used was a head-mounted Eyelink2 from SR-Research (see Figure 15). The eye tracker was calibrated before each block. The calibration consisted on focusing small spots (of about a 1/3 degrees) ordered on a 3x3 matrix, including the four corners, their midpoints, and the centre of the screen. In order to keep the calibration as accurate as possible, a chin rest was used, situated at 51 cm in front of the monitor. The estimated spatial resolution of this set-up, considering the nominal tracking resolution of 0.5°, was about 12 pixels. The experiment took place in a room without windows under indirect artificial lightning.

Subjects A total of ten students and faculty from the University of Weimar volunteered as participants (aged between 21 and 30, m=25.5, sd=2.83). All subjects had normal or corrected-to-normal vision.

Task The task consisted on selecting three presented coordinates (e.g., N - W - W, see Figure 16) on the centre top of the screen. After reading the task's coordinates, participants were required to fixate on the central start button (see Figure 16 a). Instantly, the pie menu popped up. In order to enhance selecting performance, each selection was

Figure 15: Subject during the experiment, wearing the head mounted eye tracker.

accompanied by a click sound [52]. Immediately, with a selection either the next level pie popped up (Figure 16b-d), or in case the most top layer was reached, the menus were closed and the start button appeared again with the next task (Figure 16e). Depending on the task, selections were performed either via crossing the selection border (see Figure 14) or via 400 ms of dwelling on a slice. Five of ten participants started with either condition. For each selection method, five blocks of 32 trials and a sixth marking ahead block were performed.

3.3.2. Results

For the first five blocks, task completion time (TCT, measured from onset of the first pie until closure of all pies) was entered in an ANOVA with the repeated measures factors selection method (2) and blocks (5). In mean, selecting via selection border took 569 ms (standard error se = 29 ms), and selection via dwell time took 589 ms (se =14 ms, see Figure 17). This difference was not of significance ($F < 1$) as was the interaction with blocks ($F(4, 36)$ =1.33; p =.29).

Performance differed between blocks ($F(4, 36) = 7.41$; p <.01) which should be ascribed to learning (see Figure 18).

The corresponding analysis for the error data revealed a significant effect of selection method ($F(1, 9)$ =27.5, p <.001). Selection via selection border resulted in 9.4% of errors (se =2) and selection by dwell time in 21.12% (se =3.45, see Figure 19).

As in TCTs, blocks differed significantly from each other ($F(4, 36)$ =4.89; p <.05), and there was no interaction between both variables ($F < 1$). As depicted in Figure 20,

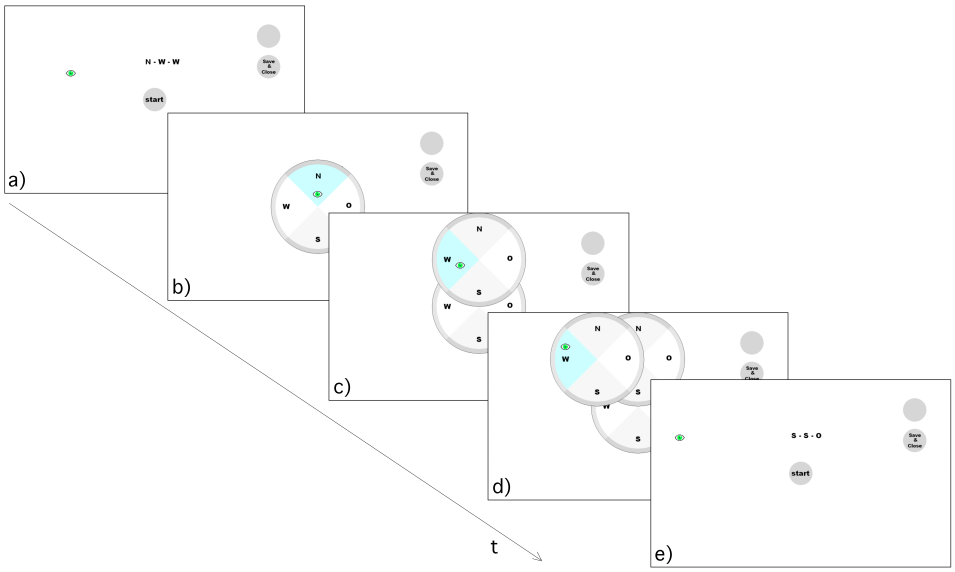

Figure 16: Selection sequence for task North - West - West.

differences between the first five blocks might be attributed to learning.

Performance in the "marking ahead" block was compared to the fifth block. As expected, TCTs in the marking ahead condition were worse than in the fifth block ($F(1,9) = 76$, $p < .001$). Of more interest, neither the selection method nor the interaction with it revealed any significant effect ($F < 1$). In the corresponding error analysis, however, not only the block ($F(1,9) = 55.8$, $p < .001$), but also the selection method ($F(1,9) = 88.5$, $p < .001$) was of significance. Moreover, there was a marginal interaction effect ($F(1,9) = 4.4$, $p = .06$) showing that the impediment by presenting no visual information was especially disturbing with dwell time selection.

3.3.3. Discusion

In mean, selecting via selection border turned out to be as fast as dwell time selection after very short training, showing thereby significantly fewer errors. Error rates for selection via dwell time were, even in the fifth block, still larger than those in the first trial achieved with the selection border. The relative high error rate for selection by dwell time suggests that the critical dwell time of 400 ms might have been too short. Extending it, however, would lead to longer TCTs. Hence, selection via selection border seems to be the more promising method. The current data did not reveal a significant difference between both selection methods in learning rates. However, for longer training durations, and for various tasks, it cannot be ruled out that both methods differ in their

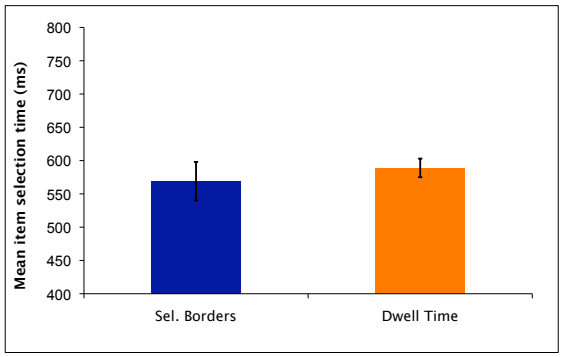

Figure 17: Mean task completion times of each selection method.

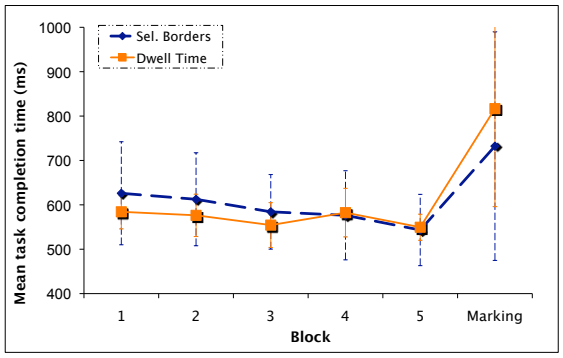

Figure 18: Mean task completion times for each block, separately for the selection methods.

steepness of the learning function. The presented results suggest that selection through selection border can be more benefited from training than selection by dwell time. The clearest superiority of selection border over dwell time selection was observed in the "marking ahead" block. This was astonishing because during dwell time selection, users had to fixate longer on the pies, which should have facilitated the spatial orientation and navigation after several repetitions. However, one point in favor of selection border is the positioning of the pies: since the upcoming pie is centered always in the selection border (see Figure 14), the distance between the centre of the pies in different layers had already been trained during the first five blocks. This distance corresponds to the amplitude of the selection movement, which in most of the cases was from the centre of the pie to the middle of the selection border (i.e. the centre of the upcoming menu). It is difficult to estimate this movement in dwell time selection, where the user can select the pie slice anywhere within the slice. This may imply an offset to the next pie's centre, which needs to be corrected in order to perform a gesture. Thus, for the marking ahead performance, and therefore also for expert behavior, these findings indicate that

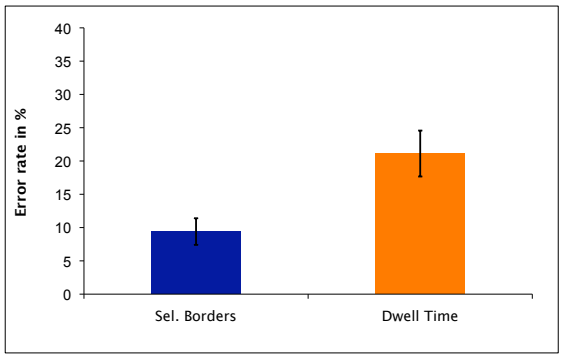

Figure 19: Mean error rates for the blocks, separately for each selection methods.

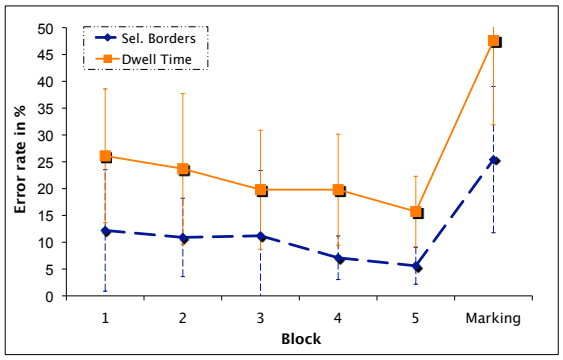

Figure 20: Mean error rates for the blocks, separately for each selection methods.

gestures with constant amplitudes are to be trained in menus, more than certain landing positions. This should be regarded in the design of pie menus. In sum, it can strongly recommend the usage of pie menus, primarily selecting via selection border.

3.4. The design of Pie Menu for Gaze Interaction

A key attribute of pie menus is that the interaction is easy to learn and the transition form novices to expert performance is a fluent process [92]. In a pie menu, items are presented always at the same position, so users can match predetermined gestures to their corresponding actions. This fact can be fully exploited by expert users, who do not need to search for menu items, as they are able to "mark ahead" without waiting for the menu to pop up [46]. Due their circular form, all items are at the same distance to the cursor, reducing the amplitude of selection movements in comparison with pull-down menus. Furthermore, since the slices increase in size towards their outer boundaries, the

spatial accuracy of selection movements can be easily adapted by the user simply by adjusting the amplitude of the movement.

Despite the fact that pie menus perform better than pull down menus using mouse or stylus [10, 47], they have not been adopted as a standard for user interaction. The main reason for this can be seen in the well established pull down menu and the barriers that unfamiliar interfaces pose (e.g. [92]).

One field in which there are only few standards established is gaze based interaction. Here, pie menus can be expected to work well; especially because small spatial resolution is still a matter in gaze input which might be compensated by pie menus. Indeed, pie menus have already been demonstrated to work well in gaze control (e.g. [36, 42]), in tasks requiring orientation (e.g. hierarchical desktop navigation [32]) as well as in tasks requiring frequent selections [81], allowing a fluent learning process of gaze gestures, contrary to other systems using gaze gestures [88, 69], where the users need to learn all gestures before being able to interact properly with the system.

3.4.1. Research Questions

When designing a pie menu the crucial factors to be consider are the number of slices and the number of depth layers in which the information is presented. For mouse and stylus input, these questions have been elaborately examined by Kurtenbach and Buxton [46]. They found that presenting two to three layers in combination with eight slices per menu results in fluent behavior and good task performance. However, the data provided for manual control cannot be directly transferred to gaze input because of various differences: especially due to technical restrictions in eye tracking systems [25] and because of motor constraints like tremor and drifts, the spatial accuracy is reduced. Moreover, voluntary control of eye movements requires conscious effort. This may vary or even restrict the complexity of the menus.

The aim was thus to investigate the limits of the menu width (i.e., the maximum number of slices (items) per pie) and menu depth (i.e, the maximum number of layers) for gaze control. This was done by replicating the study of Kurtenbach and Buxton [46] by using pie menus with gaze input.

In addition to this replication, the marking ahead strategy was explored with different layouts. Therefore, besides performance increments after training, performances and trajectories in conditions where no visual cue was depicted were investigated.

A further research question concerned the optimal method of selecting a slice. Results presented on chapter 3.3.2 pointed out, that selection borders produced less errors by a comparable selection time than dwell times. In order to confirm the reported results with

different layouts, a comparison between selection borders and dwell times was included in this study.

3.4.2. Method

Design Independent variables throughout the study were the number of slices per pie (width), the number of hierarchical layers per pie (depth), and the method of selection. These factors were varied block-wise. In total, 13 blocks by 32 trials were to be performed with the configuration and selection method described in Table 1.

Table 1: Menu layout, selection method and visualization condition for all 13 blocks.

Block #	Width	Depth	Sel. Method	Visualization
1	4	2	sel. borders	yes
2	8	2	sel. borders	yes
3	6	2	sel. borders	yes
4	12	2	sel. borders	yes
5	4	2	sel. borders	yes
6	4	3	sel. borders	yes
7	4	4	sel. borders	yes
8	4	2	sel. borders	yes
9	4	2	sel. borders	no
10	4	2	sel. borders	yes
11	4	2	dwell time	yes
12	8	3	sel. borders	yes
13	8	3	dwell time	yes

Regarding the number of slices per pie, pie menus consisting of pies with 6, 8, or 12 slices and two depth layers were presented, each within a block. For pies with six slices, the 32 trials were chosen randomly. For pies containing eight or twelve slices, 16 trials equalled the ones for the four-slices and two-layers condition. The other 16 trials were chosen randomly from the rest of possible combinations.

The question of how many layers can be effectively and efficiently controlled was examined with pie menus containing four slices. These pies were presented in either with two (e.g. Figure 16c), three (e.g. Figure 16d), or four depth levels (each on a separate block by 32 trials). Trials with two layers were the same as in described previously. Trials for the pies with three and four depth levels were chosen randomly with the restriction that trials consisted of maximally two steps either vertically or horizontally, due to the limitations of the screen size.

In a further block, which always followed directly the third replication of the four slices

and two layers-block, participants had to perform 32 trials without any visual presentation (so-called marking ahead-condition). All blocks described so far were to be performed by selecting the slices via selection borders. In selection via borders, as previously mentioned, a slice is selected by moving the eyes over the critical border (see Figure 14).

There were two blocks in which selection by dwell times was applied; one consisting of a pie with four slices arranged in two layers, the other of a pie of eight slices arranged in three layers. For dwell time selection, the critical threshold was set at 400 ms which is in the range reported on further literature (e.g [39, 51].

Errors and item selection times (ISTs, measured from the onset of the pie until the selection of one slice) served as dependent variables. ISTs were computed instead of the usual task completion times in order to compare performance between the different menu layouts. An error was defined as every single false selection. For example, for the task "N - O", the selection of "N - W" or "O - O" was counted as one, the selection of "W- N" as two errors.

Procedure The task consisted on selection several several given coordinates through the pie menu. The task was depicted above the centre top of the screen (see Figure 16a). After having fixated the start button in the centre for 200 ms, the pie menu popped up. Simultaneously, the task description disappeared from the screen, and time measurement started. Each selection was measured, and was accompanied by a click sound [53]. With a selection, either the next pie layer popped up or, the menus were closed and the start button appeared again together with a new task until the block was finished (see Figure 16).

At the beginning of each block, the eye tracker was calibrated. Participants were instructed to select the respective slices as fast and as accurate as possible.

Apparatus The experiment took place in a room without windows under indirect artificial lightning.

The pie menus were presented on a 21" Sony GDM-F520 CRT display with a resolution of 1280x960 at a frame rate of 75Hz. The experiment ran on a Dell-PC with Pentium 4 processor at 2,6 GHz and 1024 Mb DDR-Ram under Windows XP. The experiment was programmed with Matlab using the Psychtoolbox 2.54 and the EyeLink Toolbox. The eye tracking device used was a head-mounted Eyelink2 from SR-Research. The eye tracker was calibrated before each block. The calibration consisted on focusing small spots (of about a 1/3 degrees) ordered on a 3x3 matrix, including the four corners, their midpoints, and the centre of the screen. In order to keep the calibration as accurate as

possible, a chin rest was used, situated at 51 cm in front of the monitor. The estimated spatial resolution of this set-up, considering the nominal tracking resolution of 0.5°, was about 12 pixels.

Stimuli Each pie menu had a radius of 200 pixels (6.97 cm), corresponding to a visual angle of about 7.8°. Hence, the slices expanded at their outer border to 314 pixels (10.95 cm), 209 pixels (7.29 cm), 157 pixels (5.47 cm) or 105 pixels (3.66 cm) depending on the number of slices (4, 6, 8, or 12, respectively). The slices were colored alternating with white and light grey. Any gaze into a slice let to highlighting it using light blue (e.g. Figure 14).

Selections were performed via gazing through the selection border (see Figure 14). The next pie opened centered around the outer border of the selected slice (i.e see Figure 16c).

Menus with four slices were labelled as “N” - North, “O” - East [in the used language], “S” - South and “W” - West. Menus with eight slices were labelled “N”, “NO”, “O”, “SO”, “S”, “SW”, “W” and “NW”. Menus with twelve slices were labelled as a clock (from “1” to “12”), and in menus with six elements, the clock using only even numbers served as content (“2”, “4”, “6”, “8”, “10” and “12”).

The start button had a radius of 50 pixels and was presented in the centre of the display. The task was presented 150 pixels above the start button using the “Arial Black” font with a text size of 30 pixels. This font and size was used with all labels.

Participants Twelve volunteers participated in our experiment, aged between 23 and 30 (26 in mean). All reported normal or corrected-to-normal vision, and were familiar with computers and with mouse and keyboard usage. Two of them had prior experience with eye tracking and pie menus.

3.4.3. Results

Selection speed (IST) and accuracy (error rate) data were entered into analyses of variance for repeated measures. Post hoc comparisons were performed using Newman-Keuls test. Except for the investigation of learning effects, data for the menu of four slices presented in two layers were taken from the second run.

Width **Selection time:** For investigating the effects of menu width, blocks with menus of four, six, eight, and twelve slices were compared. All these menus consisted of two depth layers. For four slices, item selection took 667.141 ms (standard error se =31.18). For six slices, the time to select one slice was in mean 786.35 ms (se=38.60), for eight slices 907.01 ms (se =54.37), and for twelve slices 933.11 ms (se =40.31) (see Figure 21). These differences were of significance (F(3,33)=27.52, p < .001). Post hoc comparisons revealed that all numbers of slices differed significantly from each other except eight and twelve slices.

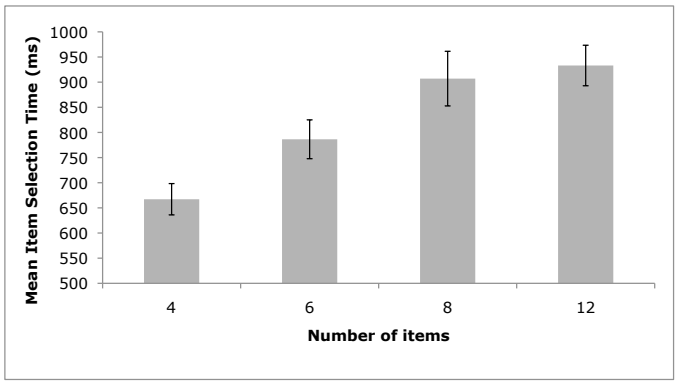

Figure 21: Effect of the number of slices on item selection times.

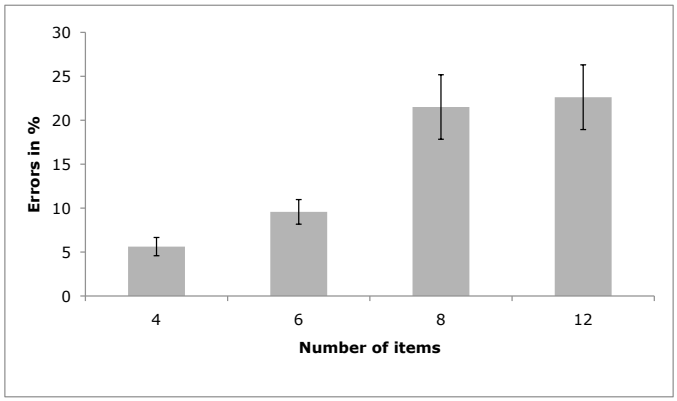

Figure 22: Effect of the number of slices on error rate.

Error rate: For four slices, 5.62% errors were produced (se=1.04). With six slices, the error rate reached 9.58% (se = 1.40), with eight slices 21.51% (se=3.67), and with twelve slices 22.62% (se=3.68) (see Figure 22). Also for the error rate the menu width had a significant effect $F(3, 33)$ =16.77, p<.001. And again, this effect was due to differences between all numbers of slices except eight and twelve slices.

These data indicate that six slices seem to be the maximal number of slices which can be suggested for using pie menus in gaze control both, in terms of fast and accurate

performance.

Depth layers **Selection time:** For examining the effects of number of layers, menus of two, three, and four layers were compared, all based on pies of four slices. Item selection took 667.14 ms (se=31.18) for two layers, 749.85 ms (*se* =48.02) for three layers, and 746.83 ms (*se* =31.76) for four layers (see Figure 23). These differences were of significance ($F(2, 22)$ =9.13, p <.001). Post hoc comparisons showed that this effect was due to the faster items selections with two layers relative to three and four which did not differ.

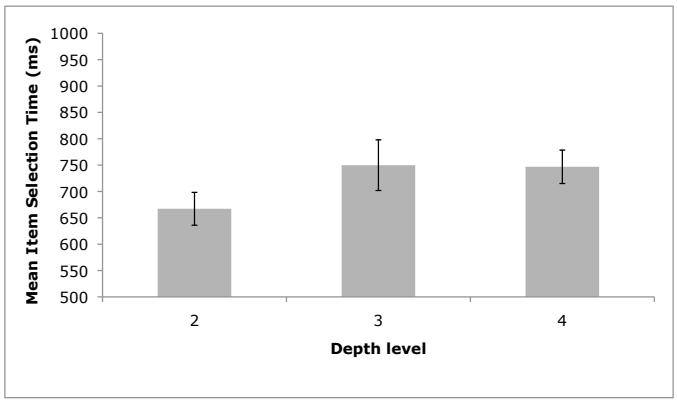

Figure 23: Effect of the number of layers on item selection time.

Error rate: Errors were conducted in 5.62% (se=1.04) for two layers, in 6.03% (se=1.04) for three layers, and in 6.06% (se=1.26) for four layers (see Figure 24). The effect of menu depth on IST was not important (F<1).

The results for menu depth show that the depth of a pie menu is not as crucial in gaze control as is the width. This is in contrast to the data provided for manual control by Kurtenbach and Buxton [46].

Learnability **Selection time:** Effects of learning were investigated comparing performance for the menu of four slices arranged in two layers which was repeated four times throughout the whole experiment. In the first run, users took 817.03 ms (se=61.81) per item. This was reduced to 667.14 ms (se= 31.18) in the second, to 633.46 ms (se=30,36) in the third, and to 586.88 ms (se=28.19) in the fourth run (see Figure 25). The effect of learning was statistically significant (F(3,33)=17.14, p<.001). Each run produced significantly faster selection times, except the second and third (p=.15). The decrease from the third to the fourth run was marginally significant (p=.06).

Error rate: In errors, learning let to a decrease from 16.05% (se=2.73) over 5.62%

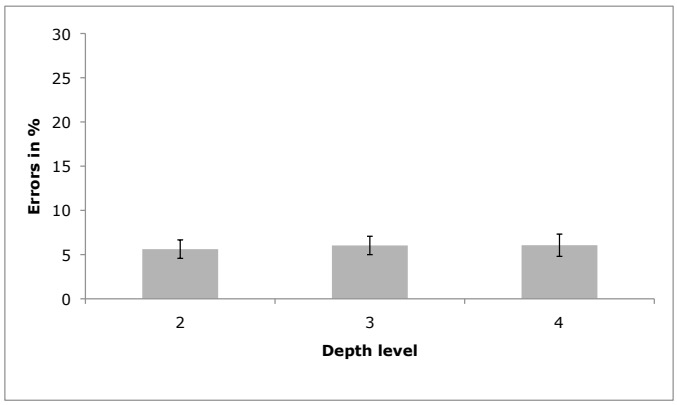

Figure 24: Effect of the number of layers on error rate.

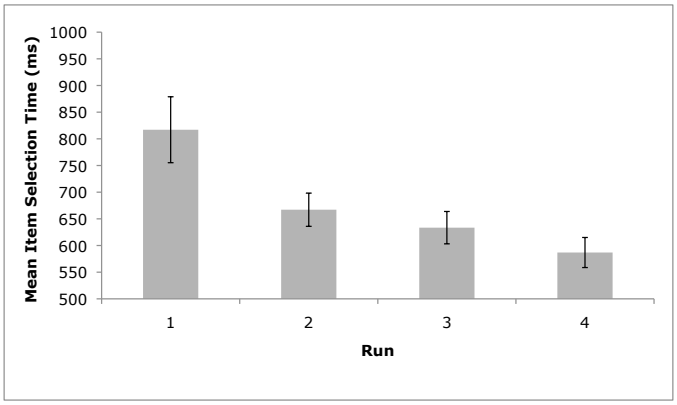

Figure 25: Effect of learning on selection times per item.

(se=1.04) and 3.30% (se=.82) to 5.72% (se=1.24) (see Figure 26). These differences were also of significance ($F(3, 33)=18.63$, $p<.001$). Post hoc comparisons revealed that performance in all further sessions was better than in the first run. The further changes were not of significance (second to third run: p=.59, second to fourth run: p=.94; third to fourth run: p=.073).

Marking Selection In order to further investigate learning effects, one block without visual feedback was performed. The assumption of this marking ahead strategy is that users have a complete mental conception of the whole series of actions. In order to test this assumption, performance in this marking ahead block was compared to performance on the very first run. Importantly, the menu layer (i.e., selection in the first versus in the second layer) was included as a further variable: If users have a mental conception of the whole task, then performance between the steps of both blocks should not differ. If, however, users solve this task step by step, in the marking ahead condition, the first

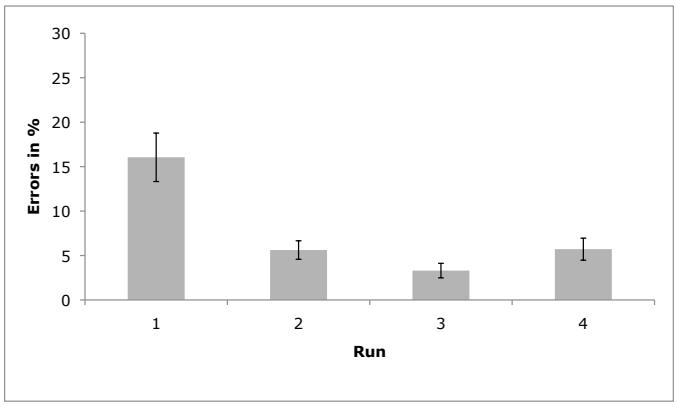

Figure 26: Effect of learning on error rates.

selection might still succeed whereas the second may be more error-prone and less fast.

Selection time: Performance between the very first block and the marking block (i.e., the block without visual presentation of the menu) did only marginally differ ($F(1,11)=4.04$, $p=.07$). In addition, the menu layer (i.e., first versus second selection) differed ($F(1,11)=11.29$, $p<.01$). Whereas the first selection took 951.09 ms (se= 90.18), the second was with 824.31 ms (se=79.53) shorter (see Figure 27). However, there was no interaction between both variables suggesting that there were no specific differences between both blocks ($F<1$).

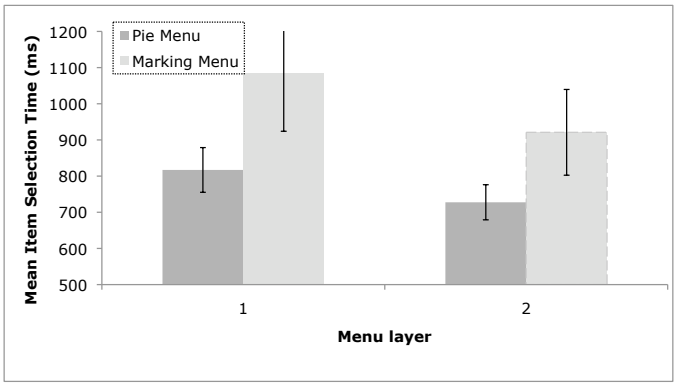

Figure 27: Item selection times for the first and second menu layer separately for the very first block of the marking menu and the marking ahead condition.

Error rate: In errors, performance between the very first run and the marking block did not differ ($F<1$). As in selection times, the menu layers (i.e., first versus second selection) produced a significant effect ($F1,11)= 14.63$, $p<.01$). This was due to with 9.5% (se=1.31) more errors in the second than in the first menu layer (5.88%, se=.83)(see Figure 28). Again, there was no interaction between both variables ($F<1$).

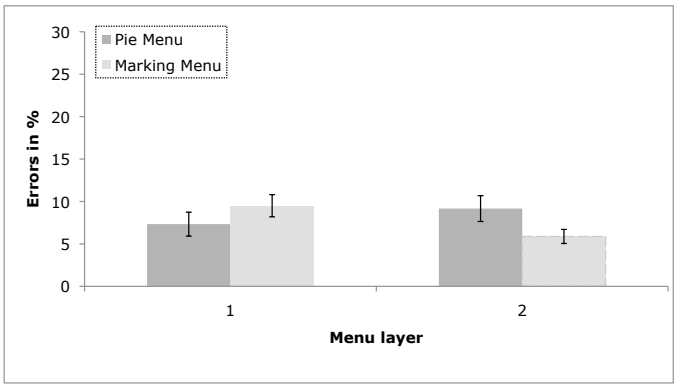

Figure 28: Error rate achieved using pie and marking menus.

Selection Method **Selection time:** The investigation of whether selection via selection borders can actually compete with the standard selection procedure using dwell times was performed on two menu designs: A small menu of four slices and two depth layers and a large one of eight slices and three depth layers. The statistical comparison between them revealed a main effect of menu size ($F(1,11)=58.04$, $p<.001$) where selection took less time in the small menu (663.37 ms, se=25.29) relative to the large one (887.59 ms, se=45.42) (see Figure 29). However, there was neither a main effect of selection method ($F<1$) nor an interaction with it ($F<1$). These data indicate that in terms of selection speed, both selection methods can be regarded as equally useable.

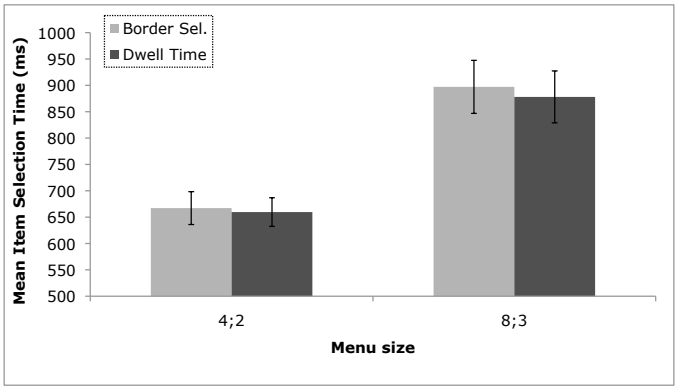

Figure 29: Item selection times using border and dwell time selection separately for the small menu of four slices in two layers and the large menu of eight slices in three layers.

Error rate: In errors, there was also an effect of the menu size ($F(1,11)=19.56$, $p<.001$ in that there were with 10.55% (se=2.02) less errors per selection for the small pie menu as for the large (21.43, se=3.49) (see Figure 30). In errors, the selection method produced an effect ($F(1,11)=7.55$, $p<.02$) in that selection via selection borders was with 11.72%

(se=1.67) more effective than selection via dwell times (20.27%, se=3.91). Again, there was no interaction between both variables (F<1).

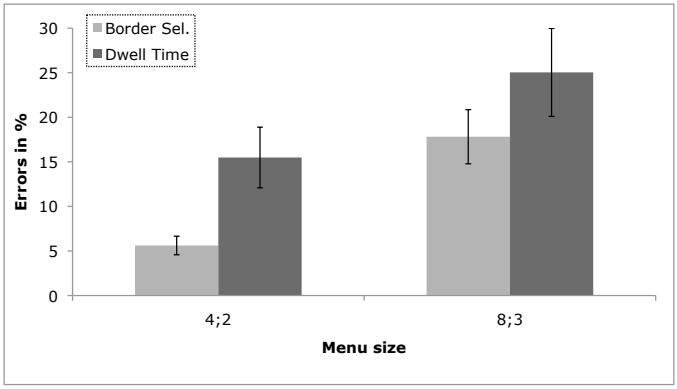

Figure 30: Effect of the selection method on error rates separately for the small and large.

3.4.4. Discussion and Conclusion

The number of items per layer in a pie menu seems to be the most crucial factor when designing pie menus for gaze control. As our data revealed, up to six slices per pie can be effectively and efficiently selected. Of course, this holds for eye trackers up to about 0.5° of spatial accuracy. But, this range can be handled by nearly every current eye tracking equipment. Additionally, one should take into consideration that the tasks for the various numbers of slices varied in difficulty: For four and eight slices, tasks were given with cardinal points, and for six and twelve slices, they were given using the clock. We suppose the cardinal points to be more difficult: Some subjects confused "W" with "O" and vice-versa (like confounding left with right), committing in mean 1.91% errors, which made up about 20% of the total errors committed. For the eight slices per menu, the cognitive load for perceiving and remembering coordinates like "SW- SW - S - W" was perceived larger than for a couple of numbers like "8 - 8 - 6 - 10" used with six and twelve slices.

Performance with two depth layers was found to be significantly faster than with more than two depth layers. One explanation may be, that the participants were able to plan the selection path completely after reading the task. This strategy was harder to follow with more than two depth layers. Even though, the performance achieved with three and four depth layers was acceptable and showed no additional costs presenting more depth layers. Therefore, to allocate more items in a pie menu, the presented data suggests increasing the number of depth layers.

The results show that for gaze control, slice width is more important than menu depth.

This is in contrast to the data provided by Kurtenbach and Buxton [46] for manual controlled marking menus who found no limitation for the number of slices per menu, but a limitation for the number of depth levels of about three. We assume that the difference in number of slices is mainly due to the lower spatial accuracy of gaze tracking, as well as to the difficulty of performing selective actions with a perceptual organ [93].

Of course, the number of layers is restricted by the screen size. Therefore, it may not be infinite. An alternative method of presenting more layers might be to arrange forthcoming pie menus either directly overlaying the former one, or to be centered on the fixation position after selection. Both of these alternatives, however, have a severe disadvantage inherent: Whereas the first solution would require additional saccades back to the starting point, destroying the navigation metaphor adopted for hierarchical menus, the second solution would reduce the capability of marking ahead, since each menu would change in position on the screen each time it appears, which may interfere with the path learning process seen in this experiment.

Subjects showed a significant learning effect using pie menus. Even after 128 selections, they continued improving significantly their speed in item selection, with a constant and relatively low error rate. This confirms the findings with mouse interaction, providing novice users a comfortable and easy to learn interface. Experienced users have been expected to be capable of marking ahead a complete path (or gesture). This can be confirmed for our observers: After already 96 trials with a menu designed with four slices and two layers, the accuracy of performance without any visual cue did not differ from performance within the first 32 trials. Even if there was a lower selection speed for these blind trials, the hypothesis of marking ahead trajectories can be confirmed also for pie menus operated by gaze.

The selection method differed in accuracy, but not in selection speed: Selections with the standard method of dwelling on an item produced more errors than selections by borders. Nevertheless, despite performing below selection via selection borders, dwell time was perceived as a “more natural”, “intuitive” but also “slower” selection method among participants without prior experience in gaze control.

As discussed on chapter 3.3.3, one might thus increase the accuracy (i.e., reduce errors) of dwell time selection by increasing the critical threshold which was set at 400 ms. Then, however, the time to select a slice will also increase thus reducing the selection performance. Hence, as a consequence, one might suppose that selection by selection borders provides a better performance for selecting items in a pie menu than dwell times. The arrangement of the pie menus might also be responsible for the superiority of selection by borders over selection by dwell time: Since all new layers were centered around the outer border of the current pie, selection by borders already brings the eye towards the centre of the next pie menu. One might suggest that with other designs like centering the pie around the current fixation position (similar to the method used by

Kurtenbach and Buxton [46], who presented the next menu level centered on the last selection position), dwell time selection can compete with selection by borders. However, as already discussed above, respective designs may be of disadvantage for the usability and learnability of pie menus.

Considering the experimental results, it can be conclude that pie menus are a suitable and promising interfaces for gaze interaction, which can allocate up to six items in width and multiple depth layers, allowing a fast and accurate navigation through hierarchical levels by using or combining multiple selection methods. This outstanding qualities may give pie and marking menus the chance to establish as a standard in gaze control.

4. Pie Navigation

This chapter is based on the publications:

Huckauf, A., and Urbina, M. H. Gazing with pEYEs: towards a universal input for various applications. In ETRA '08: Proceedings of the 2008 symposium on Eye tracking research & applications (New York, NY, USA, 2008), ACM, pp. 51-54.

and

Urbina, M. H, Huckauf, A. & Majaranta, P. (submitted). Pie Menus at a Glance: Gaze-Computer Interaction with Pie Menus. ACM Transactions on Interactive Intelligent Systems.

4.1. Interaction with Gaze-Controlled Menus

One of the principal ideas of gaze input is to help handicapped people to communicate. Beside text input, other fields of Human-Computer-Interaction are required to guarantee optimal communication and interaction. One of this is choosing items from menus.

Interacting with a menu is a complete different cognitive task as, for example, selecting characters from an on-screen keyboard while typing. It requires orientation for traversing the menu back and forth, recalling or reckoning the position of the desired item on abstract levels and also selecting the item. For this reason, the menu layout and interaction metaphor plays a crucial role for a positive user experience and usability.

4.2. Related Work

Tien and Atkins [77] tested three different menu layouts for gaze interaction: a layout that resembles a typical drop-down menu, a layout resembling gesture-based menus which are found in hand-held devices, and a variation of the gesture-based menu with big buttons and short distance adjusted specially for gaze. They did not find significant differences in task completion times between the layouts. In a follow-up experiment, they provided several improvements, like a "snap-on" feature that snapped the cursor to the center of the button if the gaze is within a tolerance range from the center, visual feedback for the state of the buttons and opening the menu with a quick off-screen glance. In the follow-up experiment, they found that, after memorizing the menu commands,

participants were able to perform correct menu selections using fairly short dwell times of 147 and 177 ms.

Kammerer and colleagues [42] investigated the usability of three pie menu designs, namely a linear menu, a full circle menu and a semi-circle menu. The semi-circular menu performed significantly better with gaze interaction than the linear and the circular menu. They reported the major drawback of circular menus was their confusing arrangement, the ungrouped menu items and the long distances between them. The semi-circle had its items located to only one side of the menu, which probably made it clearer and easier to for navigation than the full circle menu.

Summing up, menu items should have a clearly defined hierarchical order and their items might be placed close together to ensure good usability with gaze interaction.

4.3. Desktop Navigation with Pie Menus

One important question in order to recommend the universal usage of pEYEs is whether these menus work not only in tasks requiring object selection (as in eye typing), but also in tasks requiring orientation and navigation. In order to examine the suitability of pEYEs for such tasks, a further version of pEYEs providing desktop functionality called pEYEtop was implemented.

The functionality of a desktop is a bit diffuse. Desktops provide users with the possibility to start applications and to organize their files and folders. The difficulty in designing a desktop surface lies in the personalized characteristic of the desktop: Nearly no desktop will look the same. There are users who frequently use their desktop for putting currently important files on it, and there are others who refuse this possibility and start working directly in their applications. In order to configure a testable surround, we developed a certain desktop surface, which however, is open to be configured and adjusted by each user.

The idea behind pEYEtop [32] was to investigate the usability of gaze controlled pie menus on a task that not only require repeated selection but also more complex cognitive processes like fast orientation and navigation. The pEYEtop application represents a desktop, containing files and folders, organized in a hierarchical way. pEYEtop is based on a three-layer desktop, which can be enlarged to more layers when deemed necessary. Each layer is illustrated by a pie. As can be seen in Figure 5, files are marked with a green box and are typed in lower case, and folders are marked with a blue box and typed in upper case. The background color is grey. The various nuances of grey should facilitate the differentiation between slices. The currently fixated slice is highlighted by a yellow background. The functions necessary to manage files and folders are creating a new file or folder, deleting a file or a folder, and moving a file or folder (see Figure 32b).

Since cutting and pasting can also be covered by moving, we resigned these commands in order to save space.

On the first level illustrated in Figure 31, applications, files, and folders are grouped thematically in five pie slices. For a first start, the folders in the first level are entitled "music" (containing further files and folders as well as music applications), "texts" (containing pEYEwrite or other writing applications, and texts in files and folders), "pictures" containing certain images, as well as respective software, and "world" which was thought of containing internet applications, chat, and e-mail. In addition, there was one slice containing a picture file ("Aschenputtel") as placeholder. On the second level, the folders, files, and applications appear. Applications are usually started from here, or via opening a file from any level. These second-level folders can be opened to a third level. The current implementation has a maximum of three layers, but obviously, the number of layers can easily be enlarged by adding new folders to the third level. Also, the pie slices are currently restricted to five. The optimal number of levels and slices for a certain user of course will depend on the eye tracking device as well as on the users' needs. The pies for each subsequent level are centered around the chosen pie slice. In order to provide feedback through all traversed levels, they were smaller than in the former level. A selection was triggered using the already mentioned selection borders metaphor, that is, looking to the outer selection area of a slice.

There is one popup pie, which is opened via dwell time (i.e., keeping fixation on a certain pie slice for a certain threshold). In this way, the pie can be opened at each level. In order to mark the popup pie, its background color is red. It contains commands to organize the desktop. The currently chosen commands are "move", "delete", "new file", "new folder", and "x" to close the pie as is illustrated in Figure 32b. Figure 31b

Figure 31: a) In the first layer of pEYEtop, files, folders, and applications are organized in themes folders. This forms the basis of our desktop metaphor. b) Second menu layer.

Figure 32: a) Third menu layer. b) Context pie menu in pEYEtop (in red), providing move, delete, new file and new folder functionalities.

4.3.1. Evaluation

Investigating the usability of a desktop application is not trivial. Firstly, the orienting performance of users will strongly depend on their understanding of the structure of files and folders. Since desktops are personalized, every implemented structure will have its shortcomings at least for some users. Secondly, one should compare its performance with a state of the art-desktop interface for gaze interaction, which were unavailable at this point. Therefore, a standardized series of tasks was developed, which had to be performed by users within a certainly arranged desktop surface (the one described under 4.3). The tasks consist of moving a file from the first to the second level, moving one file from one folder in the third to another one in the same level, opening a file, opening an application, and creating a new folder.

In order to perform these actions, a context menu was triggered after 700 ms dwelling (see Figure 32b). Navigation through the menu levels was performed by looking through the outer border of the pie's underlying slice. Immediately, the next hierarchical level was opened, or in case of files or applications, they were opened.

In addition to measuring time and errors, participants were handed a questionnaire on various features of the desktop. The results had an observation character. An extensive performance measurement using pie menus is resented in chapter 3.

Six participants worked with pEYEtop. After five minutes of free investigation, each of the participants successfully managed to solve the five tasks within eight minutes or less and committing maximal two errors per task. Participants described working with pEYEtop as easy and fast. They reported no difficulties with the way of selecting items, and were fast and secure in finding the desired item. All participants reported to feel comfortable using and navigating with pEYEtop. From that it can be concluded that

orienting within a pie menu seems rather easy.

4.3.2. Discussion of pEYEtop

All users were effectively able to control the desktop application and reported to be able to quickly navigate within the menu and to feel comfortable while using pEYEtop. Pie menus can be used in a navigation context. The limitation of pEYEs is certainly the restricted number of slices and of hierarchy levels. Although pEYEtop is in fact extendable, the number of levels and slices per level needed for comfortable desktop navigation is unclear. Therefore, we are in need of a long-term investigation of users indeed working with pEYEtop in order to see whether it can fulfill their requirements in an efficient way. That is, in addition to testing in an experimental set-up, for the usability testing of the current interface a single case study is strongly recommend.

5. TextEntry with Pie Menus

This chapter is based on the publication:

Urbina, M. H. & Huckauf, A. (2010). Alternatives to Single Character Entry and Dwell Time Selection on Eye Typing. In Proceedings of the 2010 Symposium on Eye Tracking Research & Applications (Austin, Texas, March 22 - 24, 2010). ETRA '10. ACM, New York, NY, 315-322.

5.1. Introduction

Eye-typing describes the process of entering text with eye gaze, which is an indispensable means of communication for motor disabled people who cannot either talk or use a keyboard or a mouse. The most propagated and common way of eye-typing is to select individual characters from an on-screen keyboard by visually fixating on it for a certain time (e.g., [55]).

One of the main problems concerning eye-typing is the slow text entry speed, which ranges from 6.2 [26] to 8.9 [57] words per minute (wpm; [49]). Compared with the average manual typing speed of about 40 wpm [49], 8.9 words per minute is considerably slower - what is certainly also due to the less frequent usage of gaze input. Therefore, promising tools are not only characterized by a higher typing speed, but also by a steep learning curve. This is scarcely achieved with eye-typing systems based on dwell time selection, since a dwell time restricts the typing performance of the user. Furthermore, it is challenging to find an optimal dwell time for each user: Dwell times which are too short will increase the amount of unintended selections, whereas those that are too long will hamper user's performance, since fixating the gaze at one position slows down the intake of new information as well as the execution of new actions [33].

Another reason for the poor text entry performance reported with eye-typing is the text entry method. Character-by-character selection is too slow for eye gaze interaction. Alternatives, like word prediction, are required to enhance the input performance. Other than with Dasher [86], text entry speeds beyond 10 wpm are still uncommon, even when using word prediction algorithms [26, 25, 50].

With the desire to provide alternatives for both selection methods and text entry method, this chapter presents in a longitudinal study alternative ways of eye-typing based on a hierarchical pie menu [10, 46], combining two selection methods (dwell time and selection borders [82]) and comparing performance on character-by-character text entry, bigram text entry, and bigram combined with word prediction text entry.

5.2. Related Work

Since the early days of video based eye tracking, eye-typing has been a topic of active research [55]. Nowadays, text entry research has developed far beyond the classic on-screen QWERTY keyboard selected with the static dwell time method. Bee and André [7] distinguished three types of writing: typing, gesturing, and continuous writing. We describe the most relevant research done in text entry by gaze and have thereby related the systems and approaches to these three groups.

5.2.1. Eye-Typing

With eye-typing, letters are selected via dwell time from an on-screen keyboard. A virtual keyboard with QWERTY layout is one of the most common user interfaces for direct eye-typing. Majaranta and colleagues [52] presented such a keyboard for their research on (visual and audial) feedback for gaze-based interaction. They employed dwell time selection for typing characters, which was set to 600 ms. They discovered that the combination of click and visual feedback achieved the fastest typing rate, with a mean of 7.55 words per minute.

Contrary to direct eye-typing, multi-tap approaches require more than one selection per character. *GazeTalk* [26] (see Figure 33) is a multi-tap text entry system that provides three interaction modi:

- *Alphabetical letter-entry mode:* presents groups of letters (e.g. "ABCDEFG"). Here, letters are ordered into groups. Users need to select first the group which contains the letter and then the letter itself.

- *Primary letter-entry mode:* this mode supplies a dynamic keyboard that contains six buttons arranged in a three by two matrix that allows the user to type the six more likely letters (generated by a letter prediction algorithm). If the desired letter is not among them, the user needs to get back to the alphabetical letter-entry mode.

- *Word completion/prediction mode:* presents the current eight more likely words in a four by two matrix, as well as two buttons for changing to the other mentioned modes.

With *GazeTalk* users were able to type up to 6.22 wpm using a dwell time of 500 ms.

Miniotas and colleagues [57] developed a text entry system that required less physical space. *Multi-tap* supplies only ten keys, eight of them for text entry, one for space,

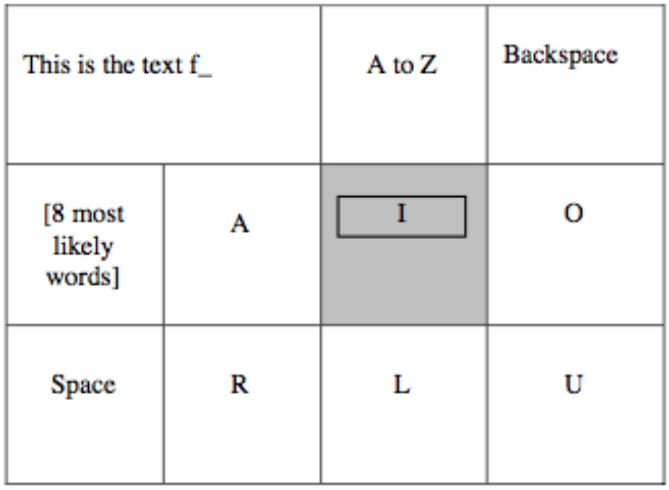

Figure 33: GazeTalk, on primary letter-entry mode [26].

dot and comma, and the last one for switching between different keyboards (function, letters, numbers and signs). Characters were ordered in groups in the same way as in ordinary mobile phones. To select a letter, the user needs to glance towards the key that contains the letter and remain over the letter for 400 ms, at which point the first letter is enlarged and changes color. To confirm its selection is required to leave the key. If the user continues looking at the key for another 400 ms, the second letter of the group will be highlighted, and so forth. This selection method means that the dwell time for the first letter of the key is 400 ms, 800 ms for the second, 1200 ms for the third, and in the worst case, at least 1600 ms for the forth letter. Nevertheless, Miniotas and colleagues [57] reported text entry speeds using Multi-tap of 8.94 wpm.

5.2.2. Eye Gesturing

Under eye gesturing systems are grouped text entry methods where the user needs to look at different locations to trigger a letter or an action.

With *Eye-S* Porta and Turnia [69] described nine "hot spots" on the screen, which the user can dwell on (for 400 ms) to describe a gesture. A specific gesture (selection order) is needed to perform for a desired action. The gestures contain an average three to four hot spots and were inspired by the "graffiti" gestures used with palms organizers. Experienced users achieved a mean text entry speed of 6.8 wpm. Faster text entry speeds may not be possible with Eye-S.

With a similar approach, Wobbrock and colleagues [88] presented *EyeWrite*, which is an

adaptation for gaze interaction of their previous work on text entry for palms EdgeWrite. Here, the four corners and center of a separate window (400 x 400 pixel) are used to enter the gaze gestures (see Figure 34). For example, for writing a "T" the user needs to glance at the top-left, top-right and bottom-right corner of the window, finishing the gesture at the center. The points on the corners do not require any dwell time for activation, however the center point is triggered by a short dwell time to confirm the end of each gesture. Even though almost no dwell times are used, users achieved a mean text entry speed of only 4.87 wpm.

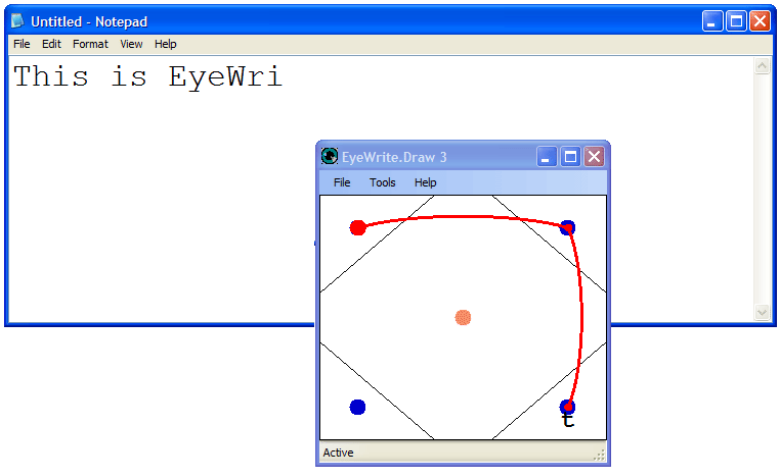

Figure 34: Text entry with EyeWrite. The user is entering the letter "T" [88].

Bee and André [7] adapted *Quickwriting* [67], which is also a text entry system for palm devices, for gaze interaction mapping the selection techniques for natural eye movements. Its interface is divided into eight selection sections. In the center is located a resting area, where eight group of letters are presented. The letters are ordered so, that their position corresponds to their position in the selection sections. In order to select a letter the user needs to glance to the corresponding selection section and go back to the center through the selection section where the letter is located. This text entry system works without any dwell time. In a first study participants achieved 5 wpm.

The the reason for Eye-S and EyeWrite's poor performance is nearly the same, namely the large number of spots that each gesture comprehends. With Eye-S an explicit dwell time restricts the performance, whereas with EyeWrite the physical constrains of eye movements limit the speed of the user's performance. Every saccade (a ballistic eye movement which last between 30-120 ms) is followed by a fixation, which lasts on average 200 ms before a new saccade can be started [39]. This means that it takes at least one second to complete the gesture of a letter.

Urbina and Huckauf [81] developed three dwell time-free text entry systems. The one

that performed best, *pEYEwrite*, takes advantage of pie menus with two hierarchical levels. On the first level the pie menu consists of six slices that contains groups of five letters or characters. On the second level each pie slice corresponds a letter or character. To take advantage of the biggest area of the slice, selection is done toward the outer border of the slice, the so-called selection border. To enter a character, the user moves their gaze so that it crosses the selection border of the slice that contains the desired letter. The pie menu on the second level opens immediately. The target letter is selected by glancing again through its respective selection border. The use of selection borders not only allows the user to inspect the pie as long he/she needs to, but also to perform fast multiple selections with saccades that follow each other. In special cases, where the target letter is located in the same direction as the group slice (like in letters "S", "R", "E", "I", "N" and space), the user is able to directly select with one long saccade that crosses the selection border of the pie in the first and second level (see Figure 35). Huckauf and Urbina [32] reported novice users achieving a mean text entry speed of 7.85 wpm while an expert achieved 12.33 wpm.

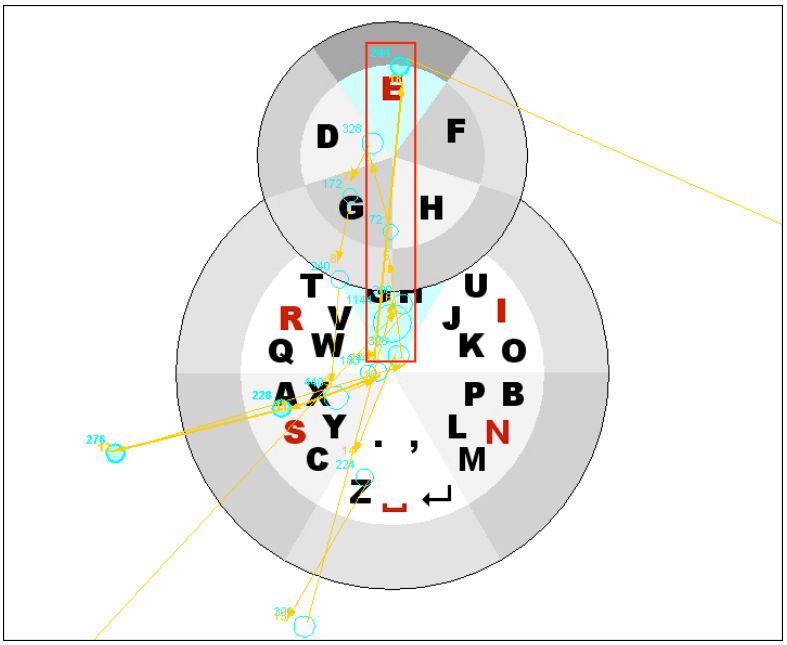

Figure 35: pEYEwirte. Selection of letter "E" with one saccade.

5.2.3. Continuous Writing

With continuous writing there are grouped text entry systems with which the eyes move continuously. Here, moving characters are followed by the user's gaze with pursuit movements, triggering their selection.

In *Dasher* [86], letters are presented vertically in alphabetical order on the right side of the screen. In order to select a letter, the user points (gazes) to the desired letter. The interface zooms in, and the letter moves towards the centre. The letter is selected when it crosses a vertical line located in the middle of the window. Gazing to the right from the vertical line leads to a selection of the target. Gazing left leads to the deletion of written text, causing the opposite visual effect. The distance between the gazing point and the vertical line controls the writing and deleting speed. This means looking to the middle of the vertical line ceases any action. The cooperation of vertical navigation and horizontal selection leads to a smooth motion of the letters. Considering the difference of text entry speeds of 4.5 wpm without text prediction [81] and around 20 wpm [79] with word prediction, it is clear that a character or word calculation algorithm is needed to make gaze based text entry more fluent.

StarGazer [25] is like Dasher, a zooming interface. However, the zooming occurs along the z-axis, while the x- and y-axis are used for panning. The letters are ordered alphabetically in circular form. The display pans towards the direction of the gaze while zooming is continuous, comparable with skydiving. Letters are selected when the desired letter has been zoomed enough. After selection, all letters appear again in the circular order. Hansen and colleagues [25] reported text entry speeds up to 8.16 wpm.

5.3. Character and Word Prediction

As mentioned above, character-by-character-based text entry is a slow process which requires multiple selections. Due to dwelling times and physical restriction by eye movements, currently text entry rates are far below 20 wpm. Selection methods which do not require dwelling times, are mostly based on an at least to-step selection method and are scarcely faster than dwell time based systems.

Character and word prediction can contribute to reach faster text entry rates, by reducing to character-searching-time with a character prediction method or reducing the kspc rate with word prediction.

Mackenzie and Zhang [50] proposed a gaze-based text entry system, which combined letter and word prediction. The goal of character prediction was to reduce the character searching time by highlighting the most probable three characters and enhance the fixation algorithm, describing a *tolerantDrift* radius, where the next probable letter (within this radius) is selected and not necessarily the letter under the point of fixation. Moreover, Mackenzie and Zhang used word prediction to reduce the number of keystrokes per character and increase the text entry rate, presenting the five most probable word candidates. They conclude that letter prediction to be as good (or even better) as word prediction, specially on interfaces with unfamiliar layout.

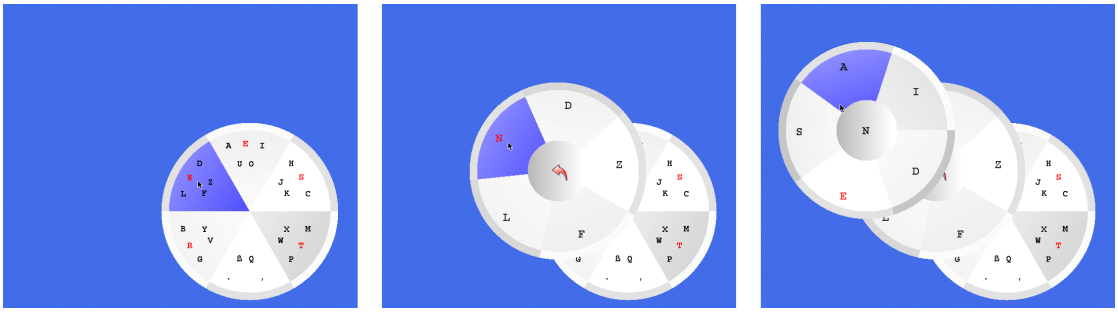

Figure 36: a) The first level of the hierarchical pie menu contains the group of characters. b) Immediately after selection, the second level pops up providing each letter for selection. On the center of the pie a back arrow can be dwelled for canceling the group selection. c) On the bigram level, five bigram characters are presented. Depending on the bigram entry mode these can be vowels, letters taken from a static bigram-probability table or dynamic calculated.

Johansen and colleagues [41] reported dynamic character layout (based on letter prediction) to be confusing and time consuming, since the participants need to search iteratively or the next intended letter. They proposed (for *GazeTalk*) the use of a "home position" for each letter, in this way letter could be found in the same cell reducing the cognitive load.

Trnka and colleagues [78] compared two different word prediction methods with letter-by-letter text entry. Their results showed the word prediction increases the text entry rates. They pointed out that the results are highly related to the quality of the word prediction and the bias of prior word prediction on the user. If, for example, the users could not find the intended word within the prediction list, they tend to ignore the upcoming word candidates.

5.4. Text entry Methods

For the longitudinal study seven different versions of *pEYEwrite* were implemented. A classic version with *single character entry*, three *bigram entry versions*, and three that combined *bigram text entry with word prediction*.

5.4.1. Single Character Entry, pEYEwrite-C

The single character entry version, *pEYEwrite-C*, is based on the aforementioned pEYEwrite [81]. Compared with [81], letters were re-arranged within the pie menu in order to facil-

itate the selection of the most frequent letters: The five most frequent letters (according to [68]), except for "I", which was replaced with "T", and the space-symbol were arranged in a way that they can be typed with one saccade (see Figures 36a and 36b). The other letters were distributed within the slices, so that the upper slice contained only vowels and the remaining letters were grouped with their neighbors in frequent words (like "ch", "br", "gr", "ck" or "nd", see Figure 36a). The letter position in the first pie equaled the letter position in the second pie. This way users know the position of the letter in the second pie in advance and are able to prepare the saccade responsible for the letter selection before the second pie pops up.

In order to type one letter using *pEYEwrite-C*, the respective slice of the first pie has to be selected. Immediately following, a next pie is opened, centered on the border of the corresponding pie (see Figure 36b). The desired letter has to be selected from that pie. Letters could be selected via selection border (crossing the outer border of the slice with the gaze) or via dwell time (600 ms). The users were free to combine both selection methods.

A cancel button is provided on the center of the pie in the second hierarchical level (see Figure 36b) to allow the user to abort the selection process initiated in the first pie. With this, even when a group of letters is already selected, no letter is written. The cancel button is triggered by a dwell time of 750 ms.

One of the main strengths of *pEYEwrite-C* is that novices can easily learn, in an unintended way, the gaze path required to select each letter. Therefore advanced users are able to perform the two-stroke gaze gesture even without looking for the letters, increasing their performance.

5.4.2. Text Entry with Bigrams

With bigram text entry the user is able to enter, in an optimal case, two letters (or characters) traversing the hierarchical pies only once. To achieve this, the *pEYEwrite-C* interface was extended with a third hierarchical level: the letter group on the first level, the single letters on the second level and up to five following letter candidates on the third level (see Figure 36c). Three different bigram entry approaches were conceived and tested. The first approach provides vowels on the bigram level after each letter, called *pEYEwrite-BiVow*. Presenting vowels as bigram candidates not only provides a high probability for using a vowel (of 40% [68], but also allows an easy extension of the selection gestures with a further step. Since the vowels are presented in alphabetical order and always have the same position within the pie, users should be able to learn their positions and find them quickly and accurately after little training with the fast three-stroke gaze gesture for selection.

The second approach, called *pEYEwrite-BiSta*, presents the five most probable letters at the bigram level that follows the letter entered in the second level, which is based on a static bigram statistic (available on [68]). This method increases the probability of a correct bigram candidate up to 70%, but at the same time increases the cognitive load of the user. The user has to learn 25 (letters) * 5 (bigram candidates) = 125 possible gaze paths to be able to enter a bigram with this method, which might require extensive training before the user is able to perform a fast three-stroke gaze gesture.

The third and last approach, *pEYEwrite-BiDyn*, relies on an "dynamic" calculation of the up to five most likely letters to follow, based on a list of 300,000 word-trigrams and their occurrence in a corpus. The corpus was automatically extracted from information available online, like newspapers and e-books. The character prediction algorithm takes in to count the last two written words in the calculation of the letter candidates, increasing the probability of a suitable letter among the candidates, compared with *pEYEwrite-BiVow* and *pEYEwrite-BiSta*. Due to the dynamic calculation, it is impossible for the user to learn which letters will be presented in the bigram level of the pie, increasing the cognitive load of the user even more and restricting a fast gaze gesture usage.

As in *pEYEwrite-C*, letters are selected via selection borders or dwell time. Selecting via selection borders implies that the user can select a letter and continue traversing the pie to the next hierarchical level. Selecting with dwell time implies selecting a letter but not to continue traversing the pie. Once a letter is selected on the last level or by dwell time, the pie menu collapses and the user starts again from the start button (see Figure 37).

5.4.3. Text Entry with Bigrams and Word Prediction

While enhancing text entry with word prediction seems to be a plausible way to increase the text entry speed and reducing the keystroke per character (kspc) rate, relieving the user. The real benefit that word prediction provides is still unclear. The presentation of predicted word candidates draws the attention of the user from the letter entry to the word candidates. The user then needs to read the predicted words and, if possible, select one of them. Moreover, the more letters that are entered, the better the predicted candidates, drastically reducing the number of letters that do not need to be typed by selecting a word candidate. Considering these two facts makes our a priori assumption about the benefits of word prediction doubtful.

In order to provide empirical evidence and conclude how text entry is affected by word prediction, we included the three already-presented bigram text entry word predictions, calling them *pEYEwrite-WoVow*, *pEYEwrite-WoSta* and *pEYEwrite-WoDyn* respectively. Up to three word candidates (according to the prediction) were presented above

the text field, which are selected by dwelling the gaze for one second on them (see Figure 37).

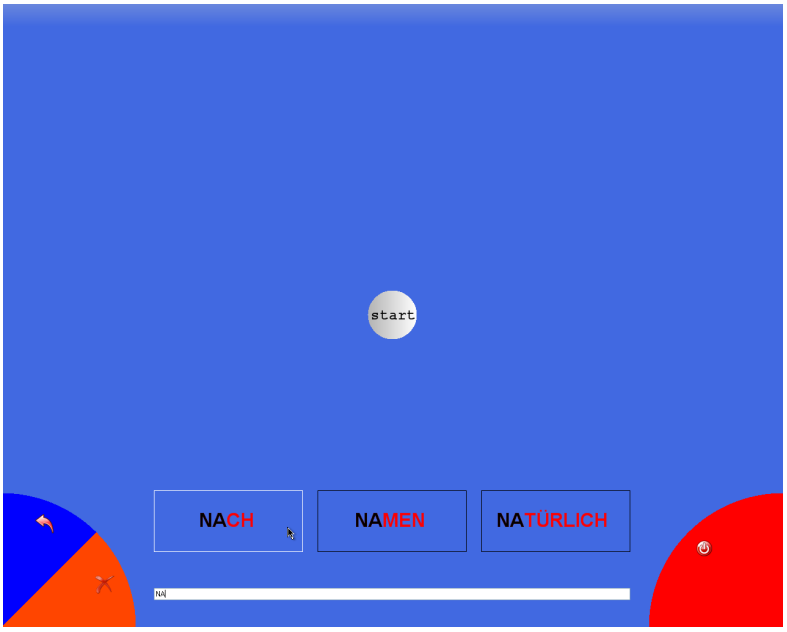

Figure 37: Word candidates are presented above the text field. The characters already entered are highlighted in black, the characters to enter in red.

It is important to note that, as expected, the quality of the prediction algorithm as well as of the corpus used has a significant influence on the predicted words and it is not the primary aim of this work to provide a linguistic masterwork. The prediction algorithm is, based on a Markov model of second order, also known as trigrams. The list of trigrams and their frequency in the corpus is the same used in *pEYEwrite-BiDyn*. Here the next word probability is calculated in relation to the two words written before.

5.5. Longitudinal User Study

The aim of this longitudinal study is to reveal the potential of pie menus for text entry by gaze and to provide and compare alternatives to single character entry. For this reason efficiency and errors were measured. Efficiency was measured in words per minute; wpm. A word is defined as a sequence of five characters, including space. Errors were measured with two variables, corrected errors and remaining errors. Corrected errors present the rate (in percent) between deleted and undone sections in relation to all key selections. Remaining errors are evaluated using the "Minimum String Distance" method (MSD) [72], which looks for the minimum distance of two strings in terms of the editing primitives, insertion, deletion and substitution, and is calculated as follows:

RE = MSD / Max(length correct text, length written text)

Furthermore, we measured the mean number of keystrokes needed to enter a letter. Text entry systems that do not use word completion ideally reach a keystroke per character (kspc) rate of 1.0 indicating that each keystroke produces a character. Therefore, deleting and undoing selections increases the kspc rate. Typing systems using word prediction/completion may reach kspc rates below 1.0, since more than one character may be entered per keystroke.

5.5.1. Method

Participants Nine university students participated voluntarily in the experiment. Their ages ranged from 24 to 30 (mean 27.4), they had normal or corrected-to-normal vision and were familiar with computers and with mouse and keyboard usage. Six participants had no prior experience with eye tracking, and the remaining three had similar experience with eye tracking and eye typing. All participants were rewarded. To keep participants motivated during the experiment, the participant who achieved the best typing performance received a prize.

Apparatus The interfaces were presented on a 21" Sony GMD-F520 CRT display with a resolution of 1280x1024 at a frame rate of 100Hz, and were run on a Dell-PC with a Core2Quad processor at 2,5 GHz and 2 Gb DDR-Ram under Windows XP. The eye tracking device used was a head-mounted Eyelink2 from SR-Research. The tracked gaze coordinates were mapped directly onto the mouse cursor, without using any smoothing algorithms. In order to keep calibration as accurate as possible, a chin rest situated 60 cm in front of the monitor was used.

Task and Procedure Nine participants were divided into three groups (by three participants). The participants with prior eye tracking experience prior were divided up, with one placed in each of the three groups. The rest of the participants were assigned randomly to each group. Each participant absolved altogether 60 sessions, 20 sessions with *pEYEwrite-C*, 20 sessions with one of the bigram-entry versions of pEYEwrite (*pEYEwrite-BiVow*, *pEYEwrite-BiSta* or *pEYEwrite-BiDyn*) and 20 sessions with the corresponding bigram-entry with word prediction (*pEYEwrite-WoVow*, *pEYEwrite-WoSta* or *pEYEwrite-WoDyn*).

Participants sat 60 cm in front of the monitor, and they were helped to mount the eye tracker to their head before the calibration routine was started. On the beginning of each test day and before testing a new interface, the subjects had five minutes training. The

participants did not type more than one hour and performed maximal five sessions per day. A session consisted of writing three well-known sayings (which had a mean length of six words per saying). Participants were instructed to enter the saying as fast and accurate as possible. Typographical errors were to be corrected if noticed immediately.

5.5.2. Results

Single Character Entry - pEYEwrite-C With *pEYEwrite-C* users reached a mean text entry speed over the 20 sessions of 7.34 wpm (standard error se = 0.35) and 8.1 wpm for the last three trials (se = 0.37). Due the significant learning rate and differences over sessions, we will refer the mean values as the average performance of the last three sessions. The maximum entry speed of this device was 10.38 wpm, accomplished by subject number seven on the eighth session. The sessions had, as expected, a significant effect ($F(19,152) = 13.42$, $p < 0.001$) showing a strong learn rate (see Figure 38). Post hoc tests showed that even the difference between the entry speed between the 10th and the 20th session was significant ($F(1,8) = 8.73$, $p < 0.05$).

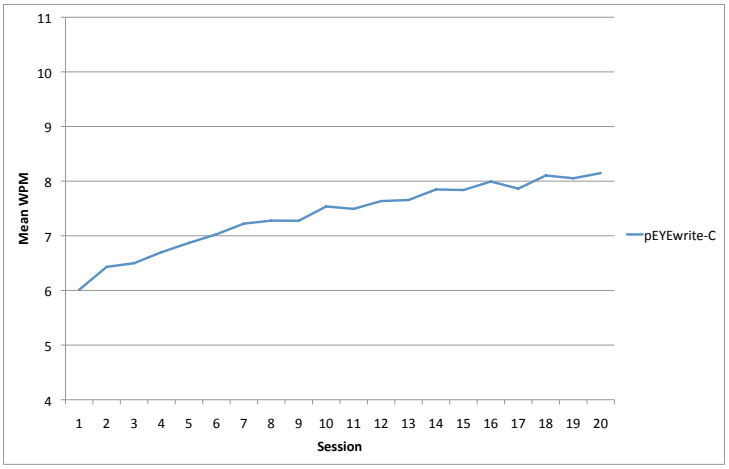

Figure 38: Word per minutes achieved with pEYEwrite-C on each session.

Users committed on average 1.41 % errors (se = 0.367) that remained in text. Sessions showed a significant effect over the error rate ($F(19,152) = 3.33$, $p < 0.001$), which ranged form 6.55% (se = 1.95) in the first session to 0.97 % on the last session (see Figure 39). Eight (of nine) subjects performed at least one session without making errors.

The mean corrected error rate produced was of 3.95% (se = 0.72). The sessions showed a significant effect, with $F(19,152) = 2.11$, $p < 0.01$, improving the error rate from 9.85% (se = 1.39) corrected text in the first session to 3.6 % (se = 0.82) in the last session.

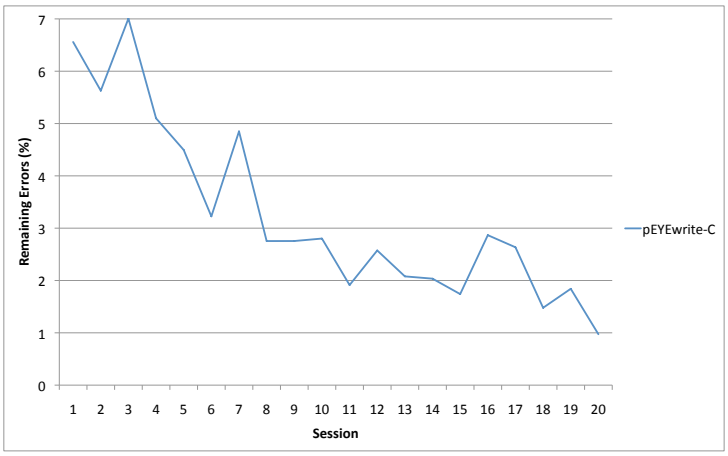

Figure 39: Rate of errors remaining in text with pEYEwrite-C on each session.

The mean kspc rate was 1.18 (se = 0.04). The sessions also showed here a significant effect, with $F(19,114) = 7.192$, $p < 0.001$, reducing the kspc rate from 1.78 (se = 0.205) corrected text on the first session to 1.18 kspc (se = 0.03) on the last session (Figure 40).

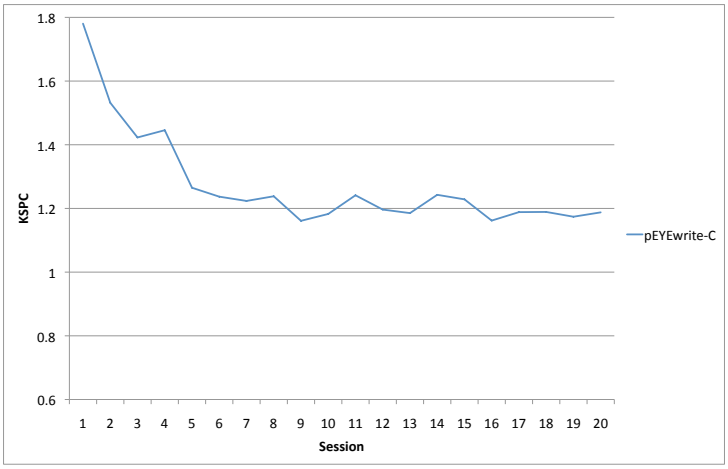

Figure 40: Mean keystroke per character rate achieved with pEYEwrite-C on each session.

Furthermore, we analyzed which selection method was preferred by the subjects. 98.57% of the characters were selected with the selection border method and only 1.43% by dwelling on the character (see Figure 42). Throughout the 20 sessions, users improved their text entry performance by 36.6% using selection borders (from 5.9 wpm in the first session to 8.06 wpm in the last session) and only by 4% (from 4.5 wpm to 4.71 wpm) (see Figure 41).

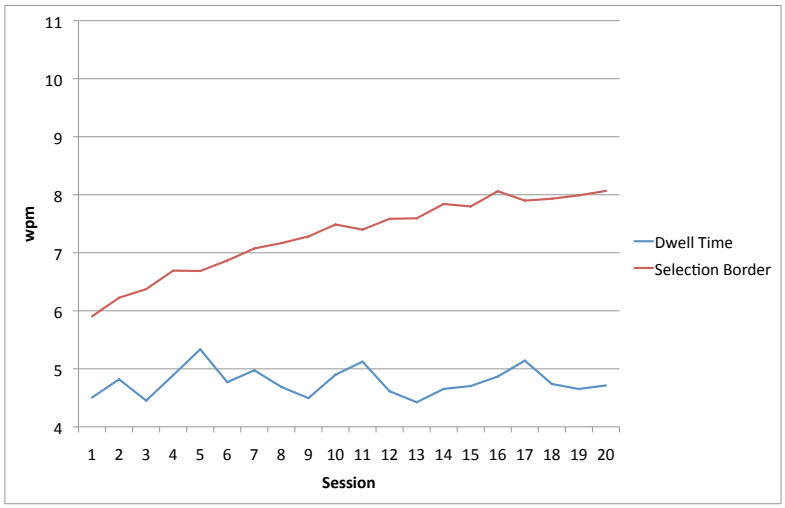

Figure 41: Word per minutes achieved with the pEYEwrite-C using selection borders and dwell time selection on each session.

Text entry with Bigrams We compared the text entry performance and learning rates of three (already mentioned) approaches based on hierarchical bigram entry, *pEYEwrite-BiVow*, *pEYEwrite-BiSta* and *pEYEwrite-BiDyn*. Subjects achieved a mean text entry speed on the last three sessions of 8.05 wpm (se = 0.83, max = 9.42 wpm) with *pEYE-write-BiVow*, 10.08 wpm (se = 0.83, max 12.43 wpm) with *pEYEwrite-BiSta* and 10.64 wpm (se = 0.83, max = 12.85 wpm) with *pEYEwrite-BiDyn*, showing no significant differences ($F(2,6) = 2.67$, $p > 0.05$). The sessions showed a significant effect on performance with $F(9,114) = 14.18$, $p < 0.001$ that can be attributed to learning as well as an interaction between the bigram methods and the sessions ($F(18,114) = 1.57$, $p < 0.05$). In mean, participants improved their text entry performance on 25.11% with *pEYEwrite-BiVow*, 29.87 % with *pEYEwrite-BiSta*, and 18.25 % with *pEYEwrite-BiDyn* (see Figure 43).

The rate of remaining errors was in mean on the last three sessions 1.35% (se = 0.38) with *pEYEwrite-BiVow*, 0.5% (se = 0.38) with *pEYEwrite-BiSta* and 0.96% (se = 0.38) with *pEYE-write-BiDyn*. Neither sessions nor the bigram entry methods produced significant effects. All participants were able to complete at least two error free sessions.

The corrected error rate was on average 2.79% (se = 1.24) with *pEYEwrite-BiVow*, 3.75% (se = 1.24) with *pEYEwrite-BiSta* and 4% (se = 1.24) with *pEYEwrite-BiDyn*, presenting no significant differences. The sessions showed a significant effect on performance with $F(9,114) = 2.115$, $p < 0.01$ which can be attributed to learning. All participants were able to complete at least two error free sessions.

Participants achieved a mean kspc rate of 1.07 (se = 0.33) with *pEYEwrite-BiVow*, 1.09

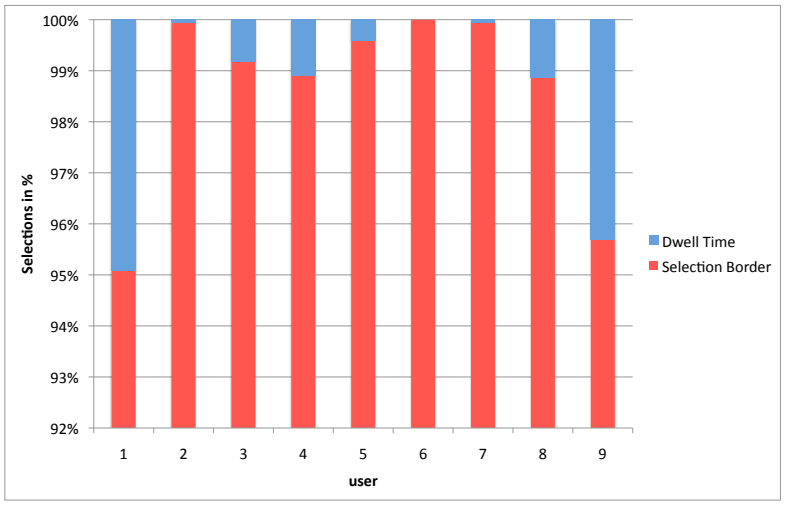

Figure 42: Rate of selections performed with the selection border and dwell time method.

(se = 0.33) with *pEYEwrite-BiSta* and 1.1 (se = 0.33) with *pEYEwrite-BiDyn*. Here, as well as in the remaining error data, no significant effects of sessions and bigram entry methods were observed. All participants were able to complete at least two error free sessions.

Text entry with Bigrams and Word Completion We measured text entry speeds combining bigram text entry with word completion. Participants achieved a mean text entry speed of 12.73 wpm (se = 1.21, max = 14.71 wpm) with *pEYEwrite-WoVow*, 12.72 wpm (se = 1.21, max 15.8 wpm) with *pEYEwrite-WoSta* and 13.47 wpm (se = 1.21, max = 17.26 wpm) with *pEYEwrite-WoDyn* (see Figure 46.). These differences were not of statistically significance and after the 20 sessions no significant learning effect was observed. We could observe differences between performance on trials with a high prediction rate (from session one to five and 17 to 20, where all words were under the first three candidates after entering maximal three characters.) and with low prediction rate (from session six to 16, where the participant need to enter in mean more than four characters to get the intended word predicted).

Participants produced 0.01% (se = 0.147) errors that remained in text using *pEYEwrite-WoVow*, 0.05% (se = 0. 147) with *pEYEwrite-WoSta* and 0.01% (se = 0.147) with *pEYEwrite-WoDyn*.

Participants achieved a rate of corrected errors of 3.27% (se = 1.46) with *pEYEwrite-WoVow*, 5.06% (se = 1.46) with *pEYE-write-WoSta* and 2.36% (se = 1.46) with *pEYEwrite-WoDyn*. Here, as well as in the remaining error data, no significant effects of sessions and bigram entry methods were observed. All participants were able to complete at least

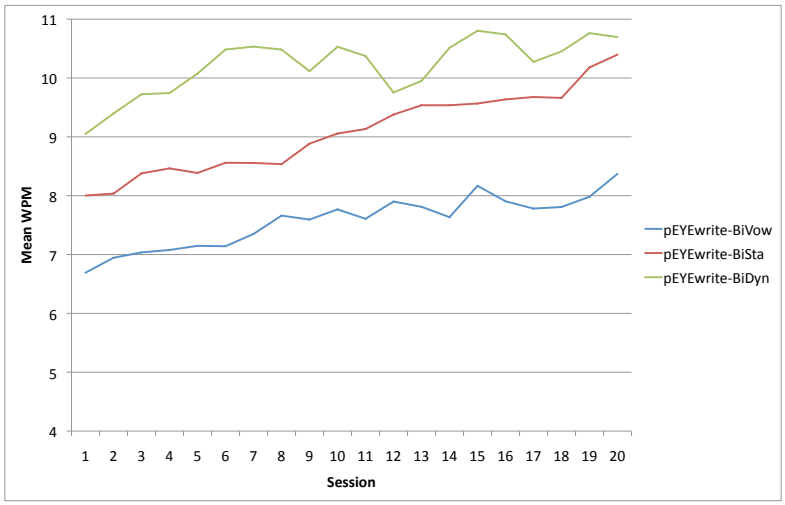

Figure 43: Word per minutes achieved with the bigram text entry methods on each session.

two error free sessions.

The mean kspc achieved by participants using *pEYEwrite-WoVow* reached 0.711 (se = 0.07), with *pEYEwrite-WoSta* 0.74 (se = 0.07) and 0.705 (se = 0.07) using *pEYEwrite-WoDyn* showing not significant differences between these three text entry modes. Sessions had a significant effect on performance with $F(9,114) = 9.186$, $p < 0.001$.

5.6. Discussion

The results obtained with the single character entry method support the results presented in the short experiments [32], showing significant learning effects in all measured aspects using pie menus. Taking the learning data into consideration, it can be assumed that the learning process had not finished even after 20 sessions. This highlights the outstanding qualities of pie menus in gaze interaction.

The selection method data results obtained using with *pEYEwrite-C* showed a massive preference for using selection borders instead of dwell time. The text entry speed improvement rate achieved with both selection methods (36% with Selection Borders, 4% with dwell time) and the text entry speed (8.06 wpm with Selection Borders and 4.71 with dwell time) help clarify the favoritism for selection borders. Subjective results reported that 8 of 9 participants reported preferring selecting characters with selection borders. Even though one participant prefered dwell time, no difference was revealed between his/her data and the averages already reported, selecting 94.45% of the characters

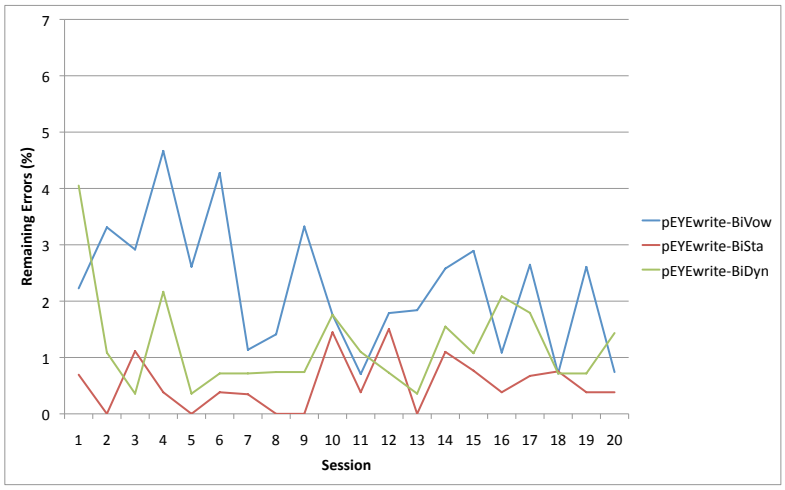

Figure 44: Rate of errors remaining in text with with the bigram text entry methods on each session.

with selection borders (see Figure 42, user 1).

Results depictured in Figure 43 confirm most of our learning hypothesis on bigram entry. *pEYEwrite-BiDyn* showed on the first sessions better performances (one wpm faster) than *pEYEwrite-BiSta*. This can be attributed to the more probable presentation of a suitable bigram candidate for selection. This gap was closed after the 20 sessions, confirming the better learning qualities of static bigrams (*pEYEwrite-BiSta*) over dynamic bigrams (*pEYEwrite-BiDyn*). Even though no significant effects were found between the bigram text entry methods, post-hoc analysis revealed a significant difference on text entry performance (p=0.05) between *pEYEwrite-BiVow* and *pEYEwrite-BiDyn*. This significant difference has mainly two reasons; First, the rate of bigram entries with both methods differed from 22.72% with *pEYEwrite-BiVow* to 35.45% with *pEYEwrite-BiDyn* (31.53% with *pEYEwrite-BiSta*). The second reason was the longer mean selection time achieved on the bigram level (third depth level of the pie menu) with *pEYEwrite-BiVow* (819 ms) compared to *pEYEwrite-BiDyn* (609.98 ms). This is an unexpected result, since the vowels were all presented on the same position, which should be easy to learn, while the dynamic predicted bigram characters had mostly a different combination and position. Huckauf and Urbina [32] reported serial scanning of characters ordered alphabetically, providing poorer performances than arbitrary ordered ones.

The main drawback observed using pEYEwrite combined with bigram entry was the repeated entry of double errors. Some participants selected the wrong first letter and thereby a wrong bigram on the next stage. Even when participants noticed the first error, they preferred to continue selecting the second letter with selection borders and not to abort by dwelling on the center of the pie, causing a double error. Therefore

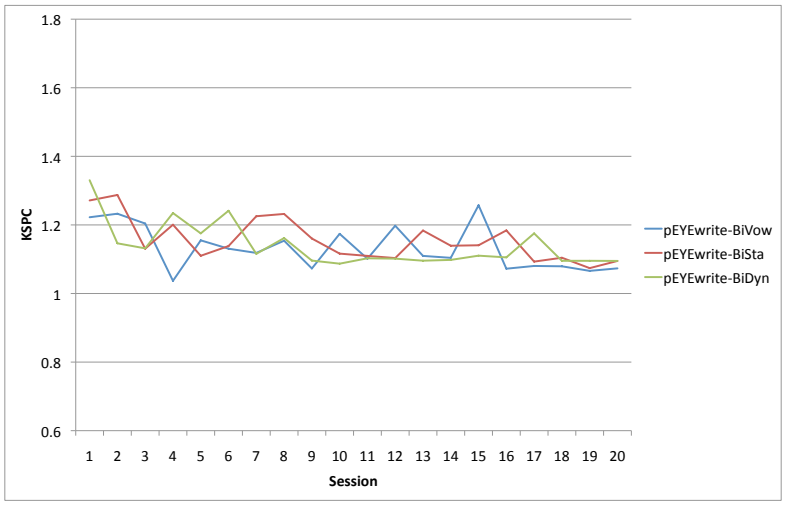

Figure 45: Mean keystroke per character rate achieved with bigram text entry methods on each session.

they needed to correct two errors, unnecessarily increasing the kspc and the corrected error rate. This can be solved applying an auto-correction algorithm, which can use the corpus and the entered letters to calculate which word the user intended to write.

Results show that by using word prediction the users achieved higher text entry rates than using only bigram entry. Reported speeds achieved in mean 13 wpm in mean (with a maximal 17.2 wpm) pointing out the importance of word prediction and completion for gaze based interaction. The text entry performance achieved did not differ among the three bigram entry systems with word prediction (*pEYEwrite-WoVow*, *pEYEwrite-WoSta* and *pEYEwrite-WoDyn*), showing a bigger impact of word prediction than each bigram presentation method. These results show that the assumed high cognitive load using word prediction does not hamper the text entry performance, which was much higher as with other text entry systems with word prediction [26, 25]. They also contradict the findings of MacKenzie and Zhang [50], who compared word and letter prediction in a gaze typing system. Based on their results, they concluded that letter prediction can be as good as word prediction, or even better in some cases. The benefits of letter and word prediction may differ depending on the complexity of the layout of the keyboard. One should also note that the quality of prediction has also an effect; if the prediction is accurate, people trust it and use it more [78]. Due order effects we omitted an analysis of variance between single character entry, bigram entry and bigram with word completion. Nevertheless, the typing speed, error and kspc rates presented in this work point out the importance of alternatives to single character entry, like bigram entry and word prediction in eye-typing.

Figure 46: Word per minutes achieved with the bigram text entry methods combined with word prediction on each session.

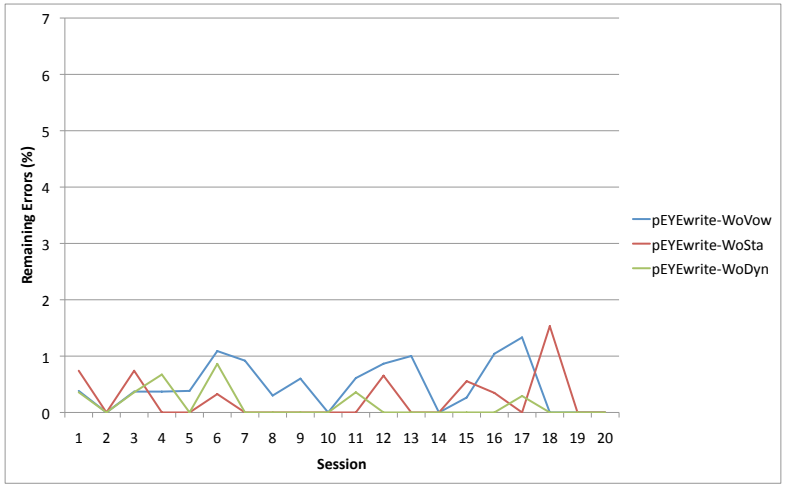

Figure 47: Rate of errors remaining in text with with the bigram text entry methods combined with word prediction on each session.

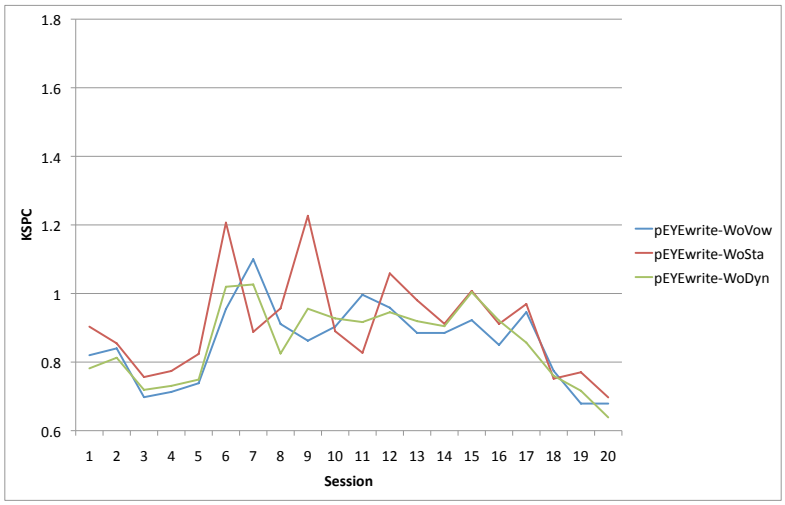

Figure 48: Mean keystroke per character rate achieved with bigram text entry methods combined with word prediction on each session.

5.7. Conclusion

Three bigram text entry methods were presented, all based on a hierarchical pie menu application. The methods based on the probability of the next character (calculated statically or dynamically) performed better than presenting vowels as bigram candidates. Augmenting the investigated text entry methods with word prediction increases the text entry performance and decreases the cognitive load during typing. The combination of two selection methods was well accepted by the users and enhanced the interaction with pie menus. While participants could not achieve text entry speeds of around 20 wpm as reported by [79], the mean of about 13 wpm (with a top speed of 17.2 wpm) achieved by combining different selection methods with bigram entry and word prediction provides an alternative to Dasher for fast text entry and encourages the continuation of research on text entry by gaze.

6. HMDs at a glance - Perceptual Issues on Visual Perception on HMDs for Mobile Gaze Input

This chapter is based on the publication:

Huckauf, A., Urbina, M. H, Böckelmann, I., Schega, L., Doil, F., Mecke, R. & Tümler, J. (2010). Perceptual Issues in Optical-See-Through Displays. In *Proceedings of Applied Perception in Graphics and Visualization*, Los Angeles, U.S.A, 41-48.

6.1. Introduction

Mobile computing is revolutionizing the interaction between humans and computers. Nowadays, small sized computers, like ultra-mobile-pc, pads or smart-phones, have become very popular. They allow ubiquitous access to information, online services, multimedia and thousands of applications, making their users almost dependent on these devices. The evolution of mobile computers does not stops here. The use of Head-Mounted-Displays (HMDs) brings mobile computing to a next level, since the user wears the display comfortably on his head, he does not need to hold the display with his hands, having them free for other tasks.

HMDs allow technologies referred to as Augmented Reality (AR) to present an observer with information additional to the information already present in the real environment [20]. Hence, this additional information can be perceived while working on a real object and without releasing attention from it.

HMDs are already used in applications like entertainment[3], medicine [8], maintenance [65], training or commissioning [70]. For instance, a commissioner can be guided from item to item through the optimal path and can get information additional information, like the number of items to be collected [22, 80]. In an other application scenario, technicians can receive valuable information while assembling wire harness bundles on an aircraft [60]. The idea is that instructions as well as other information can be presented without disturbing the current work processes while the user can keep his hands free to complete their tasks. This last fact has motivated the research for an intuitive and non intrusive interaction method, leading to gaze based interaction.

[3]For example see Nikon's Media port up300x at http://www.uplab.jp/, last visited on 27.06.2011.

6.2. Mobile Gaze Based Interaction

Graupner and colleagues [21] studied systematically the key aspects of gaze based interaction for mobile devices. They developed a mock-up system consisting on a optical see-through HMD and an head mounted eye tracker to explore the subject's performance on pointing and selection task. Among several variables, they evaluated the size and position of interactive buttons analyzing selection times and hit rates. Graupner and colleagues found that button sizes of smaller than 2° of visual angle result in hit rates below 90%, which is comparable to earlier findings [87] but below the performance reported using stationary screens [30]. Selection performance was better for central buttons than for those in the periphery of the HMD. Buttons in the upper half of the HMD were more difficult to select than in the lower half. Graupner and colleagues assigned these findings to limitations of their experimental setup.

Agustin and Hansen [1] investigated the possibilities of building a low-cost eye tracking system for mobile HMDs. They combined a conventional web camera with infrared LEDs with a binocular non see through HMD. After preliminary tests they reported an average accuracy under 1° when standing still and around 2.5° when walking. Since they did not fixed the camera to the HMD, the accuracy was affected by relative movements between the HMD and the camera while walking. However, Agustin and Hansen [1] pointed out, that the achieved accuracy was high enough to interact with noise-tolerant interfaces designed for gaze typing, like *GazeTalk* [26] or *Stargazer* [25]. Furthermore, while standing it was possible to interact with a normal windows environment using standard gaze-selecting techniques, like dwelling, zooming or two-step magnification.

Nilsson and colleagues [64] developed a head-mounted video-see-through (HMD-VST) system with an integrated eye tracker. Their main goal was to make the interaction between the HMD-VST and the user easy and efficient. A further goal was find a method to predict the user's intentions in order to anticipate what actions may be requested to the system. To test their system, Nilsson and colleagues developed an instructional application, where the user needed to complete a set of instructions. The user received the next instruction after acknowledging to the system that the previous instruction has been completed. The instructions were given to the user as statements and questions that had to be confirmed or denied vie gaze interaction. The user could choose to look at 'yes', 'no', and 'acknowledged'. The answers were selected after dwelling for one second. Nilsson et al. reported gaze controlled interaction to be equally fast as pressing an ordinary keyboard button. This is in accordance with earlier research and the results of Ware and Mikaelian [87] who illustrated that gaze interaction may even be faster since the time it takes to shift position of the cursor manually slows down the speed of interaction in traditional mouse pointing tasks. Nilsson and colleagues conclude that their system could be an alternative to manual selection for AR environments, especially when hands or speech are not available for input. Furthermore they propose to use gaze direction to de-clutter the field of view on marker based AR environments, revealing the

information of the markers which are viewed by the user, and not of all markers in the field of view of the scene camera.

All these results prove that it is possible to interact with HMDs using eye movements but at the same time open the question, if it is possible to transfer all the scientific findings of human-computer gaze-based interaction to head mounted devices. Most of the knowledge in this field has been acquired using conventional display techniques, like CRTs, TFTs, LCDs or even projection panes, where the user can not see through the pane where the information is presented. In order to provide an answer to this question (or at least me able to make an assumption) we need to understand the visual perception precesses behind head mounted display techniques. Therefore, in the next section we took a deeper look to HMDs, particularly optical see though devices.

6.3. Understanding Head-Mounted Augmented Reality Displays

As already mentioned, AR technologies refers to the view of a real world environment which is augmented by additional visual information generated by a computer [20]. A prominent method to present augmented reality information is by using head mounted displays. Among head-mounted displays there are two main technologies. Video see through (VST) devices embed virtual information directly into the rendered real world image [5]. Thus, the spatial alignment of virtual and real information can be guaranteed. However, since the head-mounted camera and the viewing position are not exactly the same, eye-hand-coordination processes are disturbed when using VST-devices [9]. In addition, there is at least a small latency between changes in the real world and their perception which also interferes with motor coordination. In the other technological method, the optical see through (OST)-devices, information is provided on a semi-transparent mirror. Hence, the perception of the real world is only marginally impaired by the frames of the head-mounted device. Therefore, the OST-devices are promising candidates for industrial applications. However, the OST-technology requires a large spatial tracking precision in order to exactly align the virtual and the real information. In first studies during the application of head-mounted OST-devices, mental strain especially for the visual system was observed [19]. The reported symptoms ranged from eye strain to headache [80]. Thus, the question about the origin of these impairments arises.

6.4. Characteristics of OST-Devices

The goal of the AR-technology lies in the enrichment of the real world with virtual information. The underlying idea is that the virtual information is perceptually integrated in the real surrounding. From a psychological viewpoint, on the one hand, this idea seems plausible: Objects appearing in spatio-temporal neighborhood become associated. On

the other hand, however, a complete integration can only be achieved when the presentation is perceived as unitary. However, there are at least three characteristics of the virtual information suggesting that integration will be hard to achieve:

1. the fact that virtual information is self-illuminating,

2. the inhomogeneous background which affects the information extraction on AR-systems [48],

3. the fact that changes in object size with changing observing distance provide counter-intuitive depth cues,

Information presented via an OST-device is self-illuminating. This is a problem since such a feature is rarely observed in real life objects. Thus, a complete integration of both, virtual and real information, should only be observed if the real world consists of self-illuminating objects. But, this is the case mainly for the sun and for objects presented on a computer screen. For all other objects, already the luminance of the objects should serve as differencing characteristic.

Visual recognition of stimuli strongly depends on the features of the background such as brightness, contrast, size of a pattern, and so on. Hence, adjusting an optimal position in the OST-device for displaying the virtual information depends on the distribution of such features on the background. Clearly, this varies in an unpredictable manner.

Another important feature of OST-devices is the information regarding the size of objects. The size usually provides one important cue for the estimation of distances [16]. In our real world, an object increasing in distance from the observer is projected on smaller retinal areas and thus becomes smaller also in perception. In OST-devices, objects are presented with a constant size onto the retina. When an observer increases the distance to the viewing background (e.g., a wall on which the virtual information is localized), the retinal area covering the projected image does not change in size. Hence, the same object has to be interpreted by the perceptual system as either closer or larger, or both. As a consequence, since virtual objects are typically localized in front of or on the next background, the size information cannot be used to compute distances. Moreover, the size of the virtual objects sometimes provides even misleading depth information.

The aim of the present studies was a careful investigation of the simultaneous perception of virtual and real world objects and thus, the perceptual capability to integrate virtual and real information. Therefore, three experimental set-ups were developed in order to investigate certain parts of the integration process. These set-ups all share the following characteristics: As real world information, stimuli were presented on a computer screen. This was done in order to reduce the effects of the self-illumination of the information of the OST-device. In addition, observers used a chin-rest in order to keep the viewing

distance constant. Hence, changes in viewing distance, which lead - as explained above - to distortions in size and depth perception [75], can be excluded. Moreover, the background is relatively constant throughout the experiment. It can thus be expected that the virtual and the real information can indeed be integrated under these artificial but optimal viewing conditions. The question at issue is thus, whether recognition performance when scanning for information simultaneously on both devices (OST-HMD and screen) is lower than performance when scanning only a single device (either the OST-HMD or the screen).

In Experiment 1, the set-up was a visual search task in which observers have to decide whether a pre-defined target had been presented in the image or not. The image was then presented either on one medium (the screen or the OST-device) or, the image was split and presented on both. With this account, we can quantify the costs of switching the display medium during fluent work. In Experiment 2, a second task was added to the visual search task. This task required a reaction whenever a rare stimulus appeared. This rare stimulus was given priority in the reactions. Hence, with this task, we aimed at examining situations in which the visual attention is already maintained on one medium. In analogue application situations, the important question is how well observers can perceive important stimuli on one medium even when they focus on the other one. In Experiment 3, vergence eye movements were measured while fixating a stimulus presented on either medium.

6.5. Experiment 1: Visual Search

6.5.1. Research Question

In order to examine how well visual information presented on a computer screen and on an OST-device can be integrated, a visual search task was applied. Observers had to decide whether a target (the digit 0) was present in a matrix of 6 * 6 characters (the letters O). In two conditions, all stimuli were presented either on the screen or on the OST-device. In two further conditions, (the left) half of the stimuli were presented on the screen, and the other (right) half on the OST-device and vice versa. The distribution in left and right was used to avoid overlapping stimuli: Small errors in the spatial alignment between both media which can already happen due to slight changes in the head-position do not result in overlapping images.

6.5.2. Methods

Stimuli As distracter, the letter O was used. As target, the number 0 was presented, both using the Courier New font by size of 25 pts. Stimuli were presented within a 6 x 6 matrix (see Figure 49) with a vertical spacing of 66 pixels and a horizontal spacing of 100 pixels. Targets could appear only in the inner 4 * 4 matrix. Due to the limitations of the OST-device which can only present stimuli in red, the color of stimuli displayed on the computer screen was set to the most comparable orange-red. The background of the computer screen was always black.

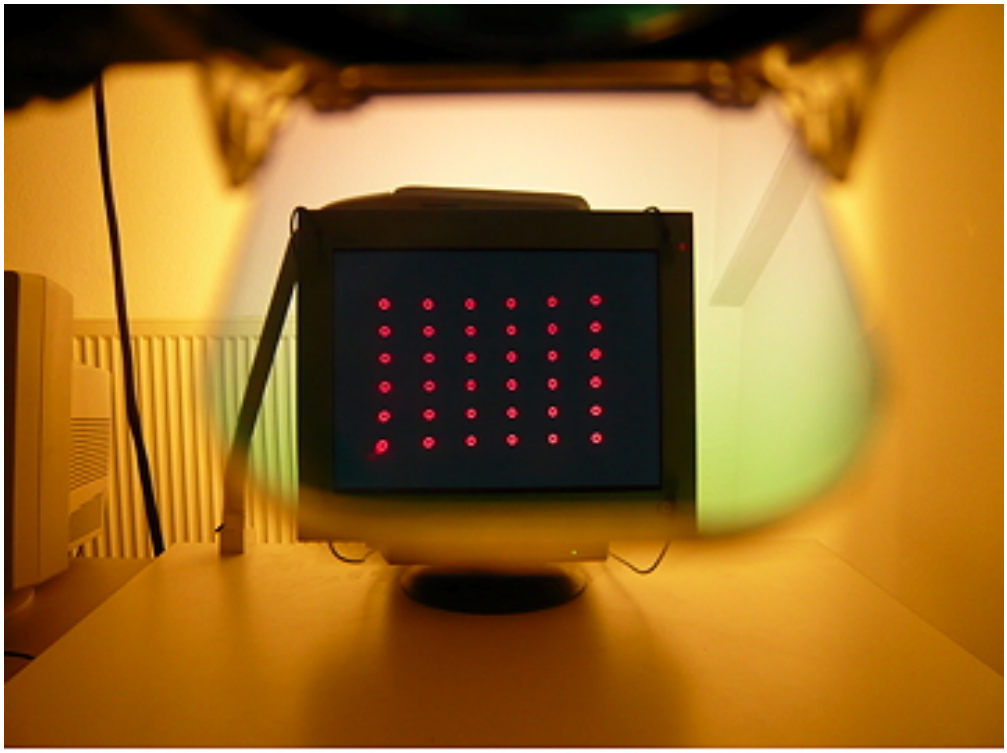

Figure 49: View through the OST-decvice on the computer screen where the 6*6 matrix is presented in red on black background.

Apparatus For presenting the stimuli, two devices were used: a 21" Sony GDM-F520 CRT-display and a monocular Microvision Nomad ND 2100 Optical-See-Through Head-Mounted-Display (OST-HMD). The OST-HMD used 10% of its maximal brightness in order to result in comparable brightness between both output media. Its focus was adjusted on the distance to the computer screen. Both devices worked with a resolution of 800x600 pixels at a frame rate of 60 Hz. The experiment ran on a Dell-PC with Pentium 4 processor at 2,6 GHz and 1024 MB DDR-Ram under Windows XP. In order

to keep the visual alignment as accurate as possible, a chin rest was used, situated at 61 cm in front of the screen (see Figure 50). The experiment took place in a room without windows under indirect artificial lightning.

Design The two main independent variables were the display (CRT screen versus OST-HMD) and the kind of presentation (mixed, full). In each of the four conditions, targets were presented once at each of the 4 * 4 = 16 potential target positions. In addition, there were 16 trials without targets. That is, the visual search matrix was presented either completely on the CRT-Display, or completely on the HMD, or the left half on the CRT and the right half on the HMD, or vice-versa. This sums up to 2 (kinds of display) * 2 (kind of presentation) * 16 (item positions) * 2 (target / no target) = 128 trials per block. A block was repeated 10 times resulting in 1280 trials per participant. The dependent variables were reaction time and correct responses.

Procedure After having mounted the OST-HMD on the right eye, observers had to adjust the display mirror and the height of the chin rest so that both media, the OST-device and the screen, covered the same visual field. This was done by presenting 36 O's on the screen as well as on the OST-device which were to be brought into overlay. Then, the visual search task was demonstrated, and 20 practice trials were performed.

Figure 50: Subject performing the visual search task.

A trial started by pressing a key. Participants were instructed to respond with the right index finger pressing the right mouse key when a target was presented among a

set of distracters ("yes"-answer), and using the left index finger and the left mouse key for answering "no". Participants were instructed to answer as fast and as accurate as possible. After 128 trials, participants could take a rest. The whole experiment lasted about one hour per participant.

Participants Five volunteers participated in the experiment aged between 23 and 30 (26 in mean). All reported normal or corrected-to-normal vision, and were familiar with computers and CRT-Displays. None of them had prior experience with OST-HMDs.

6.5.3. Results and Discussion

Reaction Times Data were entered into a 2 (device) * 2 (kind of presentation) analysis of variance with repeated measures. With the computer screen, participants achieved a mean reaction time (RT) of 801.94 ms (standard error se=109.19), with the OST-device 920.89 ms (se = 121.29 ms). In both mixed presentations, mean reaction times were 922.80 ms (se = 137.77, screen-left - OST-HMD-right) and 937.72 (se = 147.29; OST-HMD-left - screen-right). The display produced significantly faster reaction times with the CRT screen relative to the OST-HMD ($F(1,4)=11.071$, $p<.05$). The mixed presentations were with $F(1,4)=5.847$, $p=.07$ marginally slower than the pure ones. The interaction between display and kind of presentation was significant ($F(1,4)=17.701$, $p<.05$). Post hoc comparisons revealed that search times were faster when stimuli were presented on the computer screen relative to all other conditions, which did not differ (Figurere 51).

Error Rates The error rates were in mean 3.56% (se = 1.15) with the computer screen, 10.25% (se = 3.97) with the OST-device, and 5.06% (se = 1.49) and 7.43% (se = 2.14) in both mixed presentation conditions (see Figure 52). Neither the display ($F(1,4)=2.242$, $p=.209$), nor the kind of presentation ($F(1,4)=0.702$, $p=.45$), nor the interaction between display and presentation ($F(1,4)=0.288$, $p=.62$) was significant.

6.5.4. Discussion

First of all, the present results suggest that visual performance is impaired by the OST-device relative to performance on a CRT computer screen. Of course, it remains unclear whether this general difference has to be attributed to some weaknesses in the OST-HMD or simply to the higher familiarity of CRT computer screens.

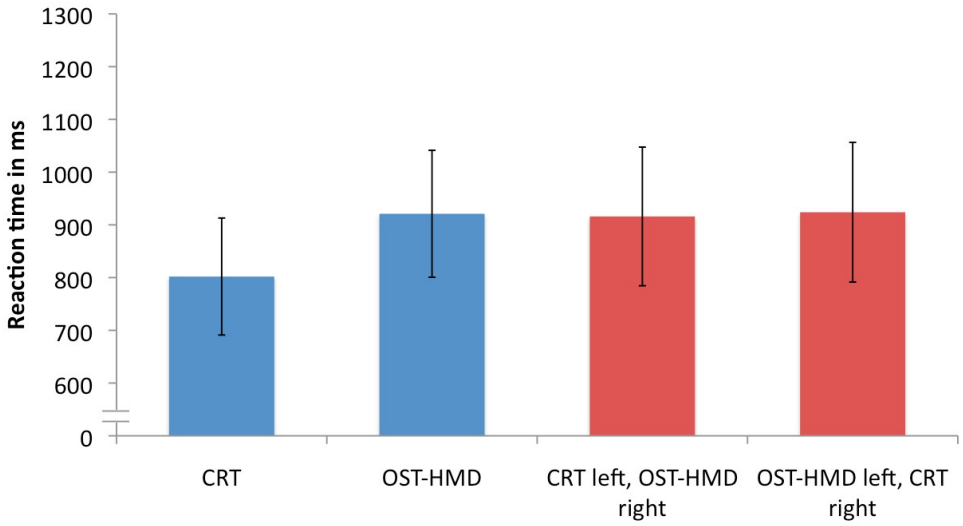

Figure 51: Mean reaction times in the visual search task when stimuli were presented on the CRT computer screen, on the OST-device, or mixed on both (screen-left OST-HMD-right; OST-HMD-left screen-right).

In addition, performance for the mixed conditions (i.e., when stimuli were distributed on both devices) was slower than it would be expected by the mean of both pure conditions. These costs in performance must arise due to switching between devices. In other words, switching between a CRT screen and the OST-device costs time. These results contradict slightly the findings of Gupta [23], where she reports a significant difference in reaction time for tasks that require switching between real and virtual information, using a similar set-up as on this experiment, at a distance of 6 meters, but not at distances of 2 meters and 70 centimeters. This effect is especially interesting since the two main characteristics of OST-devices, the self-illumination and the misleading changes in size with changing viewing distance were excluded in the current experimental set-up. Before speculating about the origins of switching costs, the nature of these costs is to be further examined using another experimental set-up.

6.6. Experiment 2: Dual Task

6.6.1. Research Question

There is one important issue related to switching between media, which is especially important for the industrial application of OST-devices. This question is how well observers can perceive information from a medium while attending to another medium.

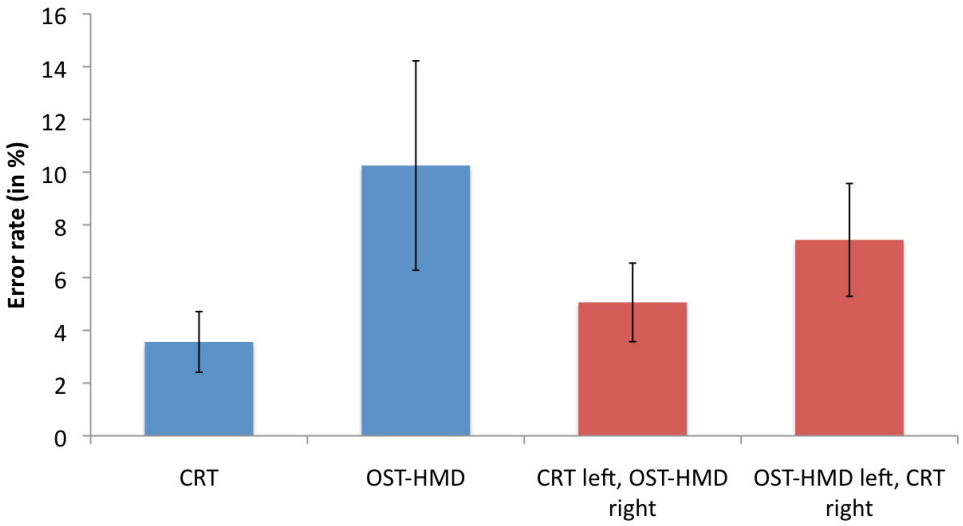

Figure 52: Error rates in the visual search task when stimuli were presented on the CRT computer screen, on the OST-device, or mixed on both (screen-left OST-HMD-right; OST-HMD-left screen-right).

For example, when users read from an OST-HMD, how well can they perceive a signal from the real environment or vice versa? In order to investigate this question, an experimental set-up was developed enlarging the focus of Experiment 1.

The main change is that Experiment 2 is based on a dual task-scenario. That is, the main task remained the visual search. But, in addition, a further task was included which was to discriminate two similar objects. The discrimination of these objects, namely "P" and "R" appearing in the center of the visual field, were presented in a Go/No Go task. That means, responses were to be given only to one stimulus (the P) while ignoring the other. This Go/No Go task appeared unpredictably in time but was to be performed primarily. The purpose of this task was a simulation of events, which might require a reaction (e.g., in case of danger) or not. Both, the visual search task as well as the Go/No Go task, could appear on the computer screen and on the OST-HMD. This allows quantifying additional capacity required for performing both tasks on separate media, that is, the switching costs.

6.6.2. Methods

The method was the same as in the visual search task described in Experiment 1 with the following differences.

The visual search task was either presented completely on the CRT-screen or on the OST-HMD. Mixed trails were not included. Additionally to the visual search task, participants performed simultaneously a Go/No Go task. This task consisted of responding to the presentation of a stimulus "P" and ignoring the presentation of a distracter "R". The letters were presented with the same font as the stimuli of the visual search task. They were displayed in the centre of the screen in random time intervals ranging from 2 to 6 seconds. The letter was presented until key-press, but maximally for two seconds. The response for this Go/NoGo task was given by pressing the space-key of the keyboard. Again, 1280 trials were presented with the visual search task lasting about 70 minutes.

Five volunteers participated in Experiment 2 who were aged between 22 and 30 (26 in mean). All reported normal or corrected-to-normal vision, and were familiar with computers and CRT-Displays. Again, none of them had prior experience with OST-HMDs.

6.6.3. Results and Discusion

6.6.4. Visual Search Task

Reaction Times. Under the condition where no visual switch took place (i.e. both, visual search and Go/No Go tasks presented on the same device) in the visual search task, with the OST-HMD, participants produced mean reaction times of 1097.4 ms (se = 71.71), and with the computer screen 887.4 ms (se = 51.67). In switching trials, participants produced mean reaction times of 1135.1 ms (se = 106.18) with the OST-HMD (and the Go/No Go task on screen) and 987.1 ms (se = 57.78) with the screen (and the Go/No Go task on the OST-device, see Figure 53).

A 2 (device: OST-HMD versus CRT screen) * 2 (switching: with, without) analysis of variance with repeated measures revealed a significant main effect of device ($F(1,4)=22.853$, $p<.01$) as well as a marginal effect of switching ($F(1,4)=5.935$, $p=.072$). The interaction between both factors was not of importance ($F(1,4)=3.264$, $p=.145$).

Error Rates. The mean error rate in the visual search task for trials without switch with the OST-HMD was 5.31% (se = 2.20) and 5.25% (se = 2.92) with the CRT screen. On trials requiring switching between the media, error rates were 6.32% (se = 2.43) with the visual search on the OST-HMD and 3.86% (se = 2.23) with the visual search on the computer screen (see Figure 54). No significant effects were found (device: $F(1,4)=0.383$, $p=.5$; switching: $F(1,4)=.009$, $p=.931$), interaction: $F(1,4)=0.406$, $p=.55$).

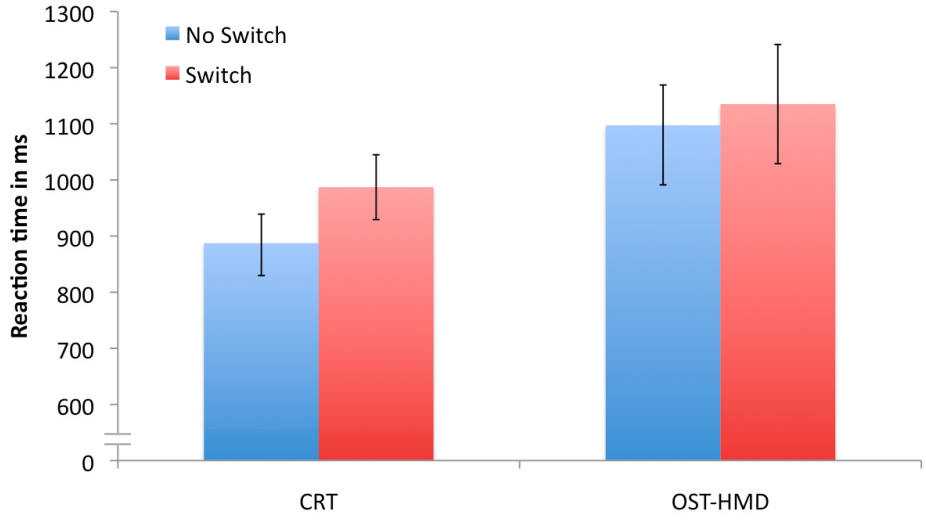

Figure 53: Reaction times in the visual search task when stimuli were presented on the CRT Display or on the OST-HMD, performing a further task on the same (no switch) or the other medium (switch).

6.6.5. Go/No Go Task

Reaction Times. Reaction times in the Go/No Go task were measured from the onset of the presentation of the target "P" until the response. In trials in which both tasks, visual search and Go/No Go, were presented on the same device, participants achieved mean reaction times of 859.4 ms (se = 75.52) with the OST-HMD and 742.0 ms (se = 33.95) with the CRT computer screen. When presenting tasks on different devices, participants achieved a mean reaction time of 830.6 ms (se = 33.77) for the Go/No Go task presented on the OST-HMD and 711.8 ms (se = 11.70) when presenting the Go/No Go task on the CRT computer screen. A 2 (switching) * 2 (devices) analysis of variance with repeated measures revealed no significant effects (device: $F(1,4)=3.313$, $p=.143$; switching: $F<1$; interaction device * switching: $F(1,4)=1.857$, $p=.245$).

Error Rates. When the Go/No Go task was presented on the same device as the visual search (i.e., inn trials without switching) participants produced a mean error rate of 0.68% (se=0.68) with the OST-HMD, and 0.54% (se=0.54) with the CRT computer screen. In trials in which switching was necessary, error rates were 0.64% (se= 0.64) on the OST-HMD and 1.09% (se=0.7) on the CRT screen. These differences were not of significance, neither for device ($F(1,4)=3.313$, $p=.143$), nor for switching ($F(1,4)=1.857$, $p=.245$), nor for the interaction between both ($F(1,4)=.001$, $p=.981$).

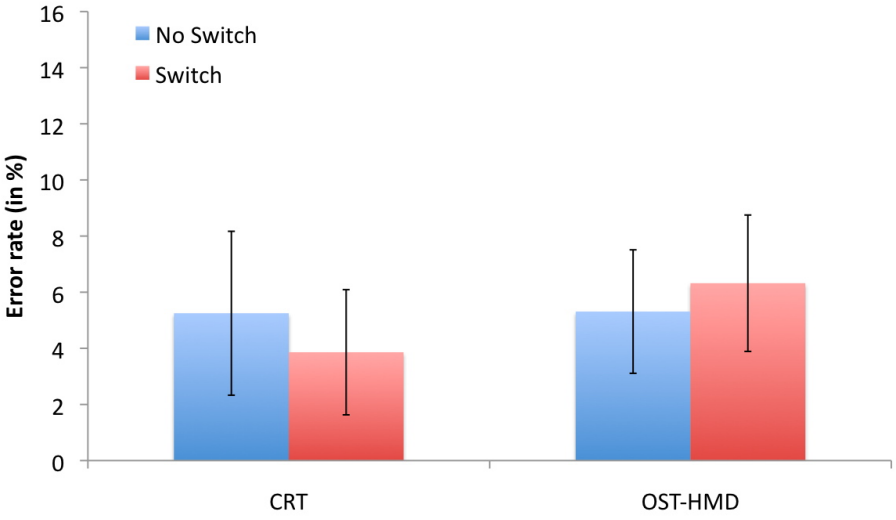

Figure 54: Mean error rate in the visual search task when stimuli were presented on the CRT Display or on the OST-HMD, performing a further task on the same (no switch) or the other medium (switch).

6.6.6. Discussion

The present results basically replicate the observations made in Experiment 1 with a different task set-up. First of all, performance was worse when stimuli were presented on the OST-HMD than on the CRT computer screen. This general difference might be due to the high familiarity of the computer screen. Hence, studies examining long-term usage of the OST-HMD (e.g. [80]) thus quantifying learning rates would be of help in estimating the effects of familiarity.

Like in Experiment 1, switching between devices resulted in decrements in performance. And again, these costs were obvious in reaction times with effect sizes of about ten percent of the performance (see Figure 53). For practical purposes, ten percent seems rather small. But, one should keep in mind that these ten percent arise under optimal, but artificial viewing conditions. Hence, for practical switching tasks, these switching costs might even be much larger.

Moreover, in Experiment 2, switching costs arose only for the secondary visual search task. Two assumptions can account for this: One might suggest that the missing effect in the Go/No Go task is due to the lower load of this task since only one stimulus was presented in the centre of the visual field. Given that the visual quality of the OST-HMD suffers especially in the visual periphery where the image is deformed because of display distortion, this assumption seems plausible. However, one might also suppose that due

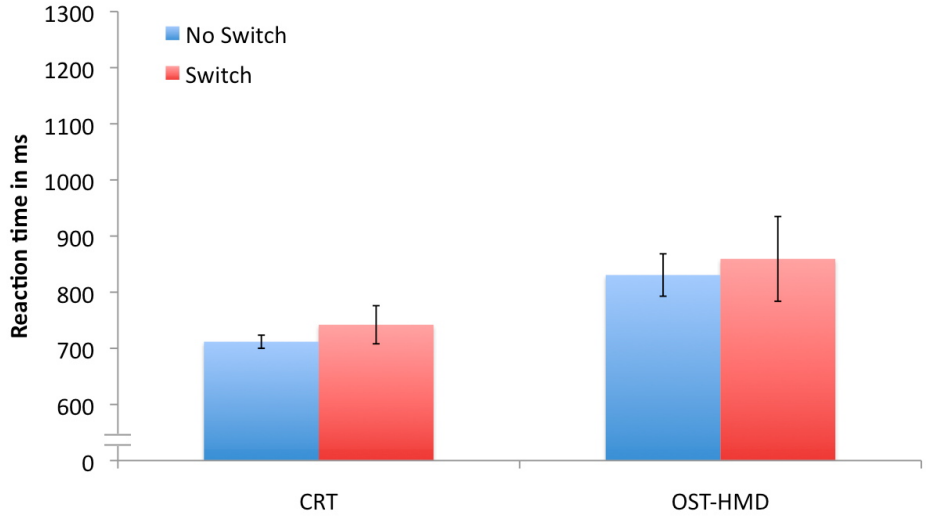

Figure 55: Mean reaction time in the Go/No Go task when performing a further task on the same (no switch) or another medium (switch).

to the primary nature of the Go/No Go task, visual attention might have compensated for the switching costs. This would imply that working with an OST-HMD causes additional attentional load.

One important question concerns the origins of such switching costs. Clearly, differential changes in object size with changing viewing distance as well as the self-illumination and the brightness of the presentation cannot cause the effects at issue. Hence, there must be an additional factor affected when viewing through an OST-HMD and on a CRT computer screen. One assumption concerning this factor responsible for the switching costs was investigated in Experiment 3.

6.7. Experiment 3: Vergence Eye Movements

6.7.1. Research Question

In Experiments 1 and 2, changes between two devices, OST-device and CRT computer screen, produced costs. That is, maintaining fixation and attention on one device was easier than switching between these two devices. This observation is especially astounding since the two most crucial characteristics of OST-HMDs, the self-illuminating nature of the virtual information as well as the disturbing size effects with changing viewing distance, were excluded in the experimental set-ups. Hence, the question is how these switching costs have emerged.

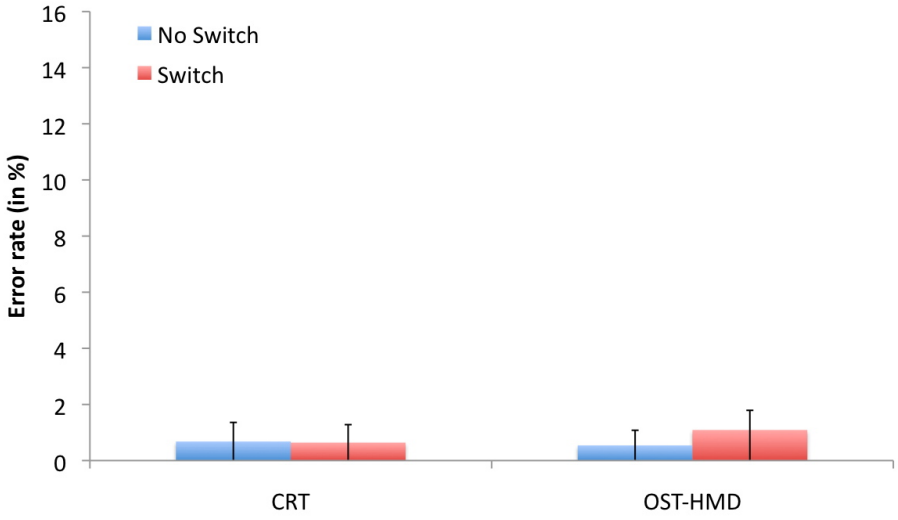

Figure 56: Error rates in the Go/No Go task when performing a further task on the same (no switch) or another medium (switch).

One assumption with respect to switching is that the perceived depth of the information differs: Ideally, the information presented on the OST-device should be perceived on the screen. However, when viewing through an OST-HMD, one has the impression that the stimuli appear not directly on the background, but shortly in front of it. For the current studies, this would imply that stimuli presented on the OST-HMD were localized further towards the observer than stimuli presented on the computer screen.

Of course, this assumption is preliminary in nature. To our knowledge, the hypothesis that stimuli displayed on an OST-HMD are mislocalized in that they seem to be perceived in front of the respective background had not been presented elsewhere. This suggests that this difference, if existing at all, might be rather small in nature. And furthermore, even if there might be a small difference in localization performance, it remains unclear whether such a small difference may be responsible for the switching effects at issue.

In order to investigate whether the visual system indeed processes stimuli presented on the computer screen and on the OST-HMD at different depth layers, the convergence point of the eyes was measured: Measuring the convergence point means establishing the point in which both viewing axes cross. By assumption, this point reflects the focus of the visual system. If there is a perceived difference in depth between stimuli displayed on the computer screen and on the OST-HMD, then there should be differences in convergence between both devices.

Figure 57: Set-up of the OST-HMD with the EyeTracker.

6.7.2. Methods

Stimuli An "X" was presented in the centre of the device for one second. Colour, size and font were the same as used in experiment 1.

Apparatus Stimuli were presented three times on the computer screen and on the OST-HMD. Additionally, a head-mounted eye-tracker (EyeLinkII, SR-Research) was used to measure binocular viewing positions at 250 Hz.

The eye tracker was calibrated using nine points distributed equally over the whole screen. After ten trials, a re-calibration took place. In order to keep the tracking calibration and visual alignment as accurate as possible, a chin rest was used, situated at 61 cm in front of the display.

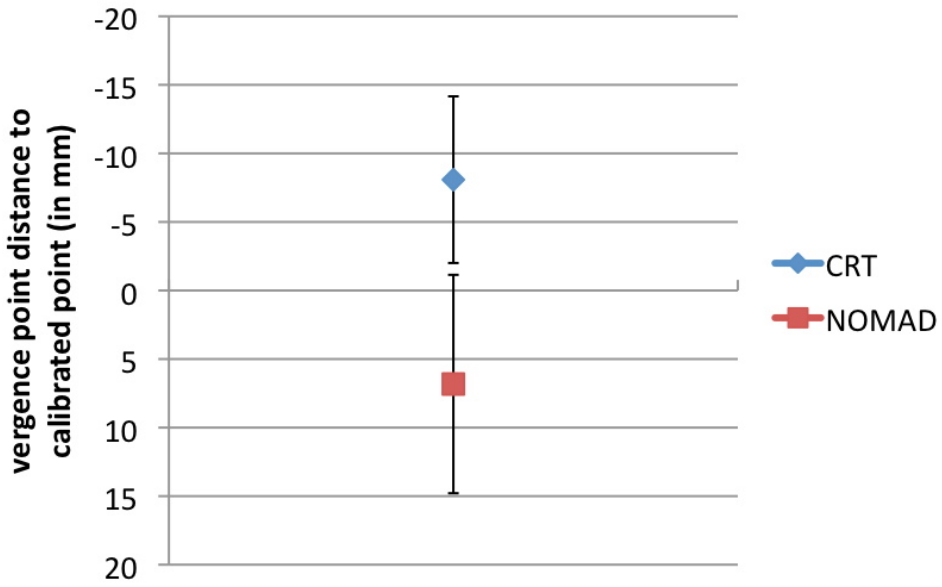

Figure 58: Convergence point when fixating a cross presented on a computer screen (in blue) or on an OST-HMD (red). 0 represents the calibrated baseline.

Procedure After aligning the OST-HMD with the computer screen so that the information displayed on one medium overlaid the same information presented on the other, the eye tracker was calibrated with the computer screen. Then, an "X" was presented for 1 second in random order 10 times on each device. The gaze coordinates of both eyes were registered after 750 ms of exposition. 750 ms were chosen since it takes a while until the eyes maintain in their intended position. The pause between each presentation was one second.

Participants The convergence point was determined in ten participants reporting normal or corrected to normal vision. Participants were aged between 21 and 32 years (26.3 in mean).

6.7.3. Results

In order to determine the convergence point, the horizontal coordinates of both eyes were analyzed, and mean differences between the horizontal coordinates of the left and right eye were computed. Since the viewing distance was constant, these differences (which are given in pixels) can then be traced back to the point in space where the left and right eyes cross. This convergence point is given in mm.

The mean distance between the left and right eye when fixating the cross on the OST-HMD was -13.46 pixels (se = 15.73), meaning that participants converged 6.82 mm in front of the background, that is the computer screen. On the CRT computer screen, the mean distance between both eyes was 15.97 pixels (se = 12.01). Hence, participants converged 8.08 mm behind the CRT-Display (see Figure 6). This difference was of significance ($F(1,9)=6.184$, $p<0.05$).

6.7.4. Discussion

In fact, there were significant differences when fixating a cross displayed on a OST-HMD viewed before a black computer screen and on the computer screen itself: Whereas for the X presented on the screen, after a certain time of fixation, the eyes converged behind the screen, for presentation on the OST-HMD, the eyes converged in front of the screen. These data suggest that the visual system mislocalizes the stimuli presented on the OST-HMD relative to the screen. Still, the origins of this mislocalization are unclear. One might suggest that the self-illumination should be an important factor since the brighter objects are the closer they are perceived. Nevertheless, also the stimuli presented on the screen are self-illuminating. In addition, since the eye tracker was calibrated using the screen, one would expect the convergence point for the screen to be on the screen. Here, the question arises why this point shifted towards the back after a short time of fixating. One might assume visual fatigue to be responsible for this effect: Given that the eyes move towards their rest position with increasing fixation duration, one might assume that the primary mislocalization with the OST-HMD is even stronger.

Nevertheless, one might argue that a difference of about 1.5 cm in the convergence point might not be of practical importance for visual performance. However, current data show that visual acuity declines for the depth axis much steeper than for the horizontal and even for the vertical axis (e.g., [31]). Hence, visual acuity declines rapidly in depth thus requiring an exact determination of the depth.

6.8. Conclusion

6.8.1. General DIfference on Visual Performance

Experiments 1 and 2 showed that visual search takes more time when performed on an OST-HMD than on a CRT computer screen. Of course, on the basis of the current data, it must remain unclear where these performance differences stem from. One might plausibly assume that the familiarity with the computer screen is one powerful factor working here. This unspecific effect alone might be able to cause effects of the size at

issue. Given that this is the case, long-term studies familiarizing observers with the device should lead to a severe reduction in these performance differences.

However, one cannot exclude other factors responsible for general performance differences between the OST-HMD and the computer screen. First of all, the specific OST-HMD used in this study (Nomad ND 2100) might include characteristics which are special only for this certain device. In order to learn more about general and specific characteristics of OST-HMDs, we recommend to replicate the present findings using various OST-HMDs such as the LitEye LE-750 or the RockwellCollins ProView™SO35-MTV. In order to discard binocular rivalry as the reason for the difference in visual search performance, we recommend to replicate this experiment under two further conditions, by occluding the eye that is not used with the monocular OST-HMD and by using a binocular OST-HMDs.

A third possibility is that a characteristic inherent in OST-HMDs produces this output loss. As the observations in Experiment 3 suggest, the information presented on the OST-HMD is mislocalized. If this effect is of general nature in OST-HMDs, then the visual strain reported in various studies might become plausible: For focused vision, not only the convergence point but also accommodation have to be adjusted on the optical information. In normal vision, these two mechanisms are associated. Given that the mislocalization happens as an effect of secondary depth cues in OST-HMDs, vergence and accommodation have to be dissociated in OST-HMDs. As has been reported [80, 19], this can cause eye strain.

6.8.2. Switching Costs

In Experiment 1, visual search performance in trials in which stimuli were presented simultaneously on both devices produced slower search performance than it would be expected by the mean performance of the two pure presentation conditions. That is, switching between the devices produces costs for the visual system. In Experiment 2, dual-task performance was studied when first and second task appeared on the same versus on differential media. Again, switching between the devices resulted in slower search performance than maintaining attention on the same device. In both experiments, switching impaired performance to about 10%.

These results are surprising, since two of the main characteristics of OST-HMDs, which might disturb visual performance, were omitted to affect the observations. That is, both devices were self-illuminating, and viewing distance was kept constant in order to avoid differential changes in the size of real and virtual objects. This is important since in applications, these switching costs can be expected to become much larger than under the current optimal, but artificial conditions.

Nevertheless, switching between devices impaired performance. Hence, there must be at least one other important factor in OST-HMDs which makes them, for the visual system, different from computer screens. The assumption that differences in depth perception contribute to the switching costs are supported by the results of Experiment 3: Here, observers converged about 1.5 cm in front of the converging point on the screen. These results show that the visual system, when viewing through an OST-HMD, adjusts at a depth layer different from the computer screen serving as background. As has been already mentioned, this would imply that the visual system, when trying to integrate information presented in the real world and on an OST-HMD, must permanently adjust accommodation and vergence. Such a reiteration of these mechanisms might explain the visual strain reported when viewing through OST-HMDs.

6.8.3. Prospects

For application scenarios, the results at issue pose the question how to escape from the costs produced by OST-devices. Importantly, these costs should be expected to be much larger than the ten percent switching costs observed under the present optimal, but artificial viewing conditions. However, some of the general disadvantages might decrease with further usage. Moreover, OST-HMDs help in saving time in other areas, if, for example, looking up in a book or sheet can be omitted. In doing so, their current usage seems to be not at all at its limits. Hence, to which extent the application of OST-HMDs is profitable should be answered by a detailed task analysis. The present findings might contribute in pointing out potential visual affordances.

6.8.4. Implications for Gaze Based Interaction

The results presented in this chapter are specially interesting for gaze based interaction. Numerous implications need to be reviewed and technical challenges need to be solve before transferring gaze based applications to OST-HMDs.

The difference observed on visual performance presents the first challenge. Supposing that the user not only wants to interact with virtual but also with the real information (for example to get context information from it), may lead to different selection (dwelling) times for real and virtual information. Considering the presented data, two alternative interaction concepts are plausible. First, one might take into count the use of two different selection methods, one to select real and one for virtual information, like already introduced an discussed in Chapter 3. An other method can be the use of different dwell times for to differentiate the selection intention between real and virtual objects. Therefore, due the longer perception times, the longer dwell time must be applied to the virtual information.

The reported switching costs between real and virtual information may have an important influence on the interaction with the system. Changing the focus from real to virtual information (and vice versa) is associated with a decay on visual perception of, under optimal conditions, ten percent. This fact needs to be considered when designing interaction for OST-HMDs and can be compensated by the use of adaptive dwell times as selection method or by the use of an adaptive offset (to the actual dwell time) to trigger selection when the system notices a switch on the user's focus.

The perception of the visual information in different depths has a strong repercussion on the tracking equipment need for mobile gaze interaction. For gaze interaction on conventional displays, the user only need to get one eye tracked to get x and y coordinates. For mobile devices (OST-HMDs) the tracking of both eye is required to calculate the vergence point of the gaze axis and the determine the point of regard in depth of the user when real and virtual information is overlaid.

6.9. Future Work

This chapter pointed out differences on visual perception between optical-see-though and conventional displays. These results should be use as a basis and motivate further research on this topic.

The first issue that might addressed is to confirm if the presented results can be transferred to other OST devices. Despite the fact, that the presented results could be replicated with two different tasks, we can not rule out, if the switching cost can be attributed to device-specific characteristics. On an examination with a reduced number of participants, we could observe comparable switching costs using an different OST-HMD (Liteye LE-750A). Please note that the results of this study were not published because of the reduced number of participants and can only be treated as an observation. I recommend to replicate this experiment with other OST-devices.

Also from interest is the question if there are long term learning effects when observing through OST displays. All participants had several years experience on perceiving information through LCD or CRT displays and were completely inexperienced with OST-HMDs. A short-term learning effects could be observed during the practice trials, which can be associated partially to the task itself, but also to the fact, that the user did not know what kind of visual information was expecting her/him.

Moreover, and as already pointed out in section 6.8.1 I encourage to replicate this experiment under conditions that can completely rule the influence of binocular rivalry, using a binocular HMD or under monocular conditions.

7. Summary and Conclusions

This thesis reviewed the physiological characteristics of eye movements and reviewed deeply the work related on gaze based selection. The most common selection method is dwell time, which consists on keeping the gaze over an item for a specific time. This time may vary dramatically according to the using context and the user's skills. As an alternative to dwell times, on a prior work to this thesis I proposed several alternative methods for gaze selection [81]. The most prominent was based on saccadic movements which, with the use of pie menus, could be join together leading to gaze gestures. Gaze gestures has become the state-of-the-art selection method and built, together with pie menus, the main part of this thesis.

Chapter 3 discussed issues related to the design of pie menus for gaze interaction. An extended comparison of selection methods, between the established dwell time and the saccade based selection method (selection borders), showed that selection borders are more suitable for pie menus than short dwell times. The menu structure and layout of pie menus were also analyzed. Therefore, the a study on pie menus for touch systems presented by Kurtenbach and Buxton[46], was replicated using gaze as input method. The results showed that the best selection performance was achieved using pie menus with six items and multiple hierarchical layers [83]. The use of marking menus (gaze selection without showing any visual information or feedback) was investigated in this study as well, showing slower selection speeds than trails where the pie menus were shown. Nevertheless, the overall performance confirmed that marking menus can be operated by gaze movements.

The transferability of gaze controlled pie menus is addressed in chapter 4 reviewing in particular the use-case of desktop navigation. The observation's results showed that pie menus can be used in on a desktop navigation context. All users could complete their tasks in an appropriate time and reported widely the system to be easy to use.

Text entry is one of the key research topic on gaze based interaction. It is fundamental to handicapped persons to communicate and also provides an ideal base for gaze based HCI-Research, since it requires several repeated selections from the user in a natural context. Chapter 5 presented two alternatives to single character selection, based on character bigrams, with and without word completion. A longitudinal study was needed to clear which of the postulated methods could fit best to the users needs. The results showed that, despite significant learning effects and remarkable text entry speeds (8.1 wpm in mean using selection borders) single character selection is not the best method for text entry. The methods based on bigrams considering the probability of the next character (calculated statically or dynamically) performed better than presenting vowels as bigram candidates. Combining text entry with bigrams and word prediction increased the text entry performance (with maximal text entry rates of 17 wpm) and decreased

the cognitive load during typing. The combination of two selection methods was well accepted by the users and enhanced the interaction with pie menus. This result remarked the need of robust word prediction algorithms and appropriate selection methods to provide a fluid text entry process.

Now a days, gaze interaction is getting very attractive for mobile devices, specially in combination with head mounted displays for augmented reality content. Chapter 6 focused on the question, if it is possible to transfer the present knowledge on gaze interaction (gained using conventional displays) to HMDs, exploring the visual perception on HMDs under mixed conditions. In three experiments comparing visual performance when stimuli were presented on an OST-HMD device versus on a computer CRT screen, differences in visual processing were observed. First of all, there were benefits when performing visual tasks on a CRT monitor relative to the OST-HMD. Second, and probably more important, switching between devices produced costs in visual search performance. Additionally, depth perception seems to differ between both devices. This leads to the conclusion that established and promising selection metaphors can not be transferred per se, but they can lead as basis for further research.

A. Aknowledgements

I would like to thank all the people who was involved and participated somehow in thesis.

First my mentors, Prof. Dr. Anke Huckauf and Prof. Dr. Bernd Fröhlich. Specially Prof. Dr. Huckauf, not only for the unconditionally support to my gaze control work, furthermore for finding always time for constructive discussions, technical and critical advises, for the outstanding engagement and always having an ear open.

A special mention deserve every single participant in my all evaluations.

I am grateful to Gudrun and Hanno Fiedler for their amazing support and interest on my work. I wish there were more people like you. I would also like to thank all my colleagues in Ulm for all the support, specially Sarah and Klaus for "adopting me" and making me feel like home, Anne for driving me nuts, correcting everything I gave to you and never forgetting my café. Thank you all for the great time.

Finally I would like to thank my family, my parents Axa and Humberto for their unrestricted support, Kathrin, for being an extraordinary wife and friend. Thank you for believing in me and for your toughness during all these years. Thanks to Lisa, Jonas, Julian and Miramaia for being my motivation and the essence of my life.

Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Weitere Personen waren an der inhaltlich-materiellen Erstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlung- bzw. Beratungsdiensten (Promotionsberater oder anderer Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt. Ich versichere, dass ich nach bestem Wissen die reine Wahrheit gesagt und nichts verschwiegen habe.

Weimar, den 12. April 2012.

Mario Humberto Urbina Cazenave

References

[1] Agustin, J. S., and Hansen, J. P. Off-the-shelf mobile gaze interaction. In *Proceedings of the 4th Conference on Communication by Gaze Interaction (COGAIN 2008)* (2008), pp. 6–10.

[2] Agustin, J. S., Skovsgaard, H. H. T., Hansen, J. P., and Hansen, D. W. Low-cost gaze interaction: ready to deliver the promises. In *CHI Extended Abstracts* (2009), pp. 4453–4458.

[3] Ashmore, M., Duchowski, A. T., and Shoemaker, G. Efficient eye pointing with a fisheye lens. In *Proceedings of Graphics Interface 2005* (School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005), GI '05, Canadian Human-Computer Communications Society, pp. 203–210.

[4] Ashtiani, B., and MacKenzie, I. S. Blinkwrite2: an improved text entry method using eye blinks. In *ETRA '10: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (New York, NY, USA, 2010), ACM, pp. 339–345.

[5] Azuma, R. A survey of augmented reality. *Presence 6* (1995), 355–385.

[6] Bates, R., and Istance, H. Zooming interfaces!: enhancing the performance of eye controlled pointing devices. In *Proceedings of the fifth international ACM conference on Assistive technologies* (New York, NY, USA, 2002), Assets 2002, ACM, pp. 119–126.

[7] Bee, N., and André, E. Writing with your eye: A dwell time free writing system adapted to the nature of human eye gaze. In *Proceedings of Workshop on Perception and Interactive Technologies for Speech-Based Systems* (2008).

[8] Bichlmeier, C., Ockert, B., Heining, S. M., Ahmadi, A., and Navab, N. Stepping into the operating theater: Arav. *Mixed and Augmented Reality, IEEE / ACM International Symposium on 0* (2008), 165–166.

[9] Biocca, F. A., and Rolland, J. P. Virtual eyes can rearrange your body: Adaptation to visual displacement in see-through, head-mounted displays. *Presence: Teleoper. Virtual Environ. 7*, 3 (1998), 262–277.

[10] Callahan, J., Hopkins, D., Weiser, M., and Shneiderman, B. An empirical comparison of pie vs. linear menus. In *CHI '88: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1988), ACM, pp. 95–100.

[11] Dorn, F. J., Atwater, M., Jereb, R., and Russell, R. Determining the reliability of the nlp eye-movement procedure. *American Mental Health Counselors Association 5(3)* (1983), 105–110.

[12] Doughty, M. Further assessment of gender- and blink pattern-related differences in the spontaneous eyeblink activity in primary gaze in young adult humans. In *Optom Vis Sci* (2002), vol. 79, pp. 439–447.

[13] Drewes, H., and Schmidt, A. Interacting with the computer using gaze gestures. In *Proceedings of Human-Computer Interaction - INTERACT 2007* (2007), Springer, pp. 475–488.

[14] Duchowski, A. T. *Eye Tracking Methodology - Theory and Practice.* Springer-Verlag, 2003.

[15] Duchowski, A. T. *Eye Tracking Methodology - Theory and Practice*, 2nd ed. Springer, 2007.

[16] Ellis, S. R., and Menges, B. M. Localization of virtual objects in the near visual field. *Human Factors: The Journal of the Human Factors and Ergonomics Society 40*, 3 (1998), 415–431.

[17] Fischer, B., and Everling, S. The antisaccade: A review of basic research and clinical studies. *Neuropsychologia 36* (1998), 885–899.

[18] Fono, D., and Vertegaal, R. Eyewindows: evaluation of eye-controlled zooming windows for focus selection. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2005), CHI '05, ACM, pp. 151–160.

[19] Fritzsche, L. Eignung von augmented reality für den vollschichteinsatz in der automobilproduktion. Master's thesis, TU Dresden, Germany, 2006.

[20] Gabbard, J. L., Swan II, J. E., Hix, D., Lucas, J., and Gupta, D. An empirical user-based study of text drawing styles and outdoor background textures for augmented reality. In *VR '05: Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 11–18, 317.

[21] Graupner, S.-T., Heubner, M., Pannasch, S., and Velichkovsky, B. M. Evaluating requirements for gaze-based interaction in a see-through head mounted display. In *ETRA '08: Proceedings of the 2008 symposium on Eye tracking research & applications* (New York, NY, USA, 2008), ACM, pp. 91–94.

[22] Grubert, J., Hamacher, D., Mecke, R., Bockelmann, I., Schega, L., Huckauf, A., Urbina, M., Schenk, M., Doil, F., and Tümler, J. Extended investigations of user-related issues in mobile industrial ar. In *ISMAR '10: Proceedings of the 2010 9th IEEE International Symposium on Mixed and Augmented Reality* (Los Alamitos, CA, USA, 2010), IEEE Computer Society, pp. 229–230.

[23] Gupta, D. An empirical study of the effects of context-switch, object distance, and focus depth on human performance in augmented reality. Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA, 2004.

[24] Hallett, P. Primary and secondary saccades to goals defined by instructions. *Vision Res 18* (1978), 1279–1296.

[25] Hansen, D. W., Skovsgaard, H. H. T., Hansen, J. P., and Møllenbach, E. Noise tolerant selection by gaze-controlled pan and zoom in 3d. In *ETRA '08: Proceedings of the 2008 symposium on Eye tracking research & applications* (New York, NY, USA, 2008), ACM, pp. 205–212.

[26] Hansen, J. P., Torning, K., Johansen, A. S., Itoh, K., and Aoki, H. Gaze typing compared with input by head and hand. In *ETRA '04: Proceedings of the 2004 symposium on Eye tracking research & applications* (New York, NY, USA, 2004), ACM Press, pp. 131–138.

[27] Heikkilä, H., and Räihä, K.-J. Speed and accuracy of gaze gestures. *Journal of Eye Movement Research 3*, 2 (November 2009), 1–14.

[28] Hopkins, D. The design and implementation of pie menus. *Dr. Dobb's Journal 16*, 12 (1991), 16–26.

[29] Hornof, A. J., and Cavender, A. Eyedraw: enabling children with severe motor impairments to draw with their eyes. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2005), ACM, pp. 161–170.

[30] Huckauf, A., Goettel, T., Heinbockel, M., and Urbina, M. What you don't look at is what you get: anti-saccades can reduce the midas touch-problem. In *APGV '05: Proceedings of the 2nd symposium on Applied perception in graphics and visualization* (New York, NY, USA, 2005), ACM Press, pp. 170–170.

[31] Huckauf, A., Müsseler, J., and Fährmann, F. Sehschärfeverteilung in der dritten dimension. *51. Tagung experimentell arbeitender Psychologen* (2009), 82.

[32] Huckauf, A., and Urbina, M. H. Gazing with peyes: towards a universal input for various applications. In *ETRA '08: Proceedings of the 2008 symposium on Eye tracking research & applications* (New York, NY, USA, 2008), ACM, pp. 51–54.

[33] Huckauf, A., and Urbina, M. H. On object selection in gaze controlled environments. In *Journal of Eye Movement Research* (2008), vol. 2 of *4*, pp. 1–7.

[34] Huckauf, A., and Urbina, M. H. Object selection in gaze controlled systems: What you don't look at is what you get. *ACM Trans. Appl. Percept. 8* (February 2011), 13:1–13:14.

[35] Isokoski, P. Text input methods for eye trackers using off-screen targets. In *Proceedings of the 2000 symposium on Eye tracking research & applications* (New York, NY, USA, 2000), ETRA '00, ACM, pp. 15–21.

[36] Istance, H., Bates, R., Hyrskykari, A., and Vickers, S. Snap clutch, a moded approach to solving the midas touch problem. In *ETRA '08: Proceedings of the 2008 symposium on Eye tracking research & applications* (New York, NY, USA, 2008), ACM, pp. 221–228.

[37] Istance, H., Hyrskykari, A., Immonen, L., Mansikkamaa, S., and Vickers, S. Designing gaze gestures for gaming: an investigation of performance. In *ETRA '10: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (New York, NY, USA, 2010), ACM, pp. 323–330.

[38] Jacob, R. J. K. The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Trans. Inf. Syst. 9*, 2 (1991), 152–169.

[39] Jacob, R. J. K. Interaction techniques: Toward non-command interfaces. *In H.R. Hartson and D. Hix (Eds.) Advances in Human-Computer Interaction 4* (1993), 151–190. Available at http://www.cs.tufts.edu/ jacob/papers/hartson.pdf.

[40] Jarvenpaa, T., and Aaltonen, V. Compact near-to-eye display with integrated gaze tracker.

[41] Johansen, A. S., Hansen, J. P., Hansen, D. W., Itoh, K., and Mashino, S. Language technology in a predictive, restricted on-screen keyboard with dynamic layout for severely disabled people. In *Proceedings of the 2003 EACL Workshop on Language Modeling for Text Entry Methods* (Stroudsburg, PA, USA, 2003), TextEntry '03, Association for Computational Linguistics, pp. 59–66.

[42] Kammerer, Y., Scheiter, K., and Beinhauer, W. Looking my way through the menu: the impact of menu design and multimodal input on gaze-based menu selection. In *In Proceedings of the 2008 Symposium on Eye Tracking Research & Applications* (New York, NY, USA, 2008), ACM, pp. 213–220.

[43] Kaur, M., Tremaine, M., Huang, N., Wilder, J., Gacovski, Z., Flippo, F., and Mantravadi, C. S. Where is "it"? event synchronization in gaze-speech input systems. In *ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces* (New York, NY, USA, 2003), ACM Press, pp. 151–158.

[44] Kristjánsson, A., Vandenbroucke, M. W., and Driver, J. When pros become cons for anti- versus prosaccades: factors with opposite or common effects on different saccade types. *Experimental Brain Research 155* (2004), 231–244.

[45] Kumar, M., Paepcke, A., and Winograd, T. Eyepoint: practical pointing and selection using gaze and keyboard. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2007), ACM, pp. 421–430.

[46] Kurtenbach, G., and Buxton, W. The limits of expert performance using hierarchic marking menus. In *CHI '93: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1993), ACM Press, pp. 482–487.

[47] Kurtenbach, G., and Buxton, W. User learning and performance with marking menus. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1994), ACM Press, pp. 258–264.

[48] Laramee, R. S., and Ware, C. Rivalry and interference with a head-mounted display. *ACM Trans. Comput.-Hum. Interact. 9*, 3 (2002), 238–251.

[49] MacKenzie, I. S., and Zhang, S. X. The design and evaluation of a high-performance soft keyboard. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1999), ACM Press, pp. 25–31.

[50] MacKenzie, I. S., and Zhang, X. Eye typing using word and letter prediction and a fixation algorithm. In *ETRA '08: Proceedings of the 2008 symposium on Eye tracking research & applications* (New York, NY, USA, 2008), ACM, pp. 55–58.

[51] Majaranta, P., Ahola, U.-K., and Špakov, O. Fast gaze typing with an adjustable dwell time. In *CHI '09: Proceedings of the SIGCHI conference on Human*

factors in computing systems (New York, NY, USA, 2009), ACM Press, pp. 357–360.

[52] Majaranta, P., Aula, A., and Räihä, K.-J. Effects of feedback on eye typing with a short dwell time. In *ETRA '04: Proceedings of the 2004 symposium on Eye tracking research & applications* (New York, NY, USA, 2004), ACM Press, pp. 139–146.

[53] Majaranta, P., MacKenzie, S., Aula, A., and Räihä, K.-J. Effects of feedback and dwell time on eye typing speed and accuracy. *Univers. Access Inf. Soc. 5*, 2 (2006), 199–208.

[54] Majaranta, P., Majaranta, N., Daunys, G., and Špakov, O. Text editing by gaze: Static vs. dynamic menus. In *In Proceedings of the 5th Conference on Communication by Gaze Interaction; COGAIN 2009* (2009), pp. 19–23.

[55] Majaranta, P., and Räihä, K.-J. Twenty years of eye typing: systems and design issues. In *ETRA '02: Proceedings of the 2002 symposium on Eye tracking research & applications* (New York, NY, USA, 2002), ACM Press, pp. 15–22.

[56] Mateo, J. C., San Agustin, J., and Hansen, J. P. Gaze beats mouse: hands-free selection by combining gaze and emg. In *CHI '08 extended abstracts on Human factors in computing systems* (New York, NY, USA, 2008), CHI EA '08, ACM, pp. 3039–3044.

[57] Miniotas, D., Špakov, O., and Evreinov, G. E. Symbol creator: An alternative eye-based text entry technique with low demand for screen space. In *INTERACT* (2003).

[58] Miniotas, D., Špakov, O., and MacKenzie, I. S. Eye gaze interaction with expanding targets. In *CHI '04 extended abstracts on Human factors in computing systems* (New York, NY, USA, 2004), CHI EA '04, ACM, pp. 1255–1258.

[59] Miniotas, D., Špakov, O., Tugoy, I., and MacKenzie, I. S. Speech-augmented eye gaze interaction with small closely spaced targets. In *ETRA '06: Proceedings of the 2006 symposium on Eye tracking research & applications* (New York, NY, USA, 2006), ACM, pp. 67–72.

[60] Mizell, D. *Boeing's Wire Bundle Assembly Project.* Fundamentals of Wearable Computers and Augmented Reality. Lawrence Erlbaum & Associates, New Jersey, 2001, ch. 14, pp. 447–467.

[61] Møllenbach, E., Lillholm, M., Gail, A., and Hansen, J. P. Single gaze gestures. In *ETRA '10: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (New York, NY, USA, 2010), ACM, pp. 177–180.

[62] Morimoto, C. H., and Amir, A. Context switching for fast key selection in text entry applications. In *ETRA '10: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (New York, NY, USA, 2010), ACM, pp. 271–274.

[63] Mueller, S., Swainson, R., and Jackson, G. Erp indices of persisting and current inhibitory control: A study of saccadic task switching. *NeuroImage 45*, 1 (2009), 191–197.

[64] Nilsson, S., Gustafsson, T., and Carleberg, P. Hands free interaction with virtual information in a real environment. In *Proceedings of the 3rd Conference on Communication by Gaze Interaction (COGAIN 2007)* (2007), pp. 53–57. Available online at http://www.cogain.org/cogain2007/COGAIN2007Proceedings.pdf.

[65] Ockerman, J. J., and Pritchett, A. R. Preliminary investigation of wearable computers for task guidance in aircraft inspection. In *Proceedings of the 2nd IEEE International Symposium on Wearable Computers* (Washington, DC, USA, 1998), ISWC '98, IEEE Computer Society, pp. 33–37.

[66] Pannasch, S., Helmert, J. R., Malischke, S., Storch, A., and Velichkovsky, B. M. Eye typing in application: A comparison of two systems with als patients. In *Journal of Eye Movement Research* (2008), vol. 2 of *6*, pp. 1–8.

[67] Perlin, K. Quikwriting: continuous stylus-based text entry. In *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology* (New York, NY, USA, 1998), ACM, pp. 215–216.

[68] Pommerening, K. Kryptologie - zeichenhäufigkeiten in deutsch (in german). http://www.staff.uni-mainz.de/pommeren/Kryptologie/Klassisch/1_Monoalph/deutsch.html, 2007. Last visited on 31/07/2007.

[69] Porta, M., and Turina, M. Eye-s: a full-screen input modality for pure eye-based communication. In *ETRA '08: Proceedings of the 2008 symposium on Eye tracking research & applications* (New York, NY, USA, 2008), ACM, pp. 27–34.

[70] Schwerdtfeger, B., Reif, R., Gunthner, W. A., Klinker, G., Hamacher, D., Schega, L., Bockelmann, I., Doil, F., and Tumler, J. Pick-by-vision: A first stress test. In *ISMAR '09: Proceedings of the 2009 8th IEEE International*

Symposium on Mixed and Augmented Reality (Washington, DC, USA, 2009), IEEE Computer Society, pp. 115–124.

[71] Shi, F., and Gale, A. Environmental control by remote eye tracking. In *Proceedings of the 3rd Conference on Communication by Gaze Interaction (COGAIN 2007)* (2007), pp. 49–52. Available online at http://www.cogain.org/cogain2007/COGAIN2007Proceedings.pdf.

[72] Soukoreff, R. W., and MacKenzie, I. S. Measuring errors in text entry tasks: an application of the levenshtein string distance statistic. In *CHI '01: CHI '01 extended abstracts on Human factors in computing systems* (New York, NY, USA, 2001), ACM Press, pp. 319–320.

[73] Su, M., Yeh, C., Lin, S., Wang, P., and Hou, S. An implementation of an eye-blink-based communication aid for people with severe disabilities. In *Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on* (july 2008), pp. 351 –356.

[74] Surakka, V., Illi, M., and Isokoski, P. Gazing and frowning as a new human–computer interaction technique. *ACM Trans. Appl. Percept. 1*, 1 (2004), 40–56.

[75] Swan, J. E. I., Livingston, M. A., Smallman, H. S., Brown, D., Baillot, Y., Gabbard, J. L., and Hix, D. A perceptual matching technique for depth judgments in optical, see-through augmented reality. In *VR '06: Proceedings of the IEEE conference on Virtual Reality* (Washington, DC, USA, 2006), IEEE Computer Society, pp. 19–26.

[76] Tall, M. Neovisus: Gaze driven interface components. In *Proceedings of the 4rd Conference on Communication by Gaze Interaction (COGAIN 2008)* (2008), pp. 47–51. Available online at http://www.cogain.org/cogain2008/COGAIN2008Proceedings.pdf.

[77] Tien, G., and Atkins, M. S. Improving hands-free menu selection using eyegaze glances and fixations. In *ETRA '08: Proceedings of the 2008 symposium on Eye tracking research & applications* (New York, NY, USA, 2008), ACM, pp. 47–50.

[78] Trnka, K., McCaw, J., Yarrington, D., McCoy, K. F., and Pennington, C. User interaction with word prediction: The effects of prediction quality. *ACM Trans. Access. Comput. 1* (February 2009), 17:1–17:34.

[79] Tuisku, O., Majaranta, P., Isokoski, P., and Räihä, K.-J. Now dasher! dash away!: longitudinal study of fast text entry by eye gaze. In *ETRA '08: Proceedings*

of the 2008 symposium on Eye tracking research & applications (New York, NY, USA, 2008), ACM, pp. 19–26.

[80] Tümler, J., Doil, F., Mecke, R., Paul, G., Schenk, M., Pfister, E. A., Huckauf, A., Bockelmann, I., and Roggentin, A. Mobile augmented reality in industrial applications: Approaches for solution of user-related issues. In *ISMAR '08: Proceedings of the 2008 7th IEEE International Symposium on Mixed and Augmented Reality* (Los Alamitos, CA, USA, 2008), IEEE Computer Society, pp. 87–90.

[81] Urbina, M. H., and Huckauf, A. Dwell-time free eye typing approaches. In *Proceedings of the 3rd Conference on Communication by Gaze Interaction (COGAIN 2007)* (2007), pp. 65–70.

[82] Urbina, M. H., and Huckauf, A. Selecting with gaze controlled pie menus. In *Proceedings of the 5th International Conference on Communication by Gaze Interaction (COGAIN 2009)* (2009), pp. 25–29.

[83] Urbina, M. H., Lorenz, M., and Huckauf, A. Pies with eyes: the limits of hierarchical pie menus in gaze control. In *ETRA '10: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (New York, NY, USA, 2010), ACM, pp. 93–96.

[84] Vickers, S., Istance, H., Hyrskykari, A., and Ali, N. User performance of gaze-based interaction with on-line virtual communities. In *In Proceedings of the 4th Conference on Communication by Gaze Interaction; COGAIN 2008* (Prague, CZ, September 2008).

[85] Špakov, O., and Miniotas, D. Gaze-based selection of standard-size menu items. In *ICMI '05: Proceedings of the 7th international conference on Multimodal interfaces* (New York, NY, USA, 2005), ACM Press, pp. 124–128.

[86] Ward, D. J., and MacKay, D. J. C. Fast hands-free writing by gaze direction. *Nature 418*, 6900 (2002), 838.

[87] Ware, C., and Mikaelian, H. H. An evaluation of an eye tracker as a device for computer input2. In *CHI '87: Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface* (New York, NY, USA, 1987), ACM Press, pp. 183–188.

[88] Wobbrock, J. O., Rubinstein, J., Sawyer, M. W., and Duchowski, A. T. Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In

ETRA '08: Proceedings of the 2008 symposium on Eye tracking research & applications (New York, NY, USA, 2008), ACM, pp. 11–18.

[89] Yamato, M., Inoue, K., Monden, A., Torii, K., and ichi Matsumoto, K. Button selection for general guis using eye and hand together. In *AVI '00: Proceedings of the working conference on Advanced visual interfaces* (New York, NY, USA, 2000), ACM Press, pp. 270–273.

[90] Yamato, M., Monden, A., ichi Matsumoto, K., Inoue, K., and Torii, K. Quick button selection with eye gazing for general gui environment. In *Proceedings of International Conference on Software: Theory and Practice, ICS 2000* (2000), pp. 712–719.

[91] Zhai, S. What's in the eyes for attentive input. *ACM Communications 46 (3)* (2003), 34–39.

[92] Zhai, S. On the ease and efficiency of human-computer interfaces. In *ETRA '08: Proceedings of the 2008 symposium on Eye tracking research & applications* (New York, NY, USA, 2008), ACM, pp. 9–10.

[93] Zhai, S., Morimoto, C., and Ihde, S. Manual and gaze input cascaded (magic) pointing. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1999), ACM Press, pp. 246–253.

[94] Zhang, X., Ren, X., and Zha, H. Improving eye cursor's stability for eye pointing tasks. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2008), CHI '08, ACM, pp. 525–534.