

Aulikki Hyrskykari

**Eyes in Attentive Interfaces:
Experiences from Creating iDict,
a Gaze-Aware Reading Aid**

ACADEMIC DISSERTATION

To be presented with the permission of the Faculty of Information Sciences of the
University of Tampere, for public discussion in Pinni auditorium B1096
on May 19th, 2006, at noon.

Department of Computer Sciences
University of Tampere

Dissertations in Interactive Technology, Number 4
Tampere 2006

ACADEMIC DISSERTATION IN INTERACTIVE TECHNOLOGY

Supervisor: Professor Kari-Jouko Rähkä,
Department of Computer Sciences,
University of Tampere,
Finland

Opponent: Principal Lecturer Howell Istance,
School of Computing,
De Montfort University,
United Kingdom

Reviewers: Docent Jukka Hyönä,
Department of Psychology,
University of Turku,
Finland

Professor Markku Tukiainen,
Department of Computer Science,
University of Joensuu,
Finland

Electronic dissertation
Acta Electronica Universitatis Tamperensis 531
ISBN 951-44-6643-8
ISSN 1456-954X
<http://acta.uta.fi>

Dissertations in Interactive Technology, Number 4

Department of Computer Sciences
FIN-33014 University of Tampere
FINLAND

ISBN 951-44-6630-6
ISSN 1795-9489

Tampereen yliopistopaino Oy
Tampere, 2006

Abstract

The mouse and keyboard currently serve as the predominant means of passing information from user to computer. Direct manipulation of objects via the mouse was a breakthrough in the design of more natural and intuitive user interfaces for computers. However, in real life we have a rich set of communication methods at our disposal; when interacting with others, we, for example, interpret their gestures, expressions, and eye movements. This information can be used also when moving human-computer interaction toward the more natural and effective. In particular, the focus of the user's attention could often be a valuable source of information.

The focus of this work is on examining the benefits and limitations in using the information acquired from a user's eye movements in the human-computer interface. For this purpose, we developed an example application, iDict. The application assists the reader of an electronic document written in a foreign language by tracking the reader's eye movements and providing assistance automatically when the reader seems to be in need of help.

The dissertation is divided into three parts. The first part presents the physiological and psychological basics behind the measurement of eye movements, and we also provide a survey of both the applications that make use of eye tracking and the relevant research into eye movements during reading. The second section introduces the iDict application, from both the user's and the implementer's point of view. Finally, the work presents the experiments that were performed either to inform design decisions or to test the performance of the application.

This work is proof that gaze-aware applications can be more pleasing and effective than traditional application interfaces. The human visual system imposes limits on the accuracy of eye tracking, which is why we, for example, are unable to narrow down with certainty the reader's focus of gaze to a target word. This work demonstrates, however, that errors in interpreting the focus of visual attention can be algorithmically compensated. Additionally, we conclude that the total time spent on a word is a reasonably good indicator in judging comprehension difficulties. User tests with iDict were encouraging. More than half of the users preferred using eye movements to the option of using the application traditionally with the mouse. The result was obtained even when the test users were familiar with using a mouse but not with the concept of the eye as an input device.

Preface and Acknowledgments

In 1996, we acquired our first eye tracking device. At the time, we were a group of people with a deep interest in studying human-computer interaction and usability issues, not so common for computer scientists at the time. The eye tracking device was valuable equipment for the line of research we undertook: we could actually record where the user's visual attention is focused in the use of an application.

We soon became interested in the idea of using the eye tracker also as an input device: to pass to an application the information on the user's visual focus at each point in time. The idea of pioneering a new research area without so many research results to plough through was inspiring to me – at the time. When digging into the subject, I found out that the idea that we would be entering a fresh field of research was more than distorted. There had been a massive amount of eye movement research done, stretching back decades. The basic findings, which still hold in eye movement research, had already been made in the 18th century. Also, the idea of using eye movement in human-computer interaction had been raised many times, starting in the 1980s. And it was also often dropped, the method being considered an infeasible technique for human-computer interaction. In his keynote address at the ETRA 2000 conference, Ted Selker likened this area of study to a phoenix, repeatedly rising from its ashes. In spring 1998, we held a seminar in which one of the pioneers in the field, Robert Jacob, gave a lecture with the title “User Interface Software and Non-WIMP Interaction Techniques.” Discussions with him attenuated my enthusiasm; at the time he was giving up on the line of research. However, accompanying the developments in eye tracking equipment, there seems to be a firm faith now, and more permanently, restored in the subject.

Doing research can be frustrating. Often, one may find oneself delving into issues and finding it hard to justify the investment of time as relevant or of practical use. I have been fortunate enough to have had good motivation for this work. There is a group of people for whom controlling the computer with eye movements is vital. The justification for this work is that developing good gaze-aware applications within reasonable cost limits requires broader user groups for gaze-aware applications. Additionally, I firmly believe that adding an eye tracking device to a standard computer setup would really make the use of computers more pleasing and effective for standard users, too.

Contents

1	Introduction.....	1
1.1	Eye movements as input for a computer	2
1.1.1	Why eyes?.....	3
1.1.2	Problems of eye input.....	5
1.1.3	Challenges for eye-based interaction	6
1.2	Research methods and research focus	10
1.2.1	The key ideas behind iDict	11
1.2.2	Focus of research and contributions.....	11
1.3	Outline of the dissertation.....	12

PART I: Background

2	Gaze Tracking	17
2.1	Biological basis for gaze tracking	17
2.1.1	Human vision – physiological background for gaze tracking	18
2.1.2	Movements of the eyes	22
2.2	Eye tracking techniques.....	23
3	Attention in the Interface.....	27
3.1	Attention in user interfaces	27
3.1.1	Orienting attention.....	28
3.1.2	Control of attention.....	28
3.1.3	Implications for interface and interaction design	29
3.2	Gaze-based attentive systems	32
3.2.1	Interacting with an appliance.....	34
3.2.2	Interacting with a computer.....	38
3.2.3	Interacting with other humans	42
4	Attention and Reading.....	47
4.1	Eye movements and reading	47
4.2	Reading as an attentional process.....	49
4.2.1	Perceptual span field	49
4.2.2	Attentional theory of reading.....	50
4.2.3	Measurement of reading behavior	51
4.3	Summary of Part I.....	52

PART II: The iDict Application

5	iDict Functionality.....	57
5.1	On iDict’s design rationale.....	58
5.2	User interface – iDict from the user’s perspective	58
5.2.1	Starting iDict	59
5.2.2	Automatic dictionary lookups	60
5.2.3	Feedback	61
5.2.4	Optional mouse operation.....	62
5.3	Personalizing the application	63

5.4	Specifying the target language and dictionaries used.....	65
6	iDict Implementation.....	67
6.1	Eye tracking devices used	67
6.1.1	EyeLink	68
6.1.2	iView X	69
6.1.3	Tobii 1750.....	69
6.1.4	Preprocessing of sample data.....	70
6.2	Overview of the iDict architecture	71
6.3	Text document preprocessing and maintaining of the session history.....	72
6.4	Linguistic and lexical processing of a text document.....	74
6.4.1	Linguistic analysis	75
6.4.2	Dictionary lookups.....	77
6.4.3	Example of linguistic processing of a sentence.....	79
6.5	Test bed features.....	80
PART III: Using Gaze Paths to Interpret the Real-Time Progress of Reading		
7	Inaccuracy in Gaze Tracking.....	85
7.1	Sources of inaccuracy.....	85
7.2	Experiences of reading paths in practice.....	86
7.2.1	Vertical inaccuracy	88
7.2.2	Horizontal inaccuracy	90
8	Keeping Track of the Point of Reading	93
8.1	Mapping of fixations to text objects	93
8.2	Dynamic correction of inaccuracy	96
8.3	Drift compensation algorithms.....	98
8.3.1	Sticky lines – a vertically expanding line mask	98
8.3.2	Magnetic lines – relocation of line masks.....	101
8.3.3	Manual correction	102
8.4	Return sweeps in reading.....	103
8.5	Analysis of new line event gaze patterns	105
8.5.1	Identification of new line events.....	105
8.5.2	Number of transition saccades in new line events	107
8.5.3	Transition saccade length	108
8.5.4	First and last NLE fixation locations.....	110
8.5.5	Vertical height of the transition during a new line event	111
8.5.6	Reinforced new line events	111
8.6	New line detection algorithm	112
8.7	Coping with atypical reading patterns	113
8.7.1	Examples of following atypical reading paths.....	114
8.8	Performance evaluation for drift compensation algorithms	115
8.8.1	Test setup.....	115
8.8.2	Analysis of the reading paths	116
8.8.3	Results	116
9	Recognizing Reading Comprehension Difficulties.....	119
9.1	Reading comprehension and eye movement measures	119
9.1.1	Definitions for the measures	120
9.1.2	Measuring reading comprehension in non-ideal conditions.....	121
9.2	Experiment on using the measures in non-ideal conditions.....	122
9.2.1	Experiment setup.....	122

9.2.2	Overview of the data.....	124
9.2.3	Scores for different measures in the experiment.....	126
9.2.4	Discussion and conclusions.....	132
9.3	Total time as a basis for detecting comprehension difficulties.....	133
9.3.1	Total time threshold.....	133
9.3.2	Personalizing total time threshold.....	135
9.3.3	Word frequency and word length.....	137
9.4	Concluding observations on the total time threshold function.....	141
10	Interaction Design of a Gaze-Aware Application.....	143
10.1	Natural versus intentional eye movements.....	143
10.2	Appropriate feedback.....	144
10.2.1	Feedback on measured gaze point.....	145
10.3	Controllability.....	146
10.3.1	Control over when the gloss appears.....	146
10.3.2	Control over the dictionary entry.....	148
10.4	Unobtrusive visual design.....	149
10.4.1	Visual design decisions in iDict.....	149
11	Evaluation of iDict's Usability.....	151
11.1	Effectiveness – accuracy in getting the expected help.....	152
11.1.1	Assumptions studied in the experiment.....	152
11.1.2	Experiment setup.....	153
11.1.3	Results concerning triggering accuracy.....	154
11.1.4	Feedback used and triggering accuracy.....	156
11.1.5	Language skills and triggering accuracy.....	156
11.2	Efficiency – subjective experience of iDict performance.....	157
11.2.1	Subjective experiences of triggering accuracy and iDict's usefulness.....	157
11.2.2	Preference for the different feedback modes.....	159
11.3	Satisfaction – comparing gaze and manual input.....	159
11.3.1	Assumptions studied in the experiment.....	160
11.3.2	Experiment setup.....	161
11.3.2	Results for different input conditions.....	163
12	Conclusions.....	169
12.1	Tempering the gaze tracking inaccuracy.....	170
12.2	Interpretation of gaze paths.....	171
12.3	Designing gaze-aware applications.....	173
12.4	Concluding remarks.....	174

List of Figures

- Figure 1.1** Taxonomy of eye-movement-based interaction (Jacob, 2003).
Figure 1.2 Prognosis for development of eye tracker markets (J. P. Hansen, Hansen, Johansen & Elvesjö, 2005).
- Figure 2.1** Cross-section of a human eye from above.
Figure 2.2 The visual angle.
Figure 2.3 Distribution of rods and cones in the retina.
Figure 2.4 The acuity of the eye (Ware, 1999, p. 59).
- Figure 3.1** Taxonomy of eye tracking systems (Duchowski, 2002).
Figure 3.2 Taxonomy of attentive gaze-based systems.
Figure 3.3 Eye-R glasses (<http://cac.media.mit.edu/eyears.htm>).
Figure 3.4 Eye-bed (Selker, Burleson, Scott & Li, 2002).
Figure 3.5 An EyeContact sensor (Shell, Vertegaal & Skaburskis, 2003).
Figure 3.6 Eye-sensitive lights (Shell, Vertegaal & Skaburskis, 2003).
Figure 3.7 VTOY (Haritaoglu et al., 2001).
Figure 3.8 EyeWindows (Fono & Vertegaal, 2005).
Figure 3.9 iTourist (Qvarfordt & Zhai, 2005).
Figure 3.10 A wearable EyeContact sensor (Vertegaal, Dickie, Sohn & Flickner, 2002).
Figure 3.11 ECSGlasses (Dickie, Vertegaal, Shell, et al., 2004).
Figure 3.12 ECSGlasses in action (Shell et al., 2004).
Figure 3.13 GAZE (Vertegaal, 1999).
Figure 3.14 GAZE-2 (Vertegaal, Weevers & Sohn, 2002).
- Figure 4.1** Distribution of fixation durations during reading (Rayner, 1998).
Figure 4.2 Distribution of forward saccade lengths during reading (Rayner, 1998).
- Figure 5.1** iDict, a general view of the application.
Figure 5.2 Toolbar shortcut buttons.
Figure 5.3 The two-level help provided by iDict.
Figure 5.4 User profile dialog.
Figure 5.5 Creating a new user profile.
Figure 5.6 Translation feedback dialog.
Figure 5.7 Language dialog.
- Figure 6.1** EyeLink.
Figure 6.2 iView X.
Figure 6.3 Tobii 1750.
Figure 6.4 iDict architecture.
Figure 6.5 Structure of the document tree.
Figure 6.6 Text object masks.
Figure 6.7 iDict test environment.
- Figure 7.1** An example gaze path in reading a passage of text (recorded with iView X).
Figure 7.2 Successfully tracked reading session (recorded with EyeLink).
Figure 7.3 A rising reading path (EyeLink).
Figure 7.4 Resuming vertical accuracy (EyeLink).
Figure 7.5 Ascending reading path (iView X).
Figure 7.6 Global vertical shift of the whole reading path of a line (EyeLink).

- Figure 7.7** Reading paths prior and after the path presented in Figure 7.6 (EyeLink).
- Figure 8.1** A stray fixation (EyeLink).
Figure 8.2 Example of local vertical shift (EyeLink).
Figure 8.3 Vertically expanded masks and the dominating current line's mask.
Figure 8.4 Constantly expanding mask of the current line.
Figure 8.5 An example of a new line event (iView X).
Figure 8.6 Returning to read a line after a short regression to the previous line (EyeLink).
Figure 8.7 Regression to the end of a previous line (EyeLink).
Figure 8.7 Number of transition saccades in new line events.
Figure 8.8 Number of transition saccades in new line events by participant.
Figure 8.9 Distribution of transition saccade lengths.
Figure 8.10 First and last NLE fixation locations.
Figure 8.11 Regressive fixations to the previous line (EyeLink).
Figure 8.12 Regression to previous line followed by new line event (EyeLink).
- Figure 8.13** The first of the three text displayed with single, 1.5, and double line spacing.
- Figure 9.1** Time spent on reading the analyzed session by each participant.
Figure 9.2 Number of problematic words identified by the participants.
Figure 9.3 The average first fixation duration.
Figure 9.4 The average gaze duration.
Figure 9.5 The average total time.
Figure 9.6 The average number of fixations.
Figure 9.7 The average number of regressions.
Figure 9.8 The distribution of regressions
Figure 9.9 Total time threshold and triggered glosses.
Figure 9.10 Personalized threshold and false alarms.
Figure 9.11 Personalized threshold and correctly triggered glosses.
Figure 9.12 Distribution of words in BNC.
Figure 9.13 The total time threshold as a function of word frequency.
Figure 9.14 Triggered glosses with a varying total time threshold.
Figure 9.15 Word length's effect on mean total time.
- Figure 10.1** Gaze cursor reflecting the recorded gaze path.
Figure 10.2 Line marker helping the reader to stay on line.
Figure 10.3 A gloss and dictionary entry for the same word.
- Figure 11.1** Preference of feedback modes.
Figure 11.2 Preference of input conditions.

List of Tables

- Table 3.1** Gaze-based attentive systems and applications.
- Table 6.1** Word class information supported by CIE.
Table 6.2 Compound and idiomatic expressions supported by CIE.
Table 6.3 Format of CLM input and output.
Table 6.4 CIE analysis for the example sentence.
- Table 8.1** The number of different new line events identified.
Table 8.2 Performance of the drift algorithms.
- Table 11.1** Measured triggering accuracy in the experiment.
Table 11.2 The effect of different feedback modes on triggering accuracy.
Table 11.3 Subjective opinions of iDict.
Table 11.4 SUS questionnaire results.



1 Introduction

Consider yourself in a situation where you should observe someone's behavior and intentions. Where do you place your attention? Voice, gestures, and facial expressions are surely important, but don't you think that also the person's eyes are high on the list of what you observe? Direction of the gaze, time spent on each direction, and the pace of the eye movements give you pointers to the person's intentions and perhaps even emotional state. If you then imagine a situation in which you are interacting with the person, the role of eyes is even greater.

Visual attention is of cardinal importance in human-human interaction (Bellotti et al., 2002). The gaze direction of others is a powerful attentional cue (Richardson & Spivey, 2004); for example, studies of face-to-face communication show that mutual gaze is used to coordinate the dialogue (e.g., Bavelas, Coates & Johnson, 2002). The ease with which people are able to interact with each other has inspired researchers to apply the conventions of human-human interactions also to human-computer interaction (Qvarfordt, 2004). However, it is not self-evident that we should mimic the interaction between humans when designing human-computer interfaces. Users do not necessarily expect human-like behavior when using a computer application, and, in fact, attempts to mimic human behavior easily lead the user to unrealistic expectations of the application's capabilities in interaction (Shneiderman & Maes, 1997; see also Qvarfordt's (2004) comparison of tool-like and human-like interfaces).

Nonetheless, the benefits gained by following the conventions users are familiar with in their everyday communication are indisputable, since in many cases doing so results in more intuitive and natural interaction. For

example, part of the credit for the success of WIMP¹ interfaces can be given to the use of a direct manipulation (Shneiderman, 1983) interaction style. Combining visible objects and a pointing device lets the users “grab” the object they want to manipulate – a natural action they are accustomed to in real-life situations.

Still, compared to the human-human communication, restricted input devices seem especially wasteful of the richness with which human beings naturally express themselves. The rapid development of techniques supporting the presentation of multimedia content is further exacerbating the existing imbalance in deploying human input and output capabilities (Zhai, 2003). Consequently, there is a broad spectrum of HCI research areas in which versatile approaches are being applied in attempts to find new, natural and efficient, paradigms for human-computer communication. Such paradigms include, for example, speech-based user interfaces, tangible interfaces, perceptual interfaces, context-aware interfaces, and the connective paradigm of multimodal user interfaces.

Attentive user interfaces (AUIs) provide one of the most recent interface paradigms that can be added to the list: in May 2003, *Communications of the ACM* dedicated a special issue to AUIs. What distinguishes the AUI from related HCI paradigms is that it emphasizes **designing for attention** (Vertegaal, 2003). As noted above, eye movements are a powerful source for inferences concerning attention.

1.1 EYE MOVEMENTS AS INPUT FOR A COMPUTER

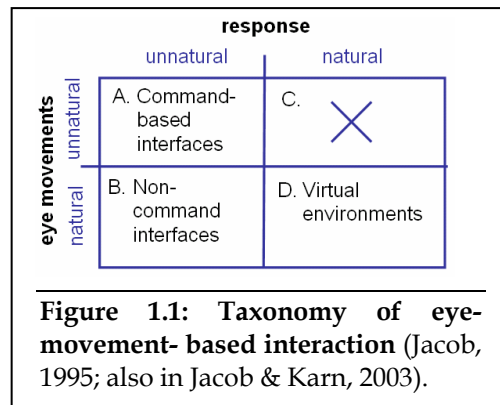
Interfaces utilizing gaze input can be divided into those requiring conscious control of the eyes and those utilizing the natural eye movements of the user in interaction with the computer.

The division can be clarified by the taxonomy of eye-movement-based interaction (Figure 1.1) presented by Jacob (1995; also, Jacob & Karn, 2003). The two axes in the taxonomy are the nature of the user’s eye movements and the nature of the responses. Both may be either natural or unnatural. This distinction in eye movements refers to whether the user is consciously controlling the eyes or not – i.e., whether the user is required to learn to use the eyes in a specific way to get the desired response from the application. The response axis, on the other hand, refers to the feedback received from the application.

¹ WIMP refers to the words “window, icon, menu, pointing device,” denoting a style of interaction using these elements. This type of interaction was developed at Xerox PARC and popularized by the Macintosh in 1984 (van Dam, 1997).

1.1 Eye movements as input for a computer

For example, in command-based interfaces, using the eyes consciously to initiate an action can be considered unnatural eye movements combined with unnatural response (A in Figure 1.1). A prompt given by an educational program counseling to read an unattended text block before proceeding to a new page can be considered as an example of



the case natural eye movements giving unnatural response (B). Unnatural (learned) eye movements with natural response (C) are, obviously, not demonstrable. An example of the last class, of an application giving a natural response to natural eye movements (D), can be found in movement in virtual environments. An early example of such an application is the automated narrator of the Little Prince story implemented by Starker and Bolt (1990), which proceeded with the story according to the user's interest as evidenced by eye movements. To differentiate the two ways of using eyes in the interface, we call applications making use of natural eye movements **eye-aware interfaces/applications** and those applications in which the eyes are used for conscious commands **eye-command interfaces/applications**. The term **eye-based interfaces/applications** refers to both together. In addition, to specially emphasize that an application makes use of the direction of gaze, the word "eye" is replaced with the word "gaze."

Using eyes as an input modality in the interface has some undeniable benefits, but the new modality also brings with it problems and challenges to overcome.

1.1.1 Why eyes?

Disregarding the user's eye movement in the interface loses a vast amount of potentially valuable information: on average, eyes make three to four saccades a second. Eye muscles are extremely fast; the maximal velocity reached by the eye is $450^\circ/\text{s}$ (during a 20° -wide saccade, according to Yarbus, 1967, p. 146). Hence, their speed is superior to that of any other input device. An early experiment performed by Ware and Mikaelian (1987, verified by, e.g., Sibert & Jacob, 2000) showed that in simple target selection and cursor positioning operations eyes performed approximately twice as quickly as conventional cursor positioning devices did (provided that the target object was not too small).

In some cases, the hands may be engaged for other tasks. One example of such a case is an application designed by Tummolini, Lorenzon, Bo & Vaccaro (2002), which supports the activities of a maintenance engineer in

an industrial environment. When working in the environment, the user must keep the hands free to work with the target of intervention. Using speech commands in such situations is often restricted either due to background noise or because using the voice may be undesired for social reasons.

One benefit of eye input derives from the fact that eye movements are natural and effortless. At present, the mouse and keyboard are the main devices used for giving input for a computer application. This results in a lot of repetitive routine tasks, such as typing, positioning the mouse cursor, clicking, double-clicking (which requires extra concentration in order that the mouse is not moved between the clicks), and repetitive switching between mouse and keyboard. Those tasks contribute to occupational strain injuries of the hand and wrist¹. Transferring some of the manual tasks to the eyes helps to reduce the problem.

For an important group of users the physical limitations are more dramatic than strain problems. For the people whose life has been impaired by motor-control disorders the eye input may give substantially easier, or in some cases even the only mean to interact with the surroundings. Motor neuron diseases (MND), such as ALS or locked-in syndrome, are quite common: there are nearly 120,000 cases diagnosed world wide every year² (see also the EU supported network concentrating on the subject, COGAIN, 2004).

For many disabled users who are unable to use manual input devices, there are optional methods, such as so-called head mice, that permit the user to address a point on the screen with head movements alone. A head mouse, compared to using eye tracking, may perform better as a pointing device for many users because it provides (at least at the moment) a simpler, cheaper, and perhaps even more accurate approach (Bates & Istance, 2003). However, head mice are reported to cause neck strain problems (Donegan et al., 2005), and some user groups are unable to perform the head movements these devices require.

Finally, we wish to emphasize the one remarkable feature unique to eyes only: use of the point of gaze as an input source for the computer is the only input method carrying information on the user's momentary focus of

¹ According to the 2004 Eurostat yearbook (*Work and health in the EU*, Eurostat, 2004), there were about 20,000 recognized wrist- and hand-related musculoskeletal occupational diseases (tenosynovitis, epicondylitis, and carpal tunnel syndrome) in 15 European countries in 2001. Report available at http://epp.eurostat.cec.eu.int/cache/ITY_OFFPUB/KS-57-04-807/EN/KS-57-04-807-EN.PDF (April 26, 2006).

² Information given by the International Alliance of ALS/MND Associations at the page <http://www.alsmndalliance.org/whatis.html> (April 26, 2006).

attention.

1.1.2 Problems of eye input

In both gaze-command and gaze-aware applications, the major problems include difficulties in interpreting the meaning of eye movements and problems with accuracy in measuring eye movements.

In considering command-based interfaces, we easily arrive at the idea of using the point of gaze as a substitute for the mouse as the pointing device – for example, to select the object being looked at. However, since an eye is operated in a very different manner than a hand is, the idea soon collides with problems.

The nature of eyes as a perceptive organ involves a problem Jacob (1991) labeled the **Midas touch** problem: since “eyes are always on” their movements get easily interpreted as activations of operations even when the user just wants to look around. The twofold role of the mouse in conventional interfaces is to function as a pointing device for assigning a target location (cursor positioning) and to select an action at the assigned position (clicking). The Midas touch problem manifests itself in both cases. If gaze is used to control the cursor position, the cursor cannot be left “off” at a position on-screen while the visual attention is momentarily targeted to another (on- or off-screen) target. If gaze is used as a selection device, the absence of a “clutch” analogous to the mouse button is a problem. “Dwell time” (prolonged gaze indicating the selection) and eye blinks have been used for this purpose. Though usable in some situations, these may generate the wrong selections and make the user feel uncomfortable, preventing the user from performing natural, relaxed browsing.

The other significant problem is the inherent inaccuracy of the measured point of gaze; Bates and Istance (2003) here refer to positional tolerance. The deduction we make in Chapter 2 is that the accuracy of the measured point of gaze can never equal the accuracy of the mouse. In command-based interfaces, this implies, for example, that the selectable objects in normal windowing systems (menus, toolbar icons, scrollbars, etc.) are too small for straightforward gaze selection.

Also, inaccuracy is a problem in using natural eye movements. More generally, interpretation of eye movements is a nontrivial problem, especially when natural eye movements are used. In which form should we transmit the eye movements received from an eye tracker to the application? In some cases, more often in gaze-command applications, it may be enough to send the “raw data points” on to the application. In these cases, the application receives solitary gaze positions received at the rate of the tracker’s temporal resolution. In current commercial eye trackers, the temporal resolution varies from 15 Hz to 1000 Hz, which quickly multiplies the quantity of data to be handled in the application.

The stream of raw gaze positions is also noisy, which means that in most cases the data must be preprocessed before transmission to the application.

Lastly, usability and availability are issues in eye tracking devices' disfavor. Even though the trackers have developed a lot since the days when Bolt (1980, 1981, 1985) first experimented with using gaze input (the "Put-that-there" and "Gaze-orchestrated windows"), they still require much more patience from the users than do other input devices. Also their price range is of different magnitude from that of most other input devices. Some economical devices (less than 5,000 euros) are available, but prices for high-quality trackers easily reach 20,000 euros.

1.1.3 Challenges for eye-based interaction

Eye tracking has been referred as having "promising" potential to enhance human-computer interaction already for about 20 years. Nevertheless, to date the situation has profoundly remained the same: eye tracking has still not yet delivered the promises (Jacob & Karn, 2003). Should this be taken as a proof that eye tracking is not viable enough and worth putting research efforts on?

A retrospective glance at the evolution of the mouse provides perspective for answering the question. Even though the mouse is technically a relatively simple device, it took more than 20 years from the days of Douglas Engelbart's early experiments in the early '60s before the mouse was popularized by its inclusion as standard equipment with the Apple Macintosh in 1984. We believe that eye tracking devices could someday belong to the standard setup of an off-the-shelf computer package, as the mouse does today. Movement toward this goal seems to be slow, however. We believe the main reasons hindering the evolution process are that

- available interaction techniques are not able to take advantage of the device,
- usability of eye tracking devices is poor, and
- they are expensive.

We now take a look at each of these issues.

New interaction techniques required

As was seen with the mouse, the penetration of a new input device is retarded due to the fact that it is not supported by the prevailing interaction paradigms. Consequently, it takes a lot of effort from developers of applications to use eye trackers, since at low level the development environments do not provide standard support for them. Further on, this results in poor portability of eye-based applications when

head. The new application field sets totally different demands concerning acceptable levels of intrusiveness. The user should be able to start using an eye-based application in the same way as any other application, just by opening it to use, and should also be able to move freely while using the application. The eye trackers that exploit remote (usually in the proximity of the screen) video cameras and track several features of the eyes so as to compensate for head movements are approaching such a standard.

Robustness of use. Eye trackers' reliability in reporting gaze position is still very vulnerable to outside effects. For example, different lighting conditions, specific eye features¹, and corrected vision (eyeglasses or contact lenses) often result in failure to track the eyes. Several less lighting-sensitive and more robust techniques have been suggested and are under development (Ebisawa, 1995; Morimoto, Koons, Amir, Flickner & Zhai, 1999; Morimoto, Koons, Amir & Flickner, 2000; Zhu, Fujimura & Qiang, 2002; Ruddaraju et al., 2003; D. W. Hansen & Pece, 2005). At the moment, eye input is constrained to desktop applications. Even though some preliminary attempts (Lukander, 2004; Tummolini et al., 2002), have been made to develop portable eye tracking solutions - taking eye tracking into "real-world" environments - portable eye tracking is very difficult (Sodhi et al., 2002). If they can be implemented, they would yield many more possibilities for eye-based applications.

One of the recent improvements, consequent of increasing computing power and improved camera optics, is that eye trackers are moving toward using large-field-of-view cameras (e.g., Vertegaal, Dickie, Sohn & Flickner, 2002; LC Technologies, 2005; Tobii Technology, 2005) instead of focusing on the camera image of the eye only. With the more recent approach, the eye can be more easily located after body and head movements without the need for servo mechanisms that try to follow the eye.

Ease of (or no) calibration. Current eye trackers require a calibration routine to be performed before they are able to detect the user's point of gaze. Through calibration, the tracker is taught the individual characteristics of each user's eyes: how the eyes are positioned when different parts of the screen are being looked at. The calibration is performed by requesting the user to follow the reference points appearing on the screen, in five to 17 (Donegan et al., 2005) different positions. Some techniques have managed to decrease the number of points needed to two (Ohno, Mukawa & Yoshikawa, 2002; Ohno & Mukawa, 2003; Villanueva, Cabeza & Porta, 2004). Most trackers need to be calibrated at the beginning of each session,

¹ For example, different ethnic features related to the eye make the tracking of some users harder (Nguyen, Wagner, Koons & Flickner, 2002).

1.1 Eye movements as input for a computer

and, since the accuracy of the calibration usually decreases during the session, often the routine has to be done repeatedly every few minutes. The need for calibration is one of the issues that should be given extra attention. Standard users will probably consider turning the eye tracker off if repetitive calibration is the other option.

Some trackers¹ support persistent calibration, in which case the calibration has to be performed only once, when the tracker is used for the first time. In subsequent sessions, the saved personal calibration data can be retrieved automatically; of course, this calls for some kind of login to identify the user. This already is a huge improvement, but since the calibration can subtly lose its accuracy, possibilities for automatically correcting it during sessions should be more thoroughly studied. Again, a review of the mouse's development reminds us that these devices too had to be calibrated in earlier stages of development (Amir, Flickner & Koons, 2002). The calibration of a mouse is now invisible to the user. Also, research on totally calibration-free tracker use is in progress (Shih, Wu & Liu, 2000; Amir et al., 2003; Morimoto, Amir & Flickner, 2002).

As a conclusion from the above, we can fairly assume that recent technical improvements and the ongoing research will eventually solve the three main usability problems. At the least, we are justified in expecting future eye trackers to be substantially easier to use than present ones.

Cost-effective eye tracking

The expensiveness of eye tracking devices derives from the fact that the volume of devices purchased is marginal at the moment, leaving the price dominated by development costs. The chicken-and-egg dilemma of eye tracking was recognized early on by Bolt (1985). With mass marketing, the cost could decrease to the hundreds, rather than today's thousands, of euros. The key factor for getting the cost to such a level that eye trackers could be included in a standard computer setup is to increase the volume of market demand. At the same time, increasing the demand calls for less expensive equipment. This is an unfortunate dilemma, since using eye input is of substantial importance for many disabled users.

Lowering the costs calls for a less narrow user base. A greater number of applications making use of eye input would increase the market for the equipment and thus decrease the production cost. So, a few general-purpose breakthrough applications could resolve the dilemma and lead evolution into the positive cycle of reducing costs and increasing the number of eye-based applications. Figure 1.2 presents one possible prognosis for development of eye tracker markets, given by J. P. Hansen,

¹ Tobii, <http://www.tobii.se/>.

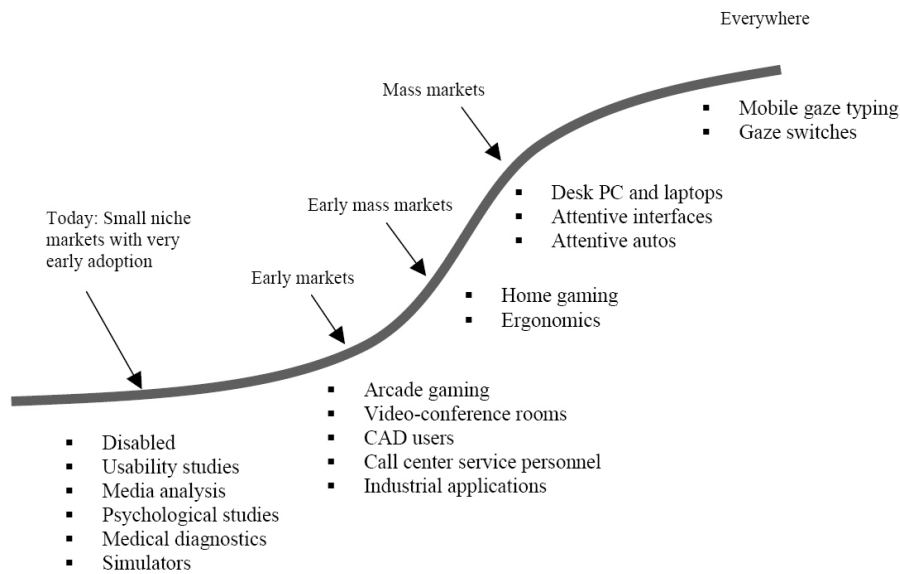


Figure 1.2: Prognosis for development of eye tracker markets (J. P. Hansen et al., 2005).

Hansen, Johansen & Elvesjö (2005).

At the moment, eye trackers are used mostly as analysis tools and also as augmentative devices for the disabled. J. P. Hansen et al. (2005) assume that the mass markets can be reached in an increasing variety of application domains.

There are also ongoing attempts to break from the dilemma by studying whether off-the-shelf web cameras could be used to give the gaze position information for an application (Corno, Farinetti & Signorile, 2002; Corno & Garbo, 2005; Frizer, Droege & Paulus, 2005). This development could play a key role in solving the dilemma.

1.2 RESEARCH METHODS AND RESEARCH FOCUS

This dissertation concentrates on studying the prospective benefits of using information on the user's natural eye movements in attentive interfaces.

The research methods used combine constructive and experimental research. We implemented a test-bed gaze-aware application, iDict. Solutions for overcoming the difficulties encountered were developed on the basis of results from experiments with the application. The performance and user experiences of the application were then evaluated. The iDict application is described in detail later, but below we introduce its key ideas in brief.

attentive application iDict. Designing and implementing iDict gave us insight of the use of eye tracking in creating gaze-aware applications in general. The most fundamental problems we encountered when trying to detect deviations from the normal flow of reading can be articulated with the two main questions **where** and **when**. The third essential question is **how** the application should react when the probable cause of digressive reading is identified.

The first class of problems arises from the limited tracking accuracy involved in eye tracking. Do we have to use abnormally large font sizes for the application? While problems with limited accuracy were anticipated, overcoming these required even more effort than was expected. Most gaze behavior studies use posterior analysis of gaze position data, which makes the job easier. When the gaze path is known in full – after the fact – it is much easier to determine the target of visual attention. In our case, this must be done immediately, in real time.

The other class of problems has to do with answering the question of when the application should provide help for the reader. What are the clues we can use to detect when the reader has difficulties comprehending the text?

As an answer to the third question (that of “how”), we summarize the design principles of a gaze-aware application that we formulated on the basis of the case study.

1.3 OUTLINE OF THE DISSERTATION

The rest of this dissertation is organized into three parts as follows.

PART I: BACKGROUND

Provides the reader with background knowledge for understanding the work. First, the biological and technical issues relevant to using natural eye movements in human-computer interaction are explained. Then, we introduce the role of eyes in attentive interfaces and review existing gaze-aware applications. Since our application tracks the reading process, a review of relevant reading research is given as well.

Chapter 2 Gaze tracking

Chapter 3 Gaze in attentive interfaces

Chapter 4 Eye movements in reading

1.3 Outline of the dissertation

PART II: THE iDICT APPLICATION

Introduces the iDict application. Its functionality from the user's perspective and the implementation issues are presented.

Chapter 5 iDict functionality

Chapter 6 iDict implementation

PART III: USING GAZE PATH TO INTERPRET READING IN REAL TIME

Describes how gaze paths are interpreted in iDict. An analysis of the problems caused by the inaccuracy of gaze tracking is presented and the development of the solutions to deal with the inaccuracy is described. The development of the function that triggers the assistance for the reader, and the lessons learnt of interaction design of gaze-aware applications are summarized. Finally the evaluation of the usability of the application is reported.

Chapter 7 Inaccuracy of eye tracking

Chapter 8 Keeping track of the point of reading

Chapter 9 Recognizing reading comprehension difficulties

Chapter 10 Interaction design of a gaze-aware application

Chapter 11 Evaluation iDict's usability

Chapter 12 Conclusions

The last chapter sums up the contributions of the dissertation and provides the conclusions that can be made on the basis of the work.



Part I

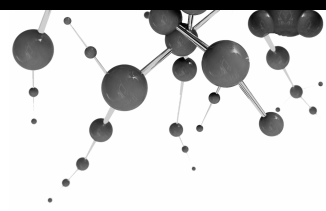
Background

Chapter 2 Gaze Tracking

Chapter 3 Attention in the Interface

Chapter 4 Reading and Attention





2 Gaze Tracking

Even if using eyes in the user interface is a new branch of eye tracking research, research on eye movements itself has a long history. Eye movements have fascinated researchers for decades. Most of the research in this area has been performed by psychologists interested in human sensory and motor systems, in both physiological and psychological details of the human vision system. In this chapter, gaze tracking is reviewed from the perspective of using eye movements as a component of human-computer interaction.

2.1 BIOLOGICAL BASIS FOR GAZE TRACKING

It is surprising to discover that most of the basic observations that still apply today in eye movement research had been made at the turn of the last century. For example, Emile Javal (1839–1907) made the observation that eyes do not move smoothly but make rapid movements from point to point; he called those movements *saccades*¹.

Although, according to Wade, Tatler & Heller (2003), introducing the term is still acknowledged as Javal's contribution (solidified by Dodge in 1916), recent historians have traced early eye movement research much further back in time. A historical review of eye movement research can be found in the book *A Natural History of Vision* by Nicholas Wade (2000). Reviews concentrating on a more recent history of eye tracking and eye movement research are given by, e.g., Paulson and Goodman (1999), Jacob and Karn (2003), and Richardson and Spivey (2004). Also Rayner and Pollatsek

¹ *Saccades are one specific type of eye movement, introduced in Section 2.1.2.*

(1989) and Rayner (1998) give thorough and insightful reviews of the history of eye movement research, though written from the perspective of research carried out in the context of reading.

These reviews report a versatile range of techniques that have been, and in some cases still are, used for tracking eye movements. However, we are not interested in eye movements per se but rather in gaze tracking. That is why we use the term “gaze tracking” (instead of “eye tracking”) when the essential issue is measuring the **direction of gaze** and – even more accurately – the point of gaze. How do we get from observing the movements of the eye to information on the point of gaze?

In order to understand that, along with the limitations of gaze tracking, we first need to know some facts about human vision. After introducing the essential particulars of human vision, we will briefly summarize the eye movements that are relevant for us (Subsection 2.1.2). In Section 2.2 the techniques used for gaze tracking are then briefly introduced.

2.1.1 Human vision - physiological background for gaze tracking

The basic knowledge we have of the human vision system is explicated in many psychology books that introduce sensory systems (e.g., Deutch & Deutch, 1966; Kalat, 1984; De Valois & De Valois, 1990; Wandell, 1995; Ware, 2000). The subsequent short introduction to vision concentrates on details that are relevant when the aim is to estimate the point of gaze by monitoring the movements of the eye.

The techniques used for gaze tracking are based on estimation of the perception of the image that is transmitted from the transparent cornea through the pupil, the lens, and the vitreous humour on to the retina (Figure 2.1). The iris, which borders the pupil and gives us the color of our

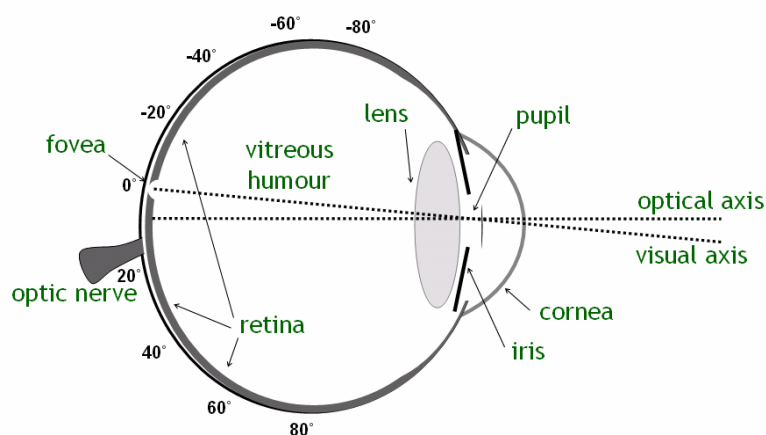


Figure 2.1: Cross-section of a human eye from above.

eyes, dynamically regulates the peripheral entry of light entering the eye and thus protects the light-sensitive retina from too bright light.

2.1 Biological basis for gaze tracking

When we want to exploit eye movements in human-technology interaction, we are interested in the focus of the gaze. Thus, we should know the user's perceived image at each point in time. How are we able to make an estimation of the image and how accurate the estimation is?

Visual angle

Focusing of the target image is performed in three dimensions. The depth focus is received by changing the shape of the lens. When the eyes are targeted on an object close to the eye, the lens is thick. When the muscles controlling the eye are at rest, the lens is flat and the focus is distant. The iris can also improve the focus; the smaller the pupil the sharper is the projection of the target image on the retina. In observing an image on a computer screen, the depth dimension stays relatively constant. To consider focusing the eye on the other two dimensions, on a vertical plane in front of the eye, we first need to introduce the concept of **visual angle**.

The visual angle, α (see Figure 2.2), is the angle that sends light from scene s through the lens onto the surface of the retina. Given d , the distance from lens to scene, the visual angle α can be calculated from the formula

$$\alpha = 2 \arctan \frac{s}{2d}.$$

One of the most handy rules of thumb for estimating the visual angle is the thumb itself: a thumb covering a scene with a radius of 2–2.5 cm at a distance of 70 cm (about an arm's length) equals a visual angle of 1.2–1.5 degrees.

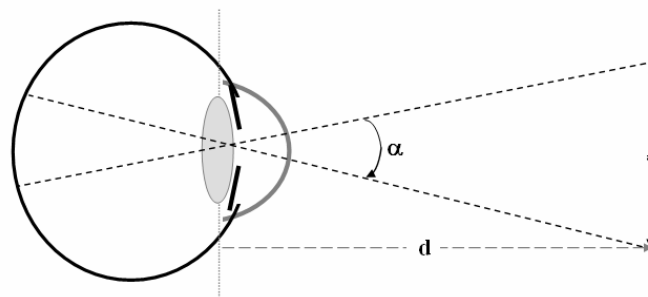


Figure 2.2: The visual angle. The visual angle α of the perceived scene s from a distance d .

Visual field

The visual angle of the view imaged on the retina surface, the **visual field**, is horizontally about 180° and vertically about 130° (De Valois & De Valois, 1990). For our purposes, there is not much use for the knowledge that the subject is able to see $180^\circ \times 130^\circ$ of the scene in front of the eyes at a given point in time.

Fortunately, we know that the retina contains two fundamentally different types of photoreceptors that get stimulated to transmit the perceived image further on to the nervous system (via the optic nerve, Figure 2.1). There are about five million cones and 100 million rods in the retina (Wandell, 1995, p. 46)¹. The cone receptors are able to transmit a highly detailed image with color (actually, there are three types of cones, sensitive to different light wavelengths). In turn, the rod receptors are more sensitive to dim light and transmit only shades of gray.

The fact that we are able to deduce the direction of gaze to be around the visual axis is due to the uneven distribution of the receptor cells across the retina. The fovea (the pit of the retina) is densely packed with cones, and hence the image entering the fovea is perceived the most sharply. The density of the cones decreases sharply right from the center of the fovea (Figure 2.3). As is illustrated in the figure, the center of the fovea contains no rods.

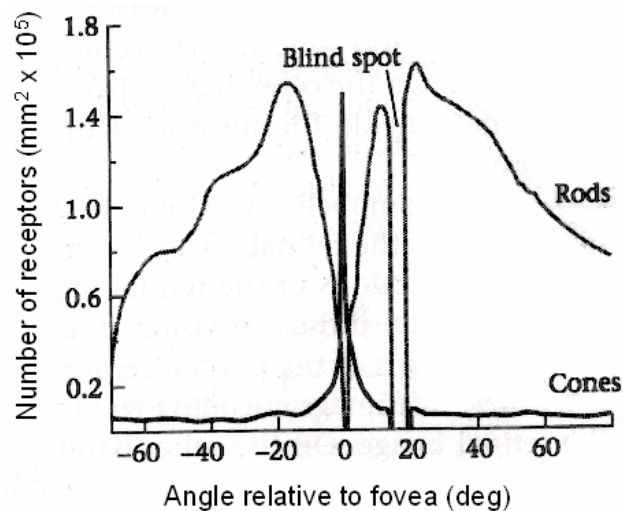


Figure 2.3: Distribution of rods and cones in the retina. (Prienne, 1967, as cited by Haber and Hershenson, 1973, p. 25; also in Wandell, 1995, p. 46).

Some of us may have experienced situations where a dim light source, such as weak starlight, appears to vanish when we look straight at it. The absence of light-sensitive cones in the fovea explains this phenomenon.

Our blind spot (the optical disk) is the spot where the optic nerve leaves the retina. The blind spot contains neither rods nor cones.

¹ The figures vary from one source to the next (possibly caused by either variation in the measuring technique or individual variations in the density of the receptors in the retina).

2.1 Biological basis for gaze tracking

Visual acuity and the visual field

The ability to perceive spatial detail in the visual field is termed **visual acuity**. Limits of the visual acuity may be either optical or neural in nature (Westheimer, 1986, pp. 7–47). The optical limits are due to degraded retinal image and can usually be compensated by corrective lenses. Neural limits are derived from individual differences in the retinal mosaic (the distribution of photoreceptors across the retina). Visual acuity has been studied in numerous experiments, resulting in measurements expressing the visual acuity of an individual.

An individual's visual acuity decreases with age. Typically the visual acuity of a young person is on the order of minutes of a visual angle, sometimes even seconds of the angle (a minute is 1/60 degree and a second is 1/60 minute). For example, the point acuity (the ability to

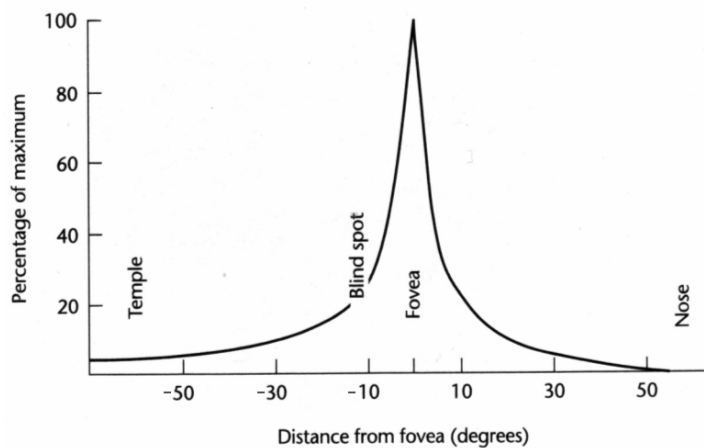


Figure 2.4: The acuity of the eye (Ware, 2000, p. 59).

differentiate two points) is about one minute of arc, the letter acuity (the ability to resolve letters) is five minutes of arc, and the vernier acuity (the ability to see whether two line segments are collinear) is 10 seconds of arc (Ware, 2000, p. 57). The visual acuity of the human eye falls off rapidly with distance from the fovea (Figure 2.4).

Even though visual acuity is measured in minutes (or in seconds), this does not mean that we can compute the focus of gaze with such accuracy. The focus of gaze cannot be considered to be a sharp point on the screen (or, more generally, in the visual field): when a point on the screen is projected into the center of the fovea pit, a person can still perceive sharply also the surrounding areas projecting onto the rest of the fovea area on the retina.

Moreover, it has been suggested that the visual attention can be shifted to some extent without the necessity of moving the eyes (see, e.g., Yarbus, 1967, p. 117; Posner, 1980; Groner & Groner, 1989; Rayner, 1998; Coren, Ward & Enns, 1999, p. 437). This means that even if we can compute the

exact point of a scene that is projected into the center of the fovea pit, the point of the person's visual attention may be focused somewhat off from this point. The visual angle that the fovea covers, and thus the amount of potential error of the measured focus of visual attention, is reported to be about 1° (in, e.g., Haber & Hershenson, 1973; Jacob, 1995; Bates & Istance, 2003; Jacob & Karn, 2003) or about 2° (e.g., Groner & Groner, 1989; Rayner, 1995; Duchowski, 2003). No absolute truth as to the size exists, since the fovea itself is a somewhat artificial concept; visual acuity does not suddenly drop at a certain point from the fovea.

One interesting issue of research for developers of gaze tracking, especially in gaze-command interfaces, is how accurately a person is able to position a point in the visual field within the center of the fovea. Our capability to see an image clearly in the fovea area does not necessarily mean that we are not able to control our eyes more accurately. According to Yarbus (1967), an observer cannot voluntarily perform saccades shorter than a certain threshold length. In one of his experiments, it was found that a subject was unable to change the point of fixation when the distance between the reference points was eight minutes. The ability to focus on a point has some limit between the visual acuity and the area rendered in the fovea (something from eight minutes to two degrees). As far as we know, so far no studies have focused on answering the question of what the accuracy is of voluntary controlled fixations. This would give interface designers for gaze-command interfaces the biological accuracy limit, given that eye trackers are going to develop to give exact accuracy without measurement errors.

In the next section, we discuss how eyes move when observing the environment. The aim is not to provide a complete survey of eye movements but to convey the basic knowledge needed for interpreting the gaze point data received from an eye tracking device. A more thorough introduction to eye movements is given by, for example, Yarbus (1967) and Young and Sheena (1975).

2.1.2 Movements of the eyes

The need to keep the parts of the visual environment we want to see in detail projected on the high-resolution fovea lays the ground for our eye movements. Eye movements can be divided into three main categories (Haber & Hershenson, 1973; Ware, 2000): (1) saccadic movements, (2) smooth pursuit movements, and (3) convergent movements. Smooth pursuit movements occur when eyes follow an object moving in the visual field or when correcting body or head movement to maintain the focus on the object. Convergent eye movements keep both eyes focused at the target of our visual attention independently of its distance from our eyes. From our perspective, saccadic eye movements are the most important.

Normally (smooth pursuit movements are an exception) the eyes do not move smoothly when targeting an image of an object into the fovea. The movement is performed with saccades, sudden jumps, from one target point to another. Saccades are fast, ballistic¹ movements, and the saccade latencies (i.e., the pauses between saccades) are called fixations. The perception of visual objects occurs during fixations; during saccades, the signals from the eyes are, at least partially, inhibited (Wandell, 1995, pp. 373–375, Gregory, 1997, p. 47; Ware, 2000, p. 153). The durations of saccades and fixations depend on the task the user is performing, but typically saccades are reported to last less than 100 ms, and the durations of fixations from 100 ms up to about 500–600 ms. For example, in reading, the average fixation duration is 250 ms (Rayner, 1995).

However, the eyes are not totally stable during the fixations, either. In addition to the three main types of eye movements, the eyes make smaller movements, sometimes called miniature eye movements, also during fixations. During a fixation, the eyes slowly drift from the fixation point, and after a while a microsaccade rapidly jerks the focus back toward the fixation point (Haber & Hershenson, 1973). The drift is considered to be essential in keeping the vision system active: if the image is artificially stabilized at the retina, it gradually disappears. Microsaccades vary from two to 50 minutes of visual angle, and their duration is 10 to 20 milliseconds (Yarbus, 1967, p. 115). During fixations, in addition to the drift and the corrective microsaccades, there is a small, constant physiological tremor in trying to hold the eye's position by balancing the forces of several strong muscles pulling the eyeball. The existence of these small movements within fixations makes the identification of a fixation more complicated than it would otherwise be.

2.2 EYE TRACKING TECHNIQUES

There are several techniques that can be used for monitoring eye movements. The thorough review of the techniques made by Young and Sheena (1975) is still for the most part valid. In a more recent survey, Collewyn (1998) reviews the principles and practice of different techniques used for recording eye movement. The techniques can be divided into three classes:

¹ The term "ballistic" refers to the assumption that the destination of a saccade is preprogrammed; that is, once a saccade is started, its destination cannot be altered during the "jump."

1. those using electro-oculography (EOG) techniques, which measure differences in electric skin potential around the eye,
2. those requiring physical connection to the eye, including
 - purely mechanical devices,
 - optical lever devices, and
 - scleral coil techniques,
 and the ones using
3. non-contact camera-based methods.

In eye-based interaction, researchers at the moment almost invariably use camera-based techniques.

By “non-contact camera-based techniques” we refer to all eye tracking systems that use cameras to take consecutive images of the user’s eye and, additionally, require no direct contact devices added to the eye. The images are processed to extract the location of some traceable feature, or several features of the eye are identified. The location is then used to compute the movement of the monitored eye in between the images obtained from the camera.

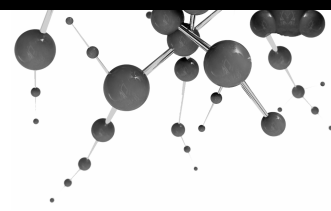
As we have noted, our interest is in gaze, not in sheer eye tracking. If head movements are allowed (one of the usability requirements we addressed in the introduction was unobtrusiveness), tracking only one element – for example, the pupil – is clearly not enough for obtaining the point of gaze. In this case, the system has to involve an additional tracing mechanism addressing head orientation. However, adding the tracking of a reflection point generated by sending a light beam to the eye (indistinguishable, near-infrared light sources are used for this purpose) does allow disassociation of eye rotation from head movements. The observation was made in the early '70s by Cornsweet and Crane (as reported by Jacob and Karn in 2003). Consider a light beam sent to a special surface (the cornea): the reflection point does not change, although the permanent features on the surface (e.g., the pupil) do. Allowing totally free head movements is not that simple, but the observation was the basis for greater freedom of movement for the person being tracked. Nonetheless, even if techniques for compensating for head movements in order to recapture the gaze location were developed to perfection, the issue still adds a new factor for inaccuracy in gaze tracking. Studies have revealed that human “head movement compensation” in order to keep gaze position fixed is imperfect: eye rotation compensates for head movements by only 95–98 percent (Kowler, 1990, p. 10).

2.2 Eye tracking techniques

The camera-based techniques have a lot of variations, depending on which eye features are tracked and whether an infrared reflection point is tracked. If the reflection point is tracked, the light sent to the eye reflects back from four different layers of the eye: (1) the front surface of the cornea, (2) the back surface of the cornea, (3) the front surface of the lens, and (4) the back surface of the lens. These are called the first, second, third, and fourth Purkinje images, respectively. However, the most commonly used technique is pupil-center-corneal reflection (sometimes referred as the PCCR technique), which uses the center of the pupil and the first Purkinje image as reference points for calculating the point of gaze. Near-infrared light sources are used in camera-based techniques, not only for the reflection point but also to help in recognizing the pupil from the picture of the eye. When the light beam is sent from a direction close enough to the camera (on-axis light beam), it brightens the pupil, and when the light beam is sent from other directions, the pupil appears very dark in the picture taken from the eye. Camera-based techniques often use the bright pupil response for eye detection (Nguyen et al., 2002), but also the technique of using them both together as suggested by Ebisawa (1995) has been used (Morimoto et al., 2002).

Further still, camera-based techniques can use either head-mounted or remote optics, depending on whether the cameras are attached to the subject or positioned remotely somewhere near the screen. Remote-optics eye trackers are, evidently, those with the best prospects in human-computer interaction.

.....



3 Attention in the Interface

Attention is a limited human resource. The development of information and communication technologies has increased the problem of information overload in today's computerized working environments. People are challenged to access and exploit information quickly and efficiently. The competition for our attention increases; yet our capacity for processing the incoming flood of information remains limited.

In this chapter, we first introduce the grounding psychological foundations of attention and its relationship to eye movement and discuss how they should be taken into account in the interface design (Section 3.1). In this dissertation, we are interested especially in the role of gaze in attentive user interfaces. In the second section (3.2), we provide a review of the main systems and application domains where gaze has been used to get information on the user's attentional state.

3.1 ATTENTION IN USER INTERFACES

Our sensory organs constantly pass us a huge amount of information from our environment. Concentrating on (or assigning mental processing power for) every stimulus received is not possible; some kind of selection of things to be processed further must take place. This cognitive process of selectively concentrating on one thing while ignoring others is referred to as attention. One branch of cognitive psychology has involved studying how the process is performed (for a review, see, e.g., Pashler, 1998; Coren et al., 1999). While many details of the theories related to attention are still disputable, on a coarse level there is wide consensus on the main observations concerning attentional processes. Those observations include how the attention is oriented and how it is controlled. After introducing

these aspects of attention, we discuss their implications for the field of human-computer interaction.

3.1.1 Orienting attention

The metaphor of a spotlight¹ is often used to illustrate the fact that our attention is limited to focusing on only one target at a time. Although visual attention is acknowledged to have a close relationship with the focus of attention, that is not always true. The focus of attention may vary independently of where the eyes are looking.

This distinction of visual attention from the general focus of attention is generalized with the concepts of **overt** and **covert attention** (Posner, 1980). When we attend to a task, like reading this text, our attention may be drawn (voluntarily or, in many cases, involuntarily) to some other issues, even though we still keep our eyes on the text. For example, the phrase “mind’s eye” used in the text may remind us of some previously read article and lead us to ponder whether the term was used there in the same sense as in this text. As another example, the ongoing discussion next door may suddenly attract our attention since we hear someone speaking out our name. Thus, overt attention refers to changes of attention that can be observed from our head and eye movements (concentration on the text), and covert attention refers to the more general “internal” focus of attention (analyzing the term “mind’s eye” or the discussion next door). By definition, only overt attention can be observed from eye movements. How is the shift of attention controlled?

3.1.2 Control of attention

We noted above that the orienting of our attention may be voluntary or involuntary. In fact, experimental psychology studies have confirmed that there are two types of control mechanisms controlling our shift of attention (Pashler, Johnston & Ruthruff, 2001; Wolfe, 1998): **top-down processing** (also referred to as endogenous or goal-driven processing) and **bottom-up processing** (also called exogenous or stimulus-driven processing). From the examples above, the first distraction of attention can be considered a top-down shift of attention, since it was driven by the internal goal of understanding the concept of “the mind’s eye” that was presented. Hearing our name was a distinct stimulus that caught our attention and is hence an example of bottom-up-driven shift of attention.

While these two coarse mechanisms are widely recognized, there are several more detailed theories on how the filtering of irrelevant information is processed. However, common to most theories is that the

¹ The terms “attentional gaze,” “zoom lens,” and “the mind’s eye” are used to denote the same notion (Coren, Ward & Enns, 1999).

entire visual field of information is “preprocessed” in a preattentive stage. During this preattentive stage, parallel processes segment the visual information into clusters to form separate objects. Gestalt laws, such as proximity, closure, similarity, and good continuation, are applied during this stage. After that, conscious, active attention is paid to the objects sequentially, one at a time (Duncan, 1984). Some distinct features in the visual field, like color, shape, and size (Wolfe, 1998), or objects with abrupt onset (Yantis & Jonides, 1990), have been shown to increase the probability of passing the preattended target to the conscious attention. Motion has been found to be an especially powerful attribute for activating the focus of attention (see, e.g., Abrams & Christ, 2003; Bartram, Ware & Calvert, 2003; Franconeri & Simons, 2005).

If we are able to focus our attention on only one object at a time, how can we explain the everyday situations where we can execute many tasks simultaneously? We are, for example, able to drink coffee and still unbrokenly read a newspaper. This is enabled by habit development (Raskin, 2000, pp. 18–20), or **automaticity**, in the language of cognitive scientists. Repeatedly performed tasks gradually become automatic, the kind of habitual reactions we often are unable to avoid. When we perform several simultaneous tasks, all but one of them are automatic. The one that is not automatic is the task that most often involves the focus of visual attention (Raskin, 2000, p. 21).

3.1.3 Implications for interface and interaction design

The main thread in considerations of human attention is that it is a limited resource; only one task at a time can reserve the user’s active attention. The prevailing graphical point-and-click interfaces, as well as most of the more recent interface metaphors, are totally uninformed about the user’s attentional state. The importance of taking this human limitation into account has recently been recognized; for example, notable publications have devoted a special issue to the subject (*Communications of the ACM* 46(3), 2003; *International Journal of Human-Computer Studies* 58(5), 2003; and *Computers in Human Behavior* 22(4), 2006). User interfaces that are able to sense or work out the user’s attention can assist the user to manage the increasing information overload. We can identify at least two different ways in which attention-awareness can be exploited in applications.

First, knowledge of the bottom-up processes in attention control can be used to guide the user’s attention through desired paths of task execution. Actually, in a broad sense, interface designers have been doing this for a long time – for example, by taking advantage of Gestalt laws or by applying the “less is more” guideline (the less irrelevant information to distract the attention, the better the user’s attention is under control). Also, interface designers have been aware for some time of the effects of automaticity. For instance, they keep the positioning of “OK”/“Cancel”

buttons consistent across dialog windows because their frequent appearance easily inspires a habitual reflection. However, they have not directly exploited knowledge of the user's real attention. For example, unobserved information crucial for completing a step in a task could try to draw the user's attention until it is noticed. Thus, the signaling would be performed only when the application knows the user has not yet paid attention to the relevant material.

On the other hand, if knowing the user's focus of attention, the application can be designed to adapt better to the user's behavior. An example might be a Web browser that could fetch a target page into the cache in advance if the user seems to be paying attention to a link to the page or information on it, thus making the prospective loading happen more smoothly.

Non-command, transparent, and proactive applications

The growing interest in the "new input channel" of attention has some interesting connections to more well-known interaction paradigms.

For example, the discussion of **non-command interfaces** (Jacob, 1993; Nielsen, 1993), which relates closely to the well-known paradigm of the **transparent interface**, suggests a shift from command-based interfaces to a non-command-based dialogue, in which, instead of the user issuing specific commands, the computer passively observes the user and provides appropriate responses. In this case, users could interact naturally, efficiently, and more directly with the task itself rather than with the mediating interface, thus making the interface transparent. This implies that the system should be able to work out the user's focus of attention.

Proactive applications share the goal of more natural and efficient interaction with the task itself. One of the vital aims of proactive applications is to "get the users out of the loop" (Tennenhouse, 2000) – that is, to decrease their burden by acting on their behalf. These kinds of applications should not just identify but even anticipate the users' needs. Here, the sense of attention is even more essential. However, proactive applications can be frustrating and annoying. In proactive applications interruptions are typical, notification systems (McCrickard, Czerwinski & Bartram, 2003) being one example. Interruptions cause an abrupt redirection of attention to a task that is often irrelevant for the primary task being executed. They can cause forgetting; distortion of the knowledge related to the main task and, as a consequence, mistakes (see, e.g., Latorella 1996; Oulasvirta & Saariluoma, 2004); and overall annoyance and anxiety (Bailey, Konstan & Carlis, 2001). Moreover, task switching has been determined to cause measurably reduced performance, since it requires mental reorientation. It involves top-down processing of attention shift, also known as task-set reconfiguration

3.1 Attention in user interfaces

(Pashler et al., 2001). These observations have led researchers to search for proper moments for the interruptions, like certain points in a task's life cycle (Cutrell, Czerwinski & Horvitz, 2001). Task decomposition on different hierarchy levels (based on psychological studies on cognitive event perception) has been used to inform the system of advisable interruption points (Adamczyk & Bailey, 2004).

Making better use of attentional processes would help in designing proactive applications that are more useful. How do attentive user interfaces make use of the attentional processes?

Attentive user interfaces

Vertegaal (2002) defines an attentive user interface as follows:

An Attentive Interface is a user interface that dynamically prioritizes the information it presents to its users, such that information processing resources of both user and system are optimally distributed across a set of tasks. The interface does this on the basis of knowledge – consisting of a combination of measures and models – of the past, present and future state of the user's attention, given the availability of system resources.

In other words, attentive user interfaces monitor the user's behavior both by using models of the user's behavior and by using different sensing mechanisms to measure the behavior. On the basis of the information collected, the system predicts what is the most relevant information that should be presented to the user at each point in time. Maglio, Matlock, Campbell, Zhai & Smith (2000) compress the same idea into a list as follows:

Attentive User Interfaces

- (a) monitor user behaviour,*
- (b) model user goals and interest,*
- (c) anticipate user needs,*
- (d) provide users with information, and*
- (e) interact with users.*

We can use several different sensing mechanisms to collect information on the behavior of a user interacting with a system. These may include, for example, microphones listening to acoustic information, cameras enabling analysis of the user's gaze and body gestures, or even electronic sensors that record muscle and brain activity and can be used to monitor the user's actions during the performance of a task (Cheng & Vertegaal, 2004; Surakka, Illi & Isokoski, 2004). In the example of attention-aware interruption handling above, the task decomposition model of the task being performed is the source of the information guiding the proper

moment for the interruption. Hence, also these systems can be considered to be examples of attentive applications.

However, monitoring the user's gaze behavior is the only one of these approaches that is able to provide reliable information on the actual focus of the user's attention. Even though it reflects only the overt attention – meaning that the user may be engaged with some cognitive processes not related to the focus of visual attention – the correlation between focus of attention and focus of visual attention is acknowledged to be very strong. Even though attention may be shifted without redirection of the focus of visual attention, there is some evidence that saccadic eye movements always produce concurrent shift of attention (Groner & Groner, 1989), which makes the correlation even stronger.

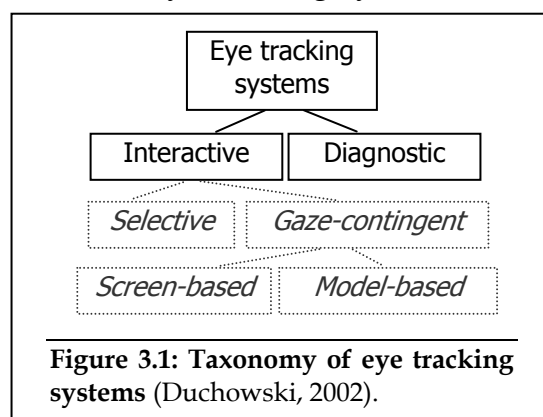
iDict uses the gaze information with the aim of enabling the reader to access the information at the exact time it is needed for performing the primary task (understanding the text being read). It provides the reader with the information without the need for task switching (to using a dictionary), thus minimizing the cost of interruption. The models of gaze behavior during reading are used for retracing the reading process and determining the best moment to provide the user with assistance.

Apart from a few experimental pilot systems (Bolt, 1980; 1981; 1985; Starker & Bolt, 1990; Jacob, 1991), such systems making use of gaze to get information on the user's attentional state have emerged only in the last few years. We next provide a review of gaze-based attentive applications.

3.2 GAZE-BASED ATTENTIVE SYSTEMS

When this work was undertaken (Hyrskykari et al., 2000), the above-mentioned piloting applications were the only systems making use of natural eye gaze behavior. The number of the applications that appeared so soon after that is forceful evidence of the emerging confidence that gaze will eventually reach a recognized role as an input channel. Duchowski (2002) provides a taxonomy in which he divides “eye tracking systems” into diagnostic and interactive systems (Figure 3.1).

By a diagnostic system he refers to the use of eye tracking as a research tool for studying visual and attentional processes. Typically, this means that an experiment presenting various stimuli for a subject is designed and performed. Eye movements during the experiment are recorded and post-analyzed off-line.



In interactive systems, gaze is used in real time, as an input modality. Duchowski (2002) divides interactive systems into selective and gaze-contingent systems. Selective systems are defined as those in which the point of gaze is used analogously to the mouse, as a pointing device. Gaze-contingent systems, on the other hand, exploit the user’s gaze for rendering complex displays, which Duchowski further divides between the screen-based and model-based according to the technique used to accomplish the rendering. At the time the taxonomy was presented, applications where gaze was used other than for pointing or rendering displays were scarce. Additionally, for many applications it is difficult to state that the point of gaze is used merely as a pointing device. While it may aid in pointing, it may at the same time be used in a more versatile way to address the focus of attention.

In order to help the reader to construct a conception of the design solutions used in the diverse set of applications, we categorize the interactive – which in our terminology are attentive – gaze-based systems according to their domain categories (Figure 3.2). Hyrskykari, Majaranta and Riih  (2005) presented an early version of this taxonomy.

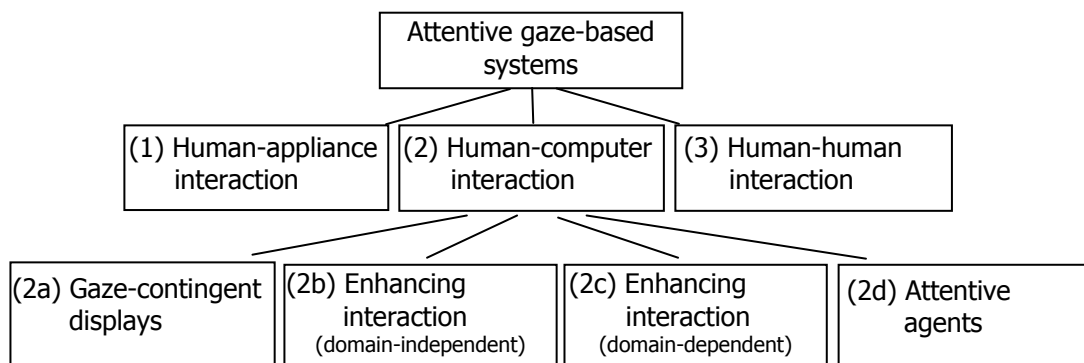


Figure 3.2: Taxonomy of attentive gaze-based systems.

The first level divides the systems to three categories: (1) human-appliance interaction, i.e., the systems interacting with a physical device that respond to gaze, (2) human-computer interaction, the ones making use of gaze in the interaction between human and a computer, and (3) human-human interaction, the systems that enhance the interaction between two or more people.

Table 3.1 brings together the gaze-based attentive systems and applications. All the systems that have been implemented are included¹,

¹ *Gaze-contingent displays (2a) and augmentative and alternative communication systems (AACS, in 2b) are exceptions on account of multiple systems in those categories (review articles for them are provided).*

although most are experimental implementations. In addition, there exist, of course, research reports and innovation papers contributing to gaze-based attentive systems. Below we provide a review of some of the systems in each domain category.

3.2.1 Interacting with an appliance

The systems in this category demonstrate that even without tracking the user's gaze direction, eyes can enhance the interaction substantially. Simply detecting the presence of eyes or recognizing eye contact with a target device gives us a variety of possibilities for establishing the desired interaction.

Selker, Lockerd and Martinez (2001) introduced **Eye-R**¹, a glasses-mounted, wireless device that is able to detect the user's eye motion and to

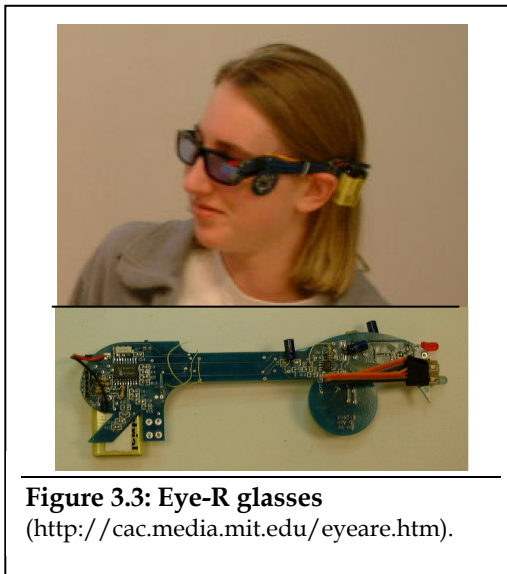


Figure 3.3: Eye-R glasses
(<http://cac.media.mit.edu/eyear.htm>).

store and transfer the information through the use of external IR devices. Eye-R consists of an infrared emitter and a detector that is positioned between the lens and the eye (Figure 3.3).

In principle, the emitter/detector unit can be mounted on any commonly used pair of eyeglasses. The transmitter (IR LED) illuminates the eye, and a photo diode (the detector) recognizes the light reflected from the surface of the eye.

Thus, even without a camera, Eye-R glasses are able to recognize the rough eye behavior of a user: whether the eye is open or closed, blinking, winking, staring, or gazing around. On the basis of the recognized behavior, the glasses are able to establish communication with a target in the environment. The target may be a PC gathering the information stored in Eye-R or another pair of Eye-R glasses detecting when two pairs of glasses are mutually aligned

¹ Later referred to also as *Eye-aRe*.

Domain category	System/application	References
1. Eye-sensing devices	Eye-R	Selker et al., 2001; Selker, 2004
	Eye-Bed	Selker, Burleson, Scott & Li, 2002; Lieberman & Selker, 2000
	EyeContact sensor	Morimoto et al., 2000; Shell et al., 2004
	EyePliances	Shell, Vertegaal & Skaburskis, 2003
	iLights	Kembel, 2003
2. Personal computing	Gaze-contingent displays (multiple)	Baudisch, DeCarlo, Duchowski & Geisler, 2003; Reingold, Loschky, McConkie & Stampe, 2003; Duchowski, Courtnia & Murphy, 2004 (reviews)
		Hutchinson, White, Martin, Reichert & Frey, 1989; Lankford, 2000; Majaranta & Riihä, 2002 (a review)
	AAC systems (multiple)	Bolt 1981; Bolt 1985; Jacob, 1991; Jacob, 1993; Fono & Vertegaal, 2005
		Jacob, 1991; Jacob, 1993
	Eye-controlled windows	Bolt, 1980; Bolt, 1985; Jacob, 1991; Jacob, 1993; Kaur et al., 2003
		Jacob, 1991; Jacob, 1993; Ohno, 1998
	Text scrolling	Zhai et al., 1999
	Moving and object	Bates & Istance, 2002; Ashmore, Duchowski & Shoemaker, 2005
	Menu selection	Jacob, 1993
	Magic Pointing	Takagi, 1997; Takagi, 1998
	Zooming interfaces	Sibert, Gokturk & Lavine, 2000; Khat, Matsumoto & Ogasawara, 2004a; Ohno, 2004
	Ship database	Ramloll, Trepagnier, Sebrechts & Finkelmeier, 2004
	Translation support	Eaddy, Blaskó, Babcock, Jason & Feiner, 2004
	Reading aid, browsing support	Qvarfordt & Zhai, 2005
	Therapeutic practice	Hyrskykari et al., 2000; Hyrskykari, 2003; Hyrskykari et al., 2003; Hyrskykari, 2006
EyeGuide	Hoanca & Mock, 2006	
iTourist	Vertegaal, Slagter, van der Veer & Nijholt, 2001	
iDict	Oh et al., 2002	
Secure passwords	Maglio, Barrett, Campbell & Selker, 2000; Maglio & Campbell, 2003	
FRED	Haritaoglu et al., 2001; Koons & Flickner, 2003; Selker, 2004 (Arroyo's dog)	
Look-to-talk	Wang, Chignell & Ishizuka, 2006	
SUITOR	Morimoto et al., 2000; Shell et al., 2004	
Attentive toys	Vertegaal, Dickie, et al., 2002; Dickie, Vertegaal, Fono, et al., 2004; Shell et al., 2004; Skaburskis, Vertegaal & Shell, 2004	
ESA	Vertegaal, 1999; Gemmell, Toyama, Zitnick, Kang & Seitz, 2000; Jerald & Daily, 2002; Vertegaal, Dickie, et al., 2002	
Attentive Cell Phones	Qvarfordt, 2004; Qvarfordt & Zhai, 2005; Qvarfordt, Beymer & Zhai, 2005	
Wearable EyeContact sensor and ECSSGlasses applications		
Videoconferencing		
RealTourist		
3. Group communication		
2d. Attentive agents		

Table 3.1: Gaze-based attentive systems and applications.

Eye-R glasses presented the idea of sending the surrounding objects the information that the user is paying attention to them. However, since they cannot recognize the direction of gaze, deducing the target object when several candidate objects are present is prone to error. As noted in the introduction, measuring the direction of gaze when the user is allowed to move freely is complicated. Selker's group used a simple set of natural eye



Figure 3.4 Eye-bed (Selker, Burleson, Scott & Li, 2002).

behavior gestures also for implementing **Eye-Bed** (Lieberman & Selker, 2000; Selker et al., 2002), an application for controlling a multimedia scene projected on the ceiling above a bed (Figure 3.4). An eye tracker was placed on a "lamp arm" over the head of the person in the bed. Cursor control was tried out with different pointing devices; thus, the eye tracker was not used to control selection of objects in the projected image. Instead, natural behaviors of the eyes such as closing and opening, gazing around, staring at one place, and nervous blinking were used to adapt the presented images to the observed state of the user's attention.

In many cases, giving voice commands to digital household (or office) devices would be a natural way of interacting with the surrounding technology. Addressing the intended device has been recognized as one of the essential communication challenges for future human-computer interaction (Bellotti et al., 2002). It has also been shown that a subject tends to establish natural eye contact with the object to which he or she is going to address the speech (Maglio, Matlock, et al., 2000).

The concept behind several experimental applications developed at Queen's University is that, instead of making the eye tracker wearable for a freely moving user, remote eye trackers, **EyeContact sensors** (Figure 3.5), are housed in the devices to make them eye sensitive. The technique used to implement EyeContact relies on two main design inspirations (Vertegaal, Dickie, et al., 2002).

First, it utilizes the ideas of two sets of on- and off-axis (aligned at the same vs. different axis with the camera) LEDs sending timely synchronized infrared light beams into the eye to produce both bright and dark pupil effects (Morimoto et al., 2000). That facilitates a robust detection of eyes from a large scale camera view. The other idea is the insight that the common tracking of the corneal reflection point can be simplified by detecting only the eye contact with the camera, disregarding the other positions of the eye. That is, when the corneal reflection point is located near the pupil centre, the eyes are looking straight at the camera.

3.2 Gaze-based attentive systems

When an EyeContact sensor is placed in a digital household appliance, the system provides the device with information on when a user is attending to it. Thus, it removes the need for using indirect referencing via naming and allows the user to address the commands directly to the device. Also, the limited vocabulary of available commands for the addressed device helps the system to sort out ambiguities and errors in speech recognition.

Examples of such **EyePliances** given by the developers include eye-sensitive lights (Figure 3.6), attentive television, and video players. In addition to the information that the user is attending to a device, also the lack of attention can be used as a valuable information source. An example is a video player that pauses when the user turns away from it to answer a phone call (Shell et al., 2003).

Gaze-sensitive toys are another example of human-appliance interaction. Haritaoglu et al. (2001) point out that machines would be more powerful if they had even a small fraction of humans' ability to perceive, integrate, and interpret visual and auditory information. Bearing this in mind, they implemented the robot VTOY (a later version of which was referred to as PONG; Koons & Flickner, 2003), which is capable of deciding when to start engaging with a human partner and of maintaining eye contact with the partner. In addition to gaze tracking, the robot also tracked the user's facial expressions and responded by mimicking them (Figure 3.7).

A similar attempt to sense human attention was experimented with in the development of Ernesto Arroyo's dog, which barks when attended to (Selker, 2004).

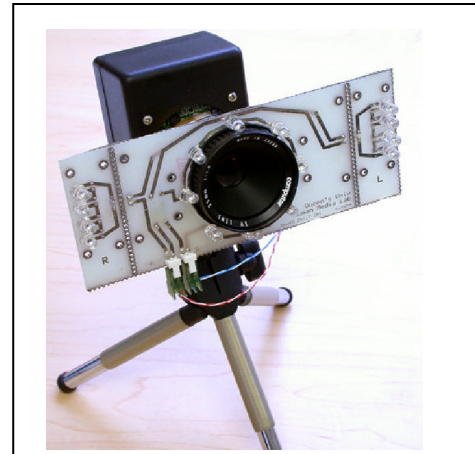


Figure 3.5: An EyeContact sensor (Shell et al., 2003).

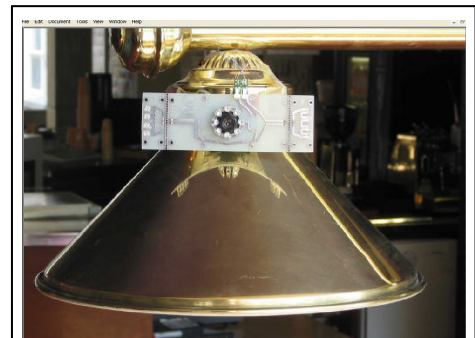


Figure 3.6: Eye-sensitive lights (Shell et al., 2003).

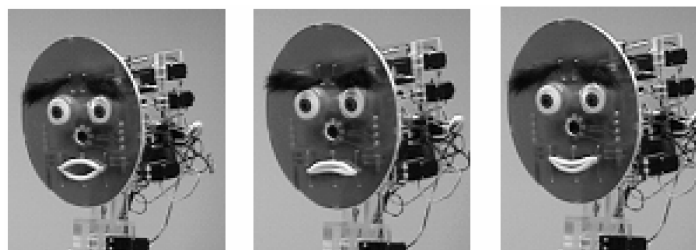


Figure 3.7: VTOY (Haritaoglu et al., 2001).

3.2.2 Interacting with a computer

We divide the personal computing applications further into four categories:

- gaze-contingent displays (present also in Duchowski's taxonomy),
- applications accelerating domain-independent interaction at the operating system level,
- applications enhancing domain-specific interaction, and more generic
- attentive agents.

We will next have a look at systems in each of these categories.

(a) Gaze-contingent displays

In a broad sense, we can use the term "gaze-contingent applications" to refer to all applications where the focus of the user's visual attention is used in real time to alter the on-screen view. However, the term is most frequently used to denote displays adapting their resolution to match the user's gaze position; high-resolution information is rendered at the user's gaze position, and the resolution is degraded in other areas. These are called **gaze-contingent (multiresolutional) displays**. The main motivation for decreasing the resolution in peripheral image regions is to minimize overall display bandwidth requirements. Several reviews of these systems have recently been published (e.g., Baudisch et al., 2003; Reingold et al., 2003; Duchowski et al., 2004).

(b) Accelerating interaction (domain-independent)

As noted in the introduction, using the point of gaze as a straightforward substitute for a mouse is difficult due to the inaccuracy involved and the Midas touch problem. In AAC systems, eye mice may sometimes be the only option allowing a disabled user to communicate and to manage devices in the environment. Eye mice are used in either standard or eye-mouse-tuned GUI applications, such as in writing using virtual keyboards (for a review, see Majaranta and R  ih  , 2002). Even though there are several commercial gaze-based AAC systems¹, not many research papers on their design and implementation have been written (with the exception of the papers on the ERICA system (Hutchinson et al., 1989; Lankford, 2000). One conceivable solution for the inaccuracy problem is to **zoom** the interface gadgets to be large enough for gaze selection. A straightforward zooming of the elements at the point of gaze does not necessarily work very well (Bates & Istance, 2002), but some

¹ For a list of them, see the COGAIN Web pages at <http://www.cogain.org/eyetrackers/>.

3.2 Gaze-based attentive systems

experimental systems demonstrate that with special solutions zooming is a viable solution for inaccuracy problems at least in command-based systems (e.g., Ohno, 1998; Lankford, 2000; Pomplun, Ivanovic, Reingold & Shen, 2001; Špakov & Miniotas, 2004; Ashmore et al., 2005).

Taking into better account the attentional property of the gaze, a well-designed approach to eye input has potential for providing more natural and effective interaction not only in AAC systems but also in general windowing systems. For example, as we have noted, people tend to look at the object they wish to interact with (Maglio, Matlock, et al., 2000); in pointing tasks, the eyes always move to the target first, and the cursor then follows. As already introduced in Chapter 1, Magic pointing (Zhai et al., 1999) is a system combining the strengths of two input modalities: the speed of the eye and the accuracy of the hand. The gaze location only indicates a dynamic “home” position for the cursor. Thus, when the user is about to point and select a target, the cursor is already “automatically” in the vicinity of the target. Using a mouse as the “clutch” for the selection also avoids the Midas touch problem.

Gaze can also help in managing multiple task windows on the desktop. The idea of selecting the active workspace by gaze was presented by Bolt (“Gaze Orchestrated Windows” in Bolt, 1981; Bolt, 1985) and by Jacob (“Listener windows” in Jacob, 1991). Tests performed with **EyeWindows** (Figure 3.8) prove that developments in eye tracking technology make the idea now viable for real-world use (Fono & Vertegaal, 2005). Controlling task windows seems to be especially suitable for gaze: windows are large enough objects to diminish the inaccuracy problem and the technique frees the hands for managing the window contents (often involving text input via keyboard).

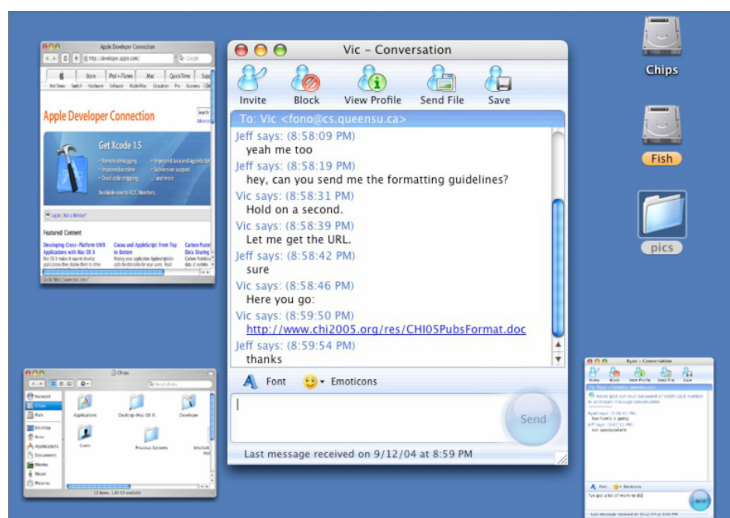


Figure 3.8: EyeWindows. The active task window is zoomed to a usable resolution, while inactive windows are zoomed out of the way (Fono & Vertegaal, 2005).

Using the eyes to indicate the focus window but letting the user perform the actual selection with a key press proved to work better than attempts to activate the selection merely by gaze (Fono & Vertegaal, 2005).

(c) *Enhancing interaction (domain-dependent)*

In designing interaction for a specific application, gaze input can provide invaluable information enabling adaptation of the behavior of the application to the user's behavior. In some cases, the information may be simply the user's focus of attention in the application. In other cases, the information may be more than just the instantaneous focus; it can be an interpretation of gaze behavior in terms of the contents of the application window over a longer period of time.

The Little Prince application, mentioned already in the introduction (an application in which the user's gaze path was used to drive the narration of the story) was an early example (Starker & Bolt, 1990). **iTourist** (Qvarfordt & Zhai, 2005) applies the same idea of an eye-guided narrator. It provides tourist information about an imaginary city, Malexander, by following the user's interest. It shows a map and photos of different places, providing prerecorded verbal information about them (Figure 3.9).

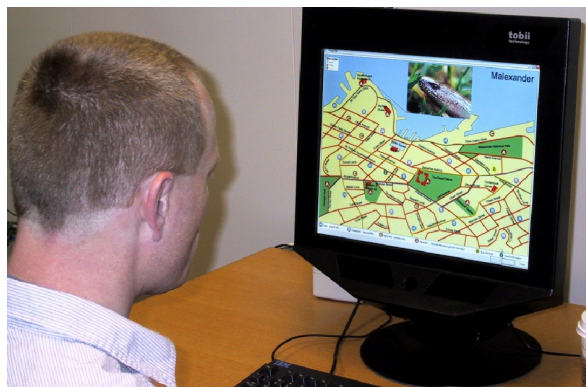


Figure 3.9: **iTourist** (Qvarfordt & Zhai, 2005).

The output is adapted on the basis of assumptions as to the user's interests, based on the user's eye-gaze patterns. User studies showed that, regardless of occasional mistakes, most of the time **iTourist** was able to tell the users about places they really were interested in.

Another map-related application, **EyeGuide** (Eaddy et al., 2004), takes a more guiding approach. It assists a traveler who is looking at a subway map. **EyeGuide** detects when the user appears to be lost. Based on the information on the user's point of gaze, it provides spoken hints to help the user to navigate the map (e.g., "Look to the far right"). The system preserves the user's privacy by whispering the instructions via an earpiece. By combining the information on what the user's goal is and what the user is currently looking at, **EyeGuide** can provide contextual

information (e.g., “Exit here for JFK airport”).

It is predictable that educational applications would benefit from information on what the user is really paying attention to. Ramloll et al. (2004) exploited gaze in this field. They monitored the eye movements of **autistic children** in an aim to reinforce appropriate gaze behavior in the children.

We categorize iDict as an example from this category (enhancing domain-dependent interaction); in our case, the domain is understanding the text (written in a foreign language) being read. Before we started our work, there was already a related application, which in part inspired our work¹. The eye-movement-enhanced **Translation Support System** (Takagi, 1997; 1998) was designed to assist in the task of translating text from Japanese into English. The system analyzes eye movements during the translation, detects patterns in eye movements, and responds appropriately. For example, when the user scans through a Japanese-to-English translation corpus, the system automatically removes the already scanned material and continuously retrieves new information. If the user pauses due to hesitation, the system finds keywords in the sentence under focus and retrieves new corpus results. Contemporaneously with our iDict design paper (Hyrskykari et al., 2000), Sibert et al. (2000) introduced their **Reading Assistant**, which shares some goals with iDict. Reading Assistant uses gaze to trigger auditory prompting for remedial reading instruction (when the user is reading text written in the native language). The application follows the user’s gaze path and highlights the words of the text as the reading proceeds from word to word. As soon as the program notices hesitation, it speaks out the word. It provides unobtrusive assistance to help the user with recognition and pronunciation of words. Like iDict, the Reading Assistant application exploits knowledge of how the gaze usually behaves during reading.

In a recent report on ongoing work, Khiat et al. (2004a) suggest using hidden Markov models and a Bayesian network for automatic detection of comprehension difficulties in reading. In their subsequent work on a **gaze-sensitive dictionary** (2004b), they used regressions² to detect occasions when the reader has problems understanding the text being read.

¹ Another clear motivating application for iDict was the ship database application (Jacob, 1993).

² The occasions when the reader’s point of gaze moves backward, to review part of a text already passed.

(d) Attentive agents

Mutual gaze is an important cue in human-to-human conversation. Interface agents, especially embodied agents, would also greatly benefit from the user's gaze direction cues. The multi-agent conversational system **FRED** (Vertegaal et al., 2001) is an application in which the artificial agents are aware of the users' eye gaze direction. By combining information from gaze and speech data, each agent is able to determine when it is spoken to, or when it should listen to the user. Similarly, **Look-to-Talk** (Oh et al., 2002) uses information on gaze direction to help in deciding when to activate automatic speech recognition. An artificial agent (Sam) knows that he is being spoken to when the human participant looks at him. In the experiment, the users preferred the perceptual look-to-talk interface over a more conventional push-to-talk interface where they had to push a button to indicate that they were talking to the agent. Also the "empathic tutoring software agent," **ESA** (Wang et al., 2006), uses real-time eye tracking to personalize its behavior, this time especially in a tutoring environment.

In the attentive information system **SUITOR** (a "simple user interest tracker," Maglio & Campbell, 2003), agents track the user's attention via multiple channels: keyboard input, mouse movements, Web browsing, and gaze behavior. Information on the gaze direction is used to determine where on the screen the user is reading. The **SUITOR** system uses the information to determine what the user might be interested in, and it automatically finds and displays the potentially relevant information. Suggestions are displayed in a timely but unobtrusive manner in a scrolling display at the bottom of the screen.

3.2.3 Interacting with other humans

In the systems above, gaze information is used to enhance human-computer interaction. We now turn to systems where human-to-human communication (communication between two or more people) is enriched with attention-sensitive devices. Videoconferencing systems are such a domain area, but Eye-R and EyeContact sensors, which were used to implement EyePliances, have also been used to augment conversations between two people.

Eye-R glasses (review Figure 3.3) were tried out in an experiment imitating a party setting. They were used to send a "business card" when a person stands talking to another person (both using the glasses) at a party. Later, when the wearer steps in front of a base station, the information gathered during the evening is brought up on display (Selker et al., 2001; Selker, 2004). However, without the sense of gaze direction (which is missing from Eye-R glasses), reliable identification of the interlocutor appears to be a problem.

3.2 Gaze-based attentive systems

When placed in the close proximity of the user's eyes, the EyeContact sensor system is able to detect whether the user is in eye contact with another person (Figure 3.10).

The concept was exploited to design an **Attentive Cell Phone** scenario. If the user is engaged in a conversation, this information may be passed to the caller, or the phone can switch the normal ringing sound for the incoming call to a less obtrusive notification mode.

The intensity of attention to a conversation can be deduced from the speech activity via microphones, but since conversation is a reciprocal action, silence does not necessarily imply that the user is not socially committed. The designers considered that sensing the ongoing eye contact gives valuable additional information on the user's state of attention (Vertegaal, Dickie, et al., 2002).

ECSGlasses (eye-contact-sensing glasses, in figures 3.11 and 3.12) are a more sophisticated version of the wearable EyeContact sensor. The camera and the off- and on-axis IR LEDs are here embedded in a pair of glasses. The on-axis illuminators producing the bright light effect are positioned around the camera on the bridge of the nose, and the off-axis illuminators reside near the temples of the glasses, producing the bright pupil effect. Also, a microphone is embedded in one arm of the glasses (Dickie, Vertegaal, Shell, et al., 2004). ECSGlasses were exploited in the implementation of a revised version of the Attentive Cell Phone (Shell et al., 2004), EyeBlog, the attentive hit counter, and the Attentive Messaging Service.



Figure 3.10: A wearable EyeContact sensor (Vertegaal, Dickie, et al., 2002).



Figure 3.11: ECSGlasses (Dickie, Vertegaal, Shell, et al., 2004).



Figure 3.12: ECSGlasses in action (Shell et al., 2004).

EyeBlog (Dickie, Vertegaal, Fono, et al., 2004) is an eye-contact-aware video recording and publishing system that is able to automatically record face-to-face conversations. The **attentive hit counter** (Shell et al., 2004) measures the number of times somebody makes eye contact with the user. Without person identification functionality, the system counts all eye contacts. Since ECSGlasses record a video of the scene experienced by the user, identification of the interlocutor is possible; the research group intends to add person identification to the system. The **Attentive Messaging Service** (AMS) (Shell et al., 2004) can communicate the availability or absence of “buddies” on the user’s buddy list who are facing toward or away from the user. Rather than making use of indirect inferences from, e.g., keyboard or mouse activity as conventional messaging clients do, AMS has the knowledge of the user’s eye contact with the computer screen.

In videoconferencing, one problem is that eye contact between people attending the session is lost. Only if someone looks directly at the camera does the image of the person on the screen seem to look at the viewers. Gemmell et al. (2000), as well as Jerald and Daily (2002), manipulated the real-time video image by rendering a modified image of the eyes upon the original video image. The idea was that, after the manipulation, the eyes seemed to look in the correct direction, creating an illusion of eye contact. A real video stream is considered better than, e.g., an animated avatar because the real video transmits facial expressions and eye blinks as they appear.

GAZE (Vertegaal, 1999) and **GAZE-2** (Vertegaal, Weevers & Sohn, 2002; Vertegaal, Weevers, Sohn & Cheung, 2003) are attentive videoconferencing systems that convey eye contact on the part of the participants in the conference. The users meet in a virtual 3D meeting room, where each member’s image (as an avatar) is displayed in a separate video panel. The direction of each user’s gaze is tracked, and each user’s image is then rotated toward the person he or she is looking at (Figure 3.13).



Figure 3.13: GAZE (Vertegaal, 1999).

In addition, a light spot is projected onto the surface of the shared table to indicate what (e.g., which document) the user is looking at. The light spot helps to resolve references to particular objects (e.g., “look at this”). GAZE showed animated snapshots of the participants, but GAZE-2 uses live video (Figure 3.14). In addition, GAZE-2 uses the information on the participants’ gaze direction to optimize the bandwidth of streaming media. For example, in Figure 3.14, the image of the person on the left is broadcast at higher resolution, since everyone is currently looking at him. Also the images of the two other participants are rotated toward him, to convey their gaze direction.

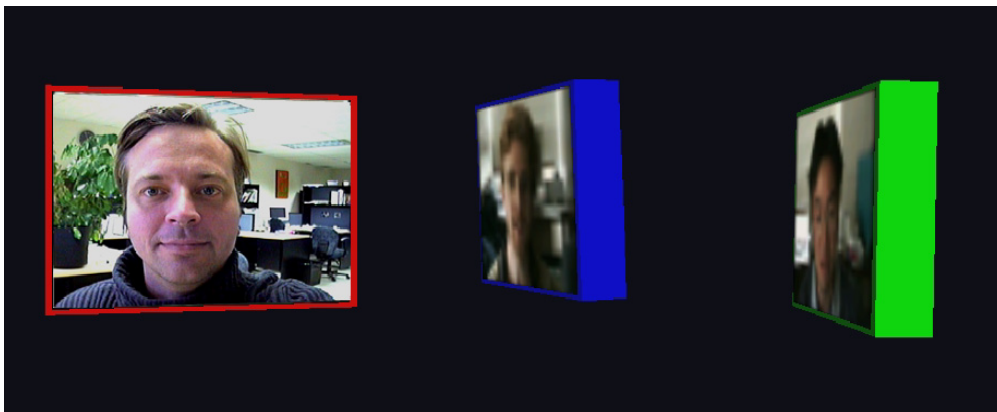


Figure 3.14: GAZE-2 (Vertegaal, Weevers & Sohn, 2002).

As the last example of using gaze to enhance mediated human-to-human interfaces, we briefly consider the **RealTourist** application (Qvarfordt, 2004), a variation of the iTourist application introduced above. RealTourist communicates the information on the place of the user’s visual attention on the computer screen to a tourist consultant, who assists the tourist remotely. Both the tourist and the consultant see the same map on their screen. In addition, the consultant sees the tourist’s gaze position superimposed on his or her screen. An experiment showed that gaze information helped in resolving references to objects and in determining how interested the tourist was in them. Gaze provided cues about when it was suitable to switch topics. Information on the visual attention helped in interpreting unclear statements and in establishing a common ground: the consultant was able to assess whether the tourist had understood instructions and to make sure both people were talking about the same object. Conveying the real-time visual attention information of a customer remotely for the consultant opens up interesting possibilities for remote consulting in general.

The gaze-based attentive systems presented above show the rich diversity of ways in which eye and gaze awareness can be exploited in human-device, human-computer, or computer-mediated human-human interaction. iDict aims to support the user in a special case of human-computer interaction: in the process of reading documents written in a

foreign language. In iDict it is essential to trace the reading process; that is why we next – before considering the details of the application – review the main findings of the research on eye movements in reading.



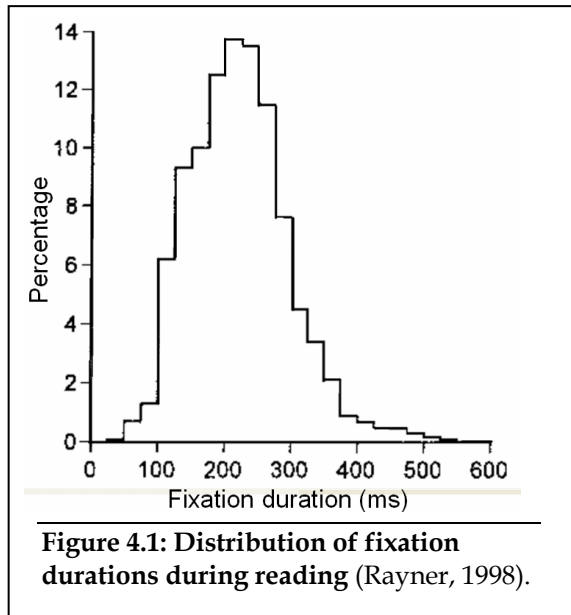
4 Attention and Reading

For iDict, it is essential to track the process of reading and to be able to identify the situations where the eye behavior differs from the norm, thus indicating that the reader has problems in understanding the text being read. In this chapter, we will give a brief review of salient studies of reading. We will introduce the general research results showing how the eyes move, then discuss how the attention is oriented during reading. The measurements used in monitoring the eyes' behavior are then presented.

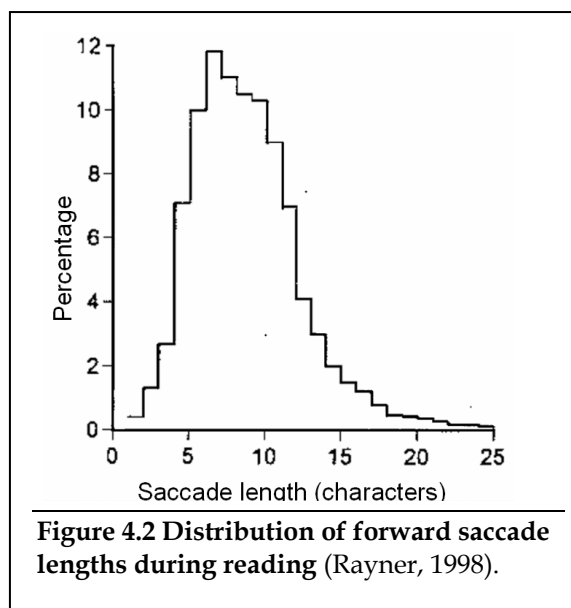
4.1 EYE MOVEMENTS AND READING

To simplify the discourse, researchers frequently divide the visual field into three regions, even though – as was noted in Subsection 2.1.1 – there is no clear biological basis for such a division. The regions used in discussion are called the **foveal**, **parafoveal**, and **peripheral** regions. Readers move the word being read into the foveal region ($< 2^\circ$) to be able to recognize the word. They are able to extract some visual information on the word also from the parafoveal region (2° – 5°). In the periphery (the region beyond the parafovea), the vision is poor and does not convey information on the text to the reader.

Readers make four to five fixations per second. Reported average fixation durations during reading vary from 225 ms (e.g., Rayner, 1998) to 250 ms (e.g., Vitu, McConkie & Zola, 1998). Typically, a reader's fixation durations vary from 100 to 500 ms, but occasionally fixations as short as 50 ms are recorded (Figure 4.1). A common pattern of eye behavior during reading is that the focus of gaze moves forward, from a word to the next word. However, when reading text compatible with their skills, readers often skip some words, and also make **regressive saccades**; that is, they



(when reading Finnish, in which the average word length is longer than in English), the average saccade length is longer, 11 character positions (Hyönä, 1995). The measurements are given in character positions because the size of the font and the reader's distance from the text has been found



to have only marginal effect on saccade lengths. About 10–15% of saccades are regressions (Rayner, 1998). Often the length of a regressive saccade is only a few characters; these cases are usually interpreted to be corrections of overshoot saccades. Longer regressions, exceeding 10 characters, are considered to reflect difficulties in understanding the text (Rayner, 1998). It has been found that readers are able to very accurately target a regressive saccade to the position in the text that caused the problem (Frazier & Rayner, 1982).

The figures describing the reading process presented above are average figures, and there is considerable variability in the measurements. This consists of not only between-reader variability but also within-reader variation in fixation durations, word skipping, saccade lengths, and regressive saccades. A massive amount of research has been performed in order to find out where the variation derives from. What are the inferences we can make concerning the attentional and linguistic processes on the basis of this variation?

4.2 READING AS AN ATTENTIONAL PROCESS

Early reading researchers assumed that cognitive processes cannot affect eye movements during reading, because the oculomotoric processes are so fast. Saccade lengths were assumed to be more or less constant, controlled autonomously by the oculomotor system and changing only as a function of the overall difficulty of the text (Brysbaert & Vitu, 1998). However, as a result of improved technology and research techniques, the last couple of decades have brought much evidence that cognitive processes do affect eye movements. The controversial question in the last two decades has been which of the systems, oculomotor or cognitive, dominates the eye movements during reading in different phases of the process.

Just and Carpenter (1980) presented two basic hypotheses concerning the relationship between eye movements and cognitive processes in reading: (1) **the immediacy assumption** and (2) **the eye-mind assumption**. According to the immediacy assumption, a word is interpreted on several levels when it is fixated, even though this sometimes leads to misinterpretations. The word is identified, it is assigned a meaning (chosen from possible candidates), and its semantics are resolved in the context of the sentence. According to the eye-mind assumption, the word stays fixated as long as it is being processed. This would mean that fixation durations should provide a direct estimate of the time used to process each word of text and hence a reliable metric of the cognitive processes involved in reading.

The description above of word skipping seems to be in contradiction with these assumptions. Processing of words occurs even when they are not fixated. For example, in 1985, Fisher and Shebilske (as cited by Rayner, 1998) conducted an experiment in which they recorded eye movements of readers reading a text. Then the text was modified by removing the words the readers did not fixate on at all. A second group had difficulties in understanding the modified text. Other researchers have proven that readers do acquire some information on the words they do not fixate upon. In 1979, McConkie presented an assumption that word skipping depends on whether the parafoveal word has been identified already, during the previous fixation (Brysbaert & Vitu, 1998). Accordingly, skipping of a word depends on the length of the **perceptual span** field, the region within which a reader is able to obtain information on each fixation. How large is the perceptual span?

4.2.1 Perceptual span field

In reading, the perceptual span field is asymmetric (Rayner, 1995). The span extends 14–15 character spaces to the right of fixation, but on the left only to the beginning of the fixated word, or 3–4 character spaces. The perceptual span, however, depends on how the text is oriented: for Kanji readers the parafoveal vision is more biased downwards (Osaka, 1993),

and Hebrew readers are able to use their parafoveal vision more efficiently to the left (Pollatsek, Bolozky, Well & Rayner, 1981). Moreover, bilingual readers fluent in English and in Hebrew are able to change the perceptual span according to the language they are reading (Pollatsek et al., 1981).

The **word identification span**, the region within which a reader is able to recognize the words, is shorter, about 7–8 character spaces to the right of fixation (Rayner, 1995). However, the word identification span field is not stable, but it varies from fixation to fixation depending on the “difficulty” of both the parafoveal information and the word fixated upon (including, e.g., the word frequency (Inhoff, 1984) and the complexity of syntactically parsing the word in the sentence). The reader may, for example, identify three short words parafoveally, even though they make up more than eight characters (Henderson & Ferreira, 1990). Also, one’s skills in reading may affect the size of the word identification span, though observations to this effect have been made only with substantially divergent readers (children just learning to read or readers suffering from dyslexia). Underwood and Zola found no differences in word identification span size between “good” and “poor” fifth-grade readers (Rayner, 1995).

The processing of a word peripherally is assumed to be used for two purposes. First, it is used to program the landing position of the forthcoming saccade to the next word. Efficiency of foveal word recognition is greater if the fixation entering a word lands at the **optimal viewing position** of the word – that is, near the center, or slightly left of the center, of the word. If the landing position differs from the optimal viewing position, the probability of refixations on the word increases (O’Regan, 1981; 1990). Second, the **preview benefit** attained by processing the “next” word peripherally is essential to the attentional theory of reading.

4.2.2 Attentional theory of reading

Morrison (1984) established a theory of reading and its interwoven collaboration of attentional and oculomotoric processes. Attention is originally focused on the fixated word. When the **lexical access**¹ to the word is complete, attention moves to the next, parafoveal, word and processing of the parafoveal word begins. Shift of attention launches the programming of the forthcoming saccade. Because saccades are ballistic operations, launching a saccade requires time to prepare the operation. This delay preceding a saccade is called **saccade latency** or **saccade programming**. The saccade programming takes 150–175 ms (Abrams & Jonides, 1988). If the parafoveal word is identified during this time, the

¹ *Lexical access refers to “the process of identifying a word’s orthographic and/or phonological pattern so that semantic information can be retrieved” (Reichle et al., 1998).*

word is skipped: attention is shifted to the next word and saccade programming is reinitiated. The speed of identification of the parafoveal word depends on the characteristics of the word, word length being the most influential. For short (two- or three-character) words, skipping is very common, but it is rare for longer (six-to-10-character) words. Morrison's model has been supplemented and refined by several researchers (e.g., Henderson & Ferreira, 1990; Vitu & O'Regan, 1995; Reichle et al., 1998).

The theory attenuates the immediacy and eye-mind assumptions: some part of the time during which a word is fixated is used to process the parafoveal word. In addition, some researchers have found evidence that if a word causes the reader difficulty, its processing may be continued even after a saccade to the next word has been performed (the so-called **spillover effect** – see, e.g., Balota, Pollatsek & Rayner, 1985; Rayner & Duffy, 1986; Rayner, Sereno, Morris, Schmauder & Clifton, 1989).

However, even in the attentional model, the time when the saccade programming for the next launch site is started is dependent on the lexical access of the word. This means that the time for which a word is fixated depends on lexical characteristics of the word – for example, on the frequency and predictability of the word.

4.2.3 Measurement of reading behavior

In iDict, we are interested in the time used for processing a word and whether this time is within the limits typical of normal reading behavior. If not, difficulties in comprehension may be indicated. In assessing the time used for processing a word, it is not possible to confirm neither preview nor spillover time through eye movements. But, as noted above, even without information on these, a prolonged duration of fixations on the word in focus indicates problems in identifying the word. Beyond the process of identifying a word during reading, also higher-level linguistic processes, such as syntactically complex sentences, or ambiguous semantics (the so-called garden path effect – see, e.g., Clifton, Bock & Radó, 2000) of the sentence, may cause prolonged fixations and regressions. However, associating these kinds of problems with the right point in the text on the basis of eye movement is difficult. Also, their effects on gaze paths may be very individual (Reichle et al., 1998).

The most commonly used metrics for processing difficulties during reading are (1) first fixation duration, (2) gaze duration, and (3) total reading time for a word or a critical region of interest (Rayner et al., 1989; Liversedge, Paterson & Pickering, 1998; Rayner, 1998).

First fixation duration is the duration of the first fixation when the word is entered for the first time (first-pass reading). Gaze duration represents the sum of all fixations made on a word during the first-pass reading prior to

movement to another word. According to Inhoff (1984), these two measurements address different processes: the first fixation duration is associated with lexical access of the word, whereas the gaze duration reflects also the text integration process¹. Unlike these two metrics, total reading time for a word takes into account also regressive fixations on the word.

In addition to these three measurements, we have already noted above that regressions – especially long inter-word regressions – and comprehension difficulties are interrelated. When considering one of the refined models of Morrison’s attentional reading model, E-Z Reader (Reichle et al., 1998), we note that the programming of the next saccade position is calculated at the point where a familiarity check of a word has been performed (the familiarity check is the first part of the lexical access process). Refixations on a word are often explained by an unfavorable landing position on a word, but on the basis of E-Z Reader we can assume that a word is refixated upon due to an unsuccessful familiarity check. Accordingly, we can add to the above list of measures possibly indicating comprehension difficulties (4) the number of fixations on a word and (5) the regressions.

4.3 SUMMARY OF PART I

In this part of the dissertation, we introduced the background for designing our gaze-aware reading aid, the iDict application.

We first introduced gaze tracking. The biological background explaining how gaze can be tracked on the basis of eye movements and the technical solutions for doing so were presented. More importantly, we focused on explaining the limitations of gaze tracking. In tracking natural gaze behavior, no matter the evidently forthcoming technical improvements, inaccuracy will always be a factor in gaze tracking. After providing this background, we surveyed the psychological research concerning human attentional processes and discussed their implications for human-computer interface and interaction design.

Attention was observed to be an underused potential resource for enriching human-computer interaction, and gaze was noted to be the only source we can use to obtain reliable information on the user’s actual focus of attention in real time. Even though using gaze information as input is a relatively new idea, such systems (at least at the experimental level) are rapidly emerging. They were reviewed in a framework created on the

¹ This has been argued from the position that if the cognitive processing of a word is very fast, it may affect also the first fixation duration (Rayner, 1998).

4. 2 Reading as an attentional process

basis of the application domains for which they were designed.

Finally, research into eye behavior in reading was reviewed, with a special focus on how attention relates to the eye movements that occur in reading. Possible metrics for judging difficulties in comprehending the text were extracted.

In the next part of the thesis, we focus on describing iDict, from both the user's (reader's) and the implementer's point of view.



Part II

The iDict Application

Chapter 5 iDict Functionality

Chapter 6 iDict Implementation





5 iDict Functionality

iDict was designed as a test-bed application to experiment with the possibilities for using gaze input to adapt the application's behavior to the user's behavior. There were three main reasons for choosing a reading-aid application for this purpose:

1. Reading is a task performed regularly in most interfaces. Therefore, the generalized results of the example application are potentially valuable in a wide range of applications.
2. Eye movement behavior in reading is a thoroughly studied field. Hence, when interpreting eye behavior during reading, we can make use of a large amount of background knowledge produced by the psychological research into reading .
3. The application is an example of a more general idea: the point of gaze can be used as a reference for the user's focus of attention. If the user's behavior cues some desired action, the target of the action can potentially be deduced from the gaze path.

In this chapter, we will describe the iDict application from the user's perspective: how is iDict used, and how does it provide help for the user? The next chapter (Chapter 6) gives an overview of the implementation of iDict.

5.1 ON IDICT'S DESIGN RATIONALE

The aim of the application is to help the user in reading on-screen documents written in a foreign language by giving the user **the right kind of help at the right time**.

In preliminary tests we observed Finnish readers while they read English text documents, and interviewed them afterwards. We found that there were two main behavioral patterns the readers adopted when they encountered a problematic word. If the word seemed to be essential for understanding the text, they could stop the reading and consult either a printed or an electronic dictionary. However, since this interrupts the normal flow of reading and requires an effort to recapture the text context afterwards, some of the readers chose another behavior pattern. They did not check the translation of the problematic word at all. They hoped that the context might eventually reveal the meaning of the problematic word or that the word would turn out to be inessential for understanding the text. Naturally, this could lead to faulty comprehension – whether through incorrect interpretation or simply incomplete understanding – of the meaning of the problematic sentence. Some readers returned to the problematic word much later, when reading subsequent sentences.

Some of the readers said that when reading text they do not like to be interrupted. According to them, it is important that the atmosphere, “the world of the text,” not be disturbed.

These observations led us to conclude that the reading aid should on the one hand **be as automatic as possible** and, on the other, **disturb the reading process as little as possible**. These two goals were adopted as the leading principles for the design of iDict.

The next three sections introduce the reader to the use of the application. First, the features of the user interface in the normal context of use are presented (5.2). Then we describe how the application can be personalized according to the reader's preferences (5.3) and how the reader can specify the language resources iDict uses to give the assistance (5.4).

5.2 USER INTERFACE – IDICT FROM THE USER'S PERSPECTIVE

An ideal use scenario for iDict is simple: the user opens a document for reading, turns the eye tracking on, starts reading the text, and automatically gets help from the system when having trouble understanding the text. Using iDict is described in more detail below.

A general view of iDict (Figure 5.1) shows that the main window of iDict is split into two frames: the **document frame** and **dictionary frame**. The

splitting can be performed either vertically (as done in the picture) or horizontally, in which case the dictionary frame is aligned at the bottom of the main window. Early test readers consistently preferred vertically aligned frames, so subsequent tests were performed with the vertical layout.

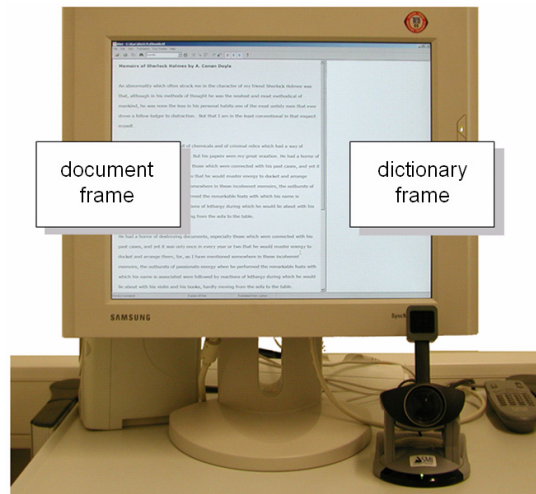


Figure 5.1: iDict, a general view of the application.

5.2.1 Starting iDict

The document is opened for reading in iDict just as in normal GUI applications via the File | open option, which displays the document in the document frame. iDict can be ported to support different eye trackers. The tracker in use is specified in a dialog opened via the Eye Tracker menu or by pressing the Eye Tracker Settings shortcut button in the toolbar (label 4 in Figure 5.2). Eye input is turned on by using the Eye Tracker menu or just by pressing the Eye Tracking on/off shortcut button (label 3 in Figure 5.2).

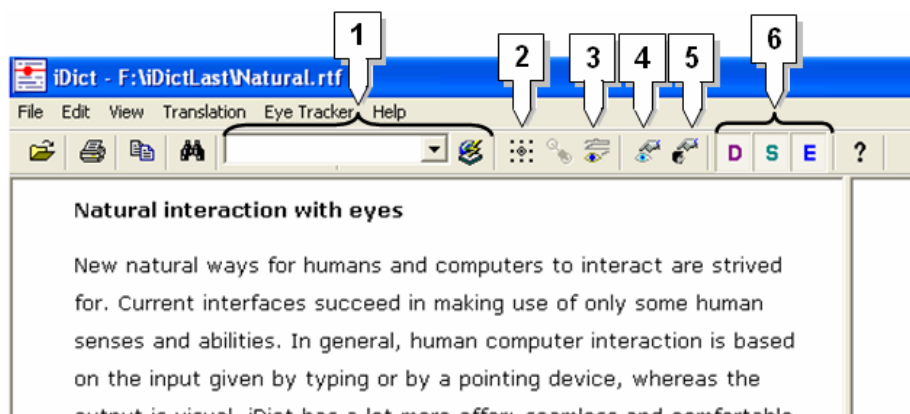


Figure 5.2: Toolbar shortcut buttons. Shortcuts are available for (1) querying the dictionaries explicitly, (2) (re)calibrating the eye tracker, (3) turning the tracking on/off, (4) setting attributes of the eye tracker, (5) making user profile settings, and (6) adjusting the layout of the dictionary entry provided.

Depending on the eye tracker being used, turning the tracking on either first calls the calibration routine or, optionally, *iDict* starts to interpret eye input and the user can just start reading. Some of the new eye trackers are able to maintain personal calibration parameters, so that calibration is not needed at the beginning of a new session, even if substantial time has elapsed since the last use.

In the event that calibration is needed, the camera(s) must first be focused on the user's eye(s). After that, the user should concentrate on following the reference points automatically displayed on the screen. An experienced user performs the calibration in about one minute, but for an inexperienced user the process may, with current eye trackers, be toilsome and slow. After the calibration, tracking is automatically on and the user may start reading normally. During the session, the user can always call for recalibration (label 2 in Figure 5.2) if the tracker's accuracy appears to be too far off.

5.2.2 Automatic dictionary lookups

The user's gaze path is followed, and when the system discovers deviant reading behavior, the help function is automatically triggered. Observations with the early test readers revealed that they sometimes wanted to see also the optional translations for words. Moreover, even though the text is lexically and syntactically analyzed, it is virtually impossible to invariably choose the right translation automatically from among several options. That is why *iDict* was designed to give translations in two stages.

When a probable occurrence of difficulty in comprehension is identified, *iDict* automatically consults the dictionaries embedded in the system. It then displays a **gloss** for the problematic passage in the space between the lines right above the problematic point identified in the text (see Figure 5.3). The gloss is a short "best-guess translation" for the problematic point. The reasoning for selecting a particular gloss from the optional translations found in the dictionary is discussed in more detail in Section 6.4. The reader's problems may arise from a single problematic word or from a larger passage of text. The syntactical parsing performed for the text is able to suggest that the word may be part of an idiomatic expression, and the lookup for embedded dictionaries is first tried for the suggested word sequences. The words for which the dictionary lookup is performed are highlighted with color when the help function gets triggered.

The gloss can also be given as voice output, either along with the written translation or, if desired, as the only output format. The goal is that the reader is able to glance at (or listen to) the translation very quickly, without serious interruption to the reading process. The number of glosses remaining visible in the document frame can be set according to the user's preference: if, for example, only one gloss is set to be visible at a time, the

gloss is erased whenever a new one is displayed.

A reader who needs more information can get it by just turning the eyes to the dictionary frame area, and the whole dictionary entry appears. Figure 5.3 illustrates a situation in which iDict has identified deviant gaze behavior that is judged to stem from reading the word “regaled.” On the basis of the linguistic analysis performed for the text, the system knows that the word is a past tense form from the infinite form “to regale.” iDict gives “viihdyttää (kestitä) jkta jllk” as the gloss for the word in Finnish and displays it right above the word.

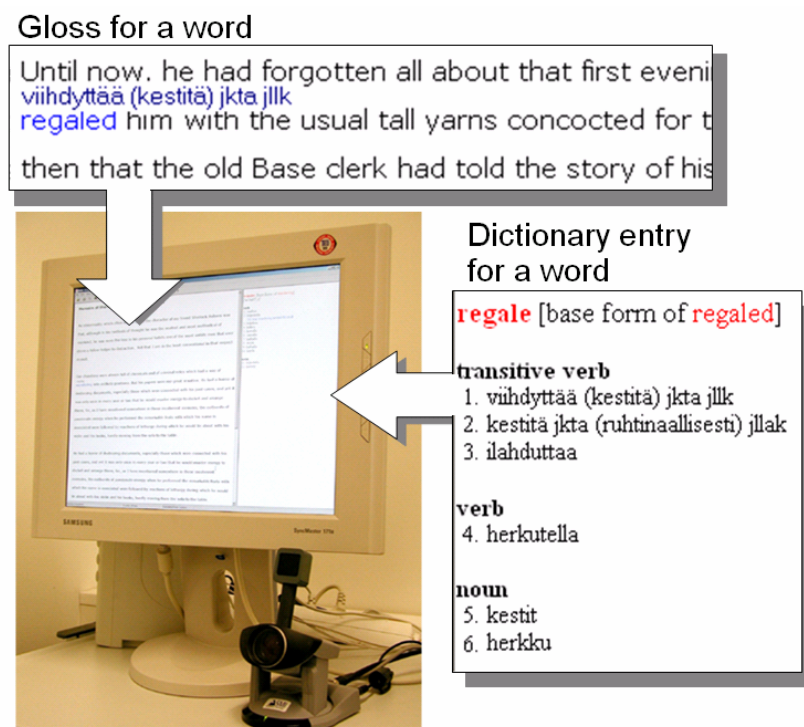


Figure 5.3: The two-level help provided by iDict.

In this example, the reader wants more information about the word and turns the eyes toward the dictionary frame. Consequently, a complete translation for the word appears in the dictionary frame as soon as the gaze enters this frame area. The translations for the verb when used as a transitive verb are given first, since the word was used as a transitive verb in the sentence in question. Translations for the word if used as an intransitive verb or as a noun are also presented.

5.2.3 Feedback

During reading, the gloss and, of course, the dictionary entry that appears are the primary means of feedback the user gets as an indication of the ongoing interpretation of eye movements in the background. However, the user has an option of making the use of eye input more transparent. A visualized point of gaze, the **gaze cursor** or the **line marker**, can be activated. The gaze cursor and the line marker are, in effect, “automated

reading sticks” rendering the interpretation of the progress of reading. The gaze cursor represents the measured real-time point of gaze in the text document during the reading process, and it is displayed as a small spot in the document frame area. The line marker automatically retraces the line of reading. It is displayed as a gray line under the “active line.” It reveals the system’s conception of the line being read at the moment.

Additionally, the reader might want the some indication of the words for which a gloss is about to be given. If the last form of visualization is activated, the words turn gray, not constantly to indicate the fixated word but only when a prolonged gaze on a word is observed, a little before the actual gloss for it is activated. In that case, the impression is that the eyes “push the words down” (analogous to clicking a button in a dialog), indicating the words for which the gloss will be given if they remain fixated.

Because of the possibility of inaccuracy, the system also gives the user the opportunity to use the arrow keys to manually correct the “active point” in the text, or, more precisely, the measured coordinates of the point of gaze. Up/down key presses correct interpretation of the target line by one line upward or downward, and, correspondingly, left/right key presses correct the target word one word left or right.

At the bottom of the application window, the status bar is used to give the user high-level feedback on the system’s status. The first part of the status bar, **an application status message**, displays information on the application’s routines that are currently in operation or those performed last. It may, for example, give the user information that syntactical analysis is being performed for the text or that a lookup for a word in dictionaries was just performed. An **eye tracker status message** displays information related to the eye tracker’s status – that is, if the tracker is being calibrated or if the eye tracker is on and eye input is in use. The third part of the status bar, a **translation status message**, informs the user as to which of the embedded dictionaries was used for the last lookup.

5.2.4 Optional mouse operation

Besides the gaze activation, the dictionary lookups can be activated also with a mouse. This leaves the user the choice of the desired input and also potentiates the application for studies of strengths and weaknesses of different input options (more details about mouse activation in the context of interface personalization are provided in Section 5.3).

When a word, or an expression, is searched for in the dictionaries, it is automatically added to a drop-down list of fetched translations (label 1 in Figure 5.2). The shortcut button to the right of the list acts as an additional interface to the dictionaries. The list can be used for retrieving previously translated words by picking them up from the list, but the text field is

editable in addition. Thus, the user can use it for retrieving a dictionary entry for any desired word from the dictionaries. This feature is especially useful in situations where the retrieved form of a word is not found in the dictionary. The user can edit the word and use the new form for performing a new dictionary lookup. For example, the compound word “fellow lodger” may not be found in the dictionaries, but a dictionary lookup for the word “lodger” (“asukki” in Finnish) probably reveals the meaning of the compound word, as well.

5.3 PERSONALIZING THE APPLICATION

iDict lets the users customize features of the interface according to their preferences. Additionally, the attributes that relate to the translation provided and to eye input interaction are maintained in a personal user profile. When entering the program, the user chooses the appropriate profile from the user profile list (Figure 5.4). The identification can be bypassed by setting the relevant user profile to be permanently active (via the “Always use this user profile” checkbox).

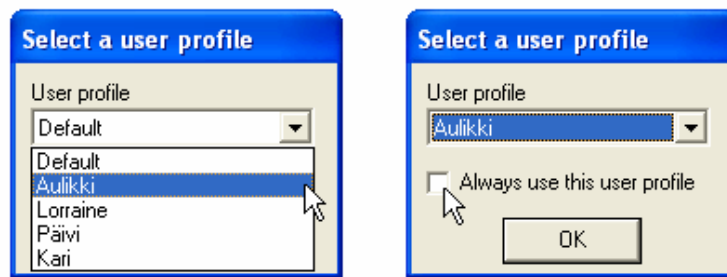


Figure 5.4: User profile dialog. Identification of the user is performed when the program is entered.

By choosing user profile settings from the Translation menu (or by using the shortcut, label 5 in Figure 5.2), a user can create a new profile (Figure 5.5), which then maintains the customized attributes. A new user can choose one of the existing profiles in order to copy some basic personal attributes. The system always contains at least one default profile that can be used as the basis for a new profile.

A user profile maintains three kinds of information on the user. These concern

- activation of dictionary lookups,
- presentation of the dictionary lookups, and
- target languages and dictionaries.

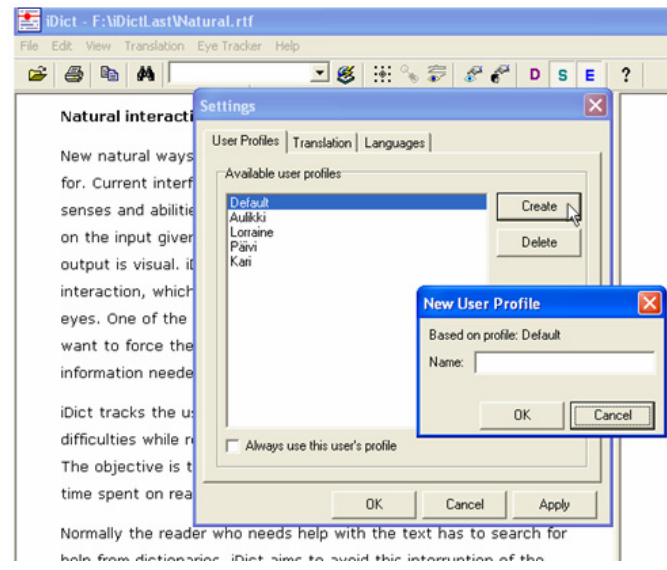


Figure 5.5: Creating a new user profile.

Figure 5.6 shows the dialog in which the settings for the first two attributes listed above are made. The language and dictionary definitions are described in the next section. As described above, the dictionary lookups are activated either by mouse or due to a deviant eye behavior. The activation methods are not mutually exclusive; both of them can be in use simultaneously. Gaze activation is always on if eye tracking is turned on in the main window. In addition to using gaze alone for triggering the dictionary lookups, the user can select a mode in which the point of gaze is used only to select the words for the lookup. In this case, pressing the spacebar triggers the lookup for the selected words. If gaze-alone mode is selected, the **triggering sensitivity** of the system can be tuned along a scale of 1 to 20. This affects the eagerness with which the system, on the basis of gaze behavior, interprets the user as having troubles with understanding the text.

Mouse activation for displaying a gloss and a dictionary entry, and also synthesized speech of a gloss, can be activated by selecting the corresponding checkboxes. There are alternative ways to use the mouse for activating the feedback: the action can be initiated by a mouseover event, by a single-click event, or by a double-click event.

Also, the number of visible glosses can be specified in the translation feedback dialog. It may be any preferred positive number. Often the selection is one visible gloss at a time, but some users may want to review previously retrieved glosses also. If multiple glosses are visible, their color is fainter the less recently they have been fetched; the oldest ones gradually “fade out” before totally disappearing. The color used for highlighting the words in the text for which the last glossary lookup was performed can be selected according to the user’s preferences.

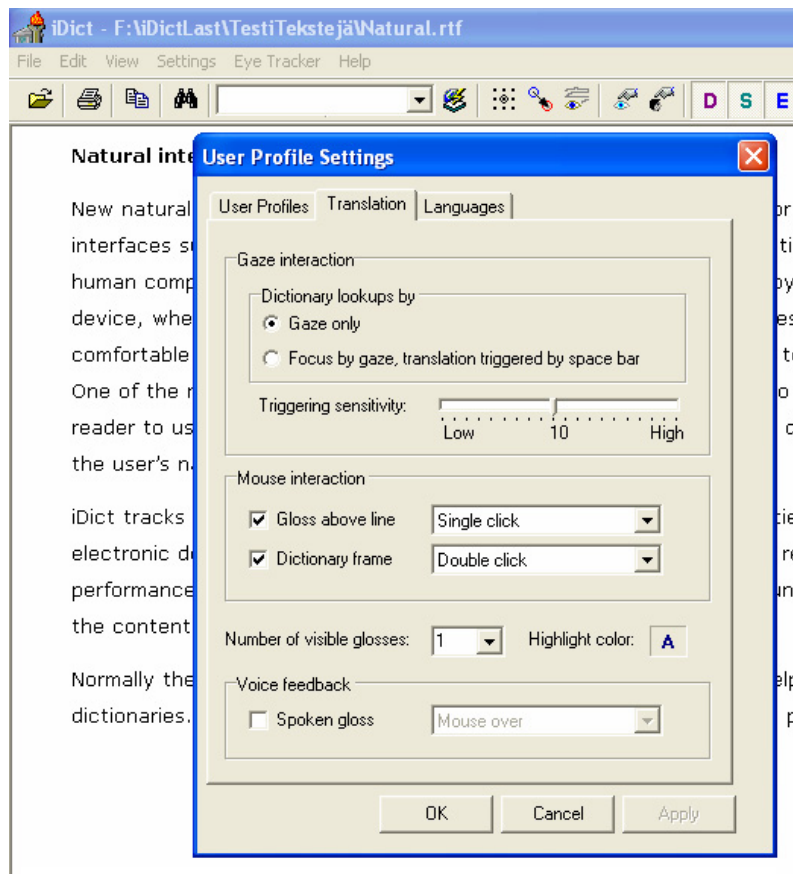


Figure 5.6: Translation feedback dialog. The user can personalize the mode of help provided.

5.4 SPECIFYING THE TARGET LANGUAGE AND DICTIONARIES USED

On the Languages tab (Figure 5.7), the user can specify the target language (the language into which the problematic text passages are translated) and the dictionaries to be used during the reading session. The selections are saved in the user's profile for subsequent reading sessions.

The source language (the language in which the document being read is written) of the implemented version of iDict is English, but the target language may be Finnish, Italian, German, or English. At the moment, there are two commercial dictionaries integrated with iDict, the WSOY (2000) and Sandstone (2001) dictionaries. In addition, the **custom dictionary** is a dictionary into which the user can save translations of words or idioms him- or herself. With English–English chosen as the language pair, the dictionary provides definitions for the requested word(s) in “other words.”

The dictionaries are consulted in the order defined by the presented list. For example, in Figure 5.7 the dictionaries that are available (and also all of them are checked to be in use) are custom, WSOY, and Sandstone dictionaries. Their order in the list defines that the custom dictionary is consulted first, and if a translation for the passage of text in question does not exist in the dictionary, the next dictionary in the list is consulted for the translation. The lookup order of the dictionaries is customizable via the application's initialization file.

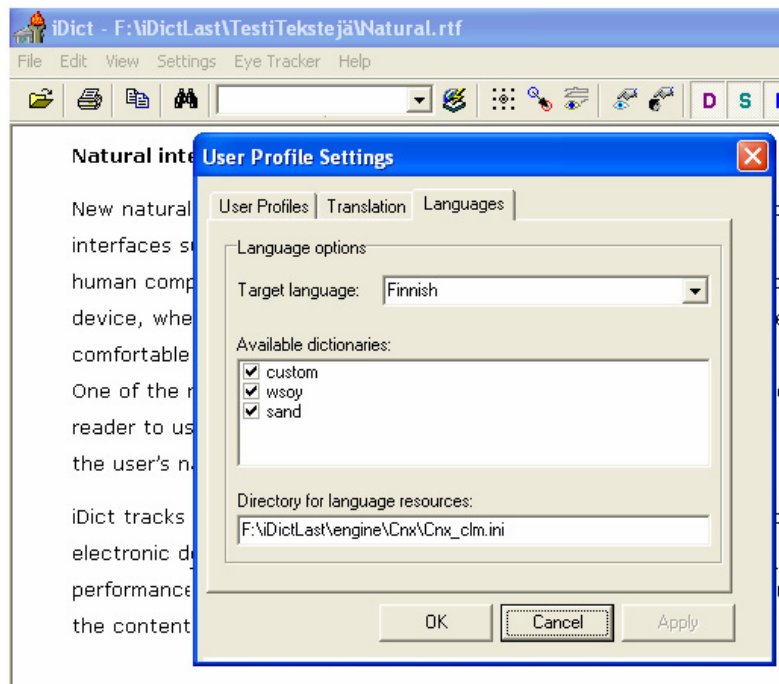


Figure 5.7: Languagesdialog.

Of course, not all of the embedded dictionaries support all of the available target languages. The list of dictionaries available for the selected target language is updated once the user specifies the target language. Currently, the English–English language pair is supported only for selected text documents by the custom-created dictionary. English-to-Finnish translations can be retrieved from all three dictionaries, and English-to-German and English-to-Italian translations are supported with the custom and Sandstone dictionaries.



6 iDict Implementation

The primary problem whose solution potentially serves implementation of gaze-aware applications in general is how the reading process can be interpreted in real time on the basis of tracked gaze behavior. The issue is addressed in the third part of this dissertation. In this section, we describe the more general issues of iDict's implementation (those not related to interpretation of gaze behavior). The eye trackers used during the development of the application are first briefly introduced (Section 6.1). Then, after an overview of the iDict architecture (Section 6.2), the preprocessing of the document opened for reading in order to construct an internal representation of its layout, is described (Section 6.3). The internal representation is needed to enable the **mapping** of the recorded point of gaze and the objects in the text document. The text is parsed to obtain syntactical and lexical information, in order to increase the correctness of the dictionary lookups provided (Section 6.4). Finally, the features embedded in the application to enable reviews of gaze paths for reading sessions (for research purposes) are described in Section 6.5.

6.1 EYE TRACKING DEVICES USED

Three different eye trackers were used during the development and implementation of iDict. EyeLink¹ (SR Research, 2005) was used in the development phase in designing the algorithms for interpreting eye behavior because EyeLink has a very high resolution in terms of both time and space. However, since EyeLink uses head-mounted optics for

¹ *In fact, the eye tracker used was EyeLink I, which is no longer available. SR Research has continued its development with a descendant of this tracker (EyeLink II).*

capturing the image of the eye (see Figure 6.1), it is not acceptable for normal use: it is not reasonable to assume that a user would bother, whenever opening a text written in a foreign language, to put on the head band and to calibrate it for getting the eye movement information passed to the application appropriately. That is why iDict was ported also to two eye trackers that use remote optics for monitoring eye movements: to iView X (SMI, 2005) and to Tobii 1750 (Tobii Technology, 2005).

6.1.1 EyeLink

EyeLink is an eye tracker with head-mounted optics. The two cameras (see Figure 6.1) record video images of both eyes, and the images are then processed for identifying the locations of the pupil in each of the recorded images. Calibration is needed prior to each tracking session. The



Figure 6.1: EyeLink.

calibration includes setting the cameras so that they are properly aligned in relation to the eyes of the user. After that, the user has to follow the reference points displayed on the screen. The head movements of the user are compensated for with a separate IR-based system.

The **temporal resolution** of EyeLink is 250 Hz, and the **spatial resolution** is reported to be within 0.01 degrees. Temporal resolution refers to sampling rate; thus, EyeLink records a sample of

the gaze position every four milliseconds. The spatial resolution that eye tracker manufacturers usually report refers to the resolution to which the eye position can be detected from the image of the eye, thus reflecting the camera resolution rather than the resolution of the real point of gaze on a screen. As was discussed in Chapter 2, the resolution of point of gaze, if it refers to the point having our attention, is only between 0.5 and two degrees of the visual field (even if the spatial resolution reported by the manufacturer is higher).

6.1.2 iView X

In iView X, the optics are placed on a table in the proximity of the screen (Figure 6.2). iView X tracks only one eye. One-eye tracking is sufficient for tracking the point of gaze because the two eyes move in synchrony. Like EyeLink, iView X requires calibration prior to each tracking session, and the first step of the calibration is to orient the remote camera such that it detects the user's eye. After the coordinates of the screen are set for the tracker (the user follows the reference points on the screen), the servo mechanism on top of which the camera is positioned enables iView X to follow moderate head movements of the user.

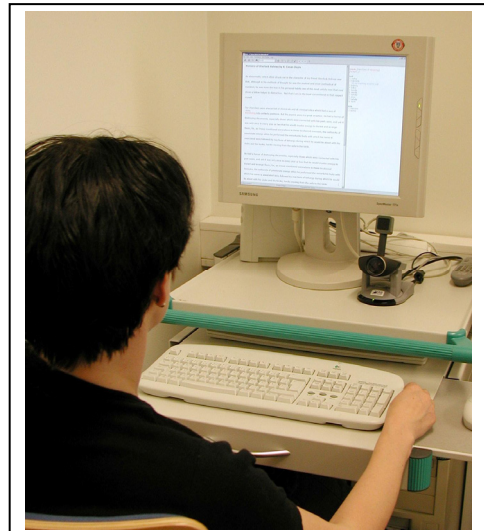


Figure 6.2: iView X.

The temporal resolution of iView X¹ is 50 Hz, and the spatial resolution is reported to be within 0.025 degrees.

6.1.3 Tobii 1750

In Tobii 1750 (Figure 6.3), the eye tracking system is integrated into the display hardware. EyeLink and iView X require two different computers, a subject computer that runs the gaze-aware application and an operator computer running the eye tracker software. In addition to the custom-made display, Tobii needs only one computer (a desktop or a laptop computer), which makes both installation and use of the tracker simpler. The large-field-of-view camera tracks both eyes of the user, with comfortably robust tracking that allows large natural movements, of about 20 x 15 x 15 cm (horizontal x vertical x depth – Cheng & Vertegaal, 2004), without loss of calibration. Tobii requires only one calibration for a user, which is saved and thereafter obtained from the user's personal profile.



Figure 6.3: Tobii 1750 (Tobii Technology, 2005).

The temporal resolution of Tobii 1750 is 50 Hz. The spatial resolution for Tobii has not been cited, but the accuracy of gaze position is reported to be 0.5 degrees.

¹ In the version of iView X used, the temporal resolution is 50 Hz; for the recent iView X trackers, the temporal resolution is higher, varying from 240 to 350 Hz.

6.1.4 Preprocessing of sample data

Straightforwardly using the raw sample points in iDict would have been problematic. First, the number of raw gaze positions would have been responsible for causing delays upon execution of the algorithms mapping the gaze positions in real time to the objects in the text. Second, the miniature eye movements might in some cases end up changing the mapped text object even if the fixation in reality continues. That is why fixations were identified on the basis of raw sample data and were passed on to the iDict algorithms.

For both EyeLink and iView X, we used the real-time fixation detection functions provided by their application programming interfaces. Salvucci and Goldberg (2000) recognize **velocity-based** and **dispersion-based** fixation detection algorithms for identifying fixations by using spatial characteristics of the measured sample points¹. Velocity-based algorithms take advantage of the fact that movements inside a fixation have low velocity and movements ending a fixation have high velocity. Dispersion-based algorithms, on the other hand, emphasize the physical spread between fixation points, under the assumption that sample points belonging to the same fixation generally occur near one another. EyeLink provides real-time fixation detection functions, in which the fixation is computed mainly on the basis of the velocity, even though in some cases also spatial change of gaze position or acceleration of the eye movement may break a fixation. The real-time fixation detection of iView X is grounded in dispersion-based algorithms; a fixation is computed on the basis of the distance of sample points, but additionally a minimum fixation time is used to filter out too short fixations.

The Tobii API does not provide (at least at the moment) on-line fixation detection, so we implemented our own dispersion-based fixation detection algorithm for the tracker. As long as the sample points stay within a threshold radius from the fixation center calculated as currently applicable, the fixation is judged to continue. A threshold for minimum fixation duration was also used, to discard very short fixations.² The sample points we got for the fixation detection through the application programming interfaces of Tobii were already filtered to exclude digressive sample points, which were probably caused by transient failures in measuring the point of gaze.

¹ They add to the list also the “area-based” algorithms, but in that case the fixation identification is not general but uses information on the given areas of interest (AOIs) of the application as well.

² The threshold value we used in the tests was 30 pixels, and the minimum fixation duration used was 70 ms.

Much discussion has addressed how the fixation detection should be performed. The issue is relevant especially in the psychology research field, in study of the human sensory and motor system. In our case, the more practical angle calls for a more liberal point of view concerning fixation detection, since the algorithms provided by different tracking devices may differ somewhat. In the experiments with various tracking devices, we settled on always confirming that the distribution of fixation durations during reading did not notably differ from those reported in the literature (described in Section 4.1).

6.2 OVERVIEW OF THE iDICT ARCHITECTURE

Figure 6.4 illustrates the architecture of iDict. The development environment used for the implementation was Borland Builder (and C++). The environment was chosen mainly due to Builder's efficient tools for implementing the user interface. In order not to constrict the development of later versions of iDict to Builder, we pursued a goal of keeping the user interface separate from the rest of the application. Also, eye trackers should be easily interchangeable. That is why the modules that feed in input for the application (**User Interface** and **Eye Tracker**) interact with the rest of the application via a **Message Manager** module.

The core of the application, **iDict Engine**, contains the algorithms that interpret the reading process. It was designed to be independent of the implementation environment. The goal was to facilitate its reuse in similar applications, which may have a different interface and may, or may not, use linguistic analysis and different sets of lexica, as well as use various eye trackers.

iDict Engine consists of three modules. **Document Manager** annotates the text with layout and linguistic information, **User Profile Manager** personalizes the application for the user, and the **Intention Extraction Module** (IEM) monitors the reading process.

When a document is opened for reading, Document Manager preprocesses the text and saves it in an internal dynamic data structure that reflects the layout of the text document.

It seems obvious, and the reading research affirms this assumption (e.g., Just & Carpenter, 1980; Hyönä, 1995), that the complexity of the text affects the gaze behavior. Our original hypothesis was that reading behavior differs from individual to individual. Therefore, User Profile Manager was added to iDict to maintain individually tuned features related to the user's reading habits.

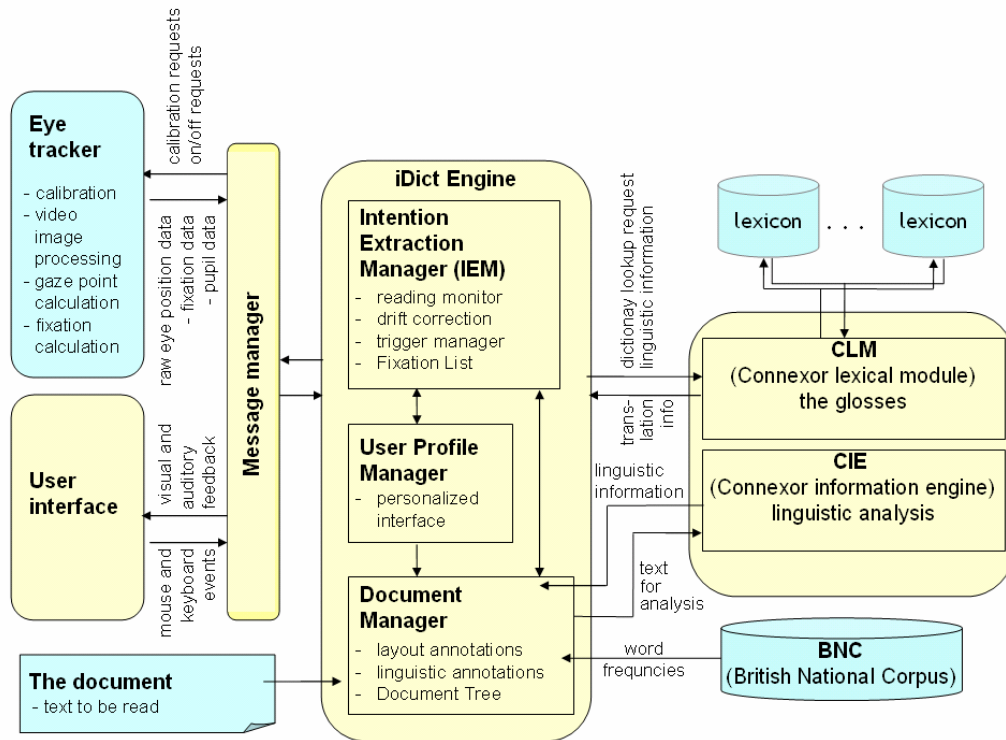


Figure 6.4: iDict architecture.

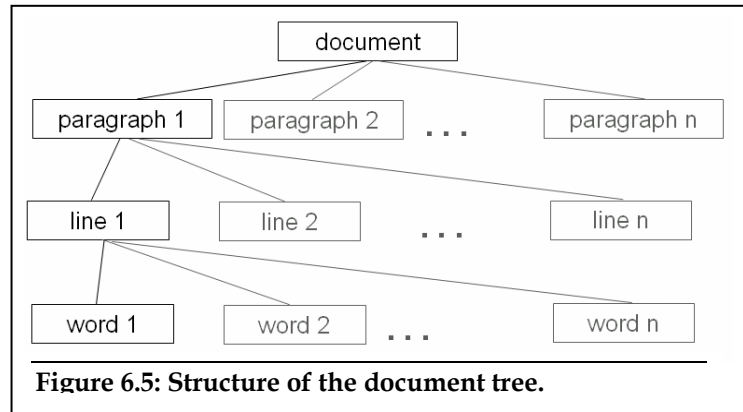
The Intention Extraction Module makes use of existing knowledge of eye behavior during the reading process. The naming of the module, though provocative from the cognitive point of view, reveals the goal of the module: it aims to interpret the reading path for detecting the divergent behavior that is presumed to expose the difficult points in the text. In order to be able to map the fixations to the correct target words, it also makes an effort to correct error in the measured eye position. The drift compensation algorithms are discussed in more detail in Chapter 7. The reading path is saved to the **fixation list**, a list containing a chronological history of the reader's fixations and the target words of the fixations.

The **Connexor Lexical Module** (CLM) implements the interface for the lexica by performing dictionary lookups for the requested words or word sequences.

6.3 TEXT DOCUMENT PREPROCESSING AND MAINTAINING OF THE SESSION HISTORY

iDict keeps track of the words that are fixated upon during a reading session. In order to know the currently and previously targeted words, the system must be aware of the text layout and maintain information on the past reading path.

The Document Tree (Figure 6.5) is the internal representation that Document Manager creates upon opening of the text. It contains the **text objects** (paragraphs, lines, and words) organized into a hierarchical dynamic data structure on the basis of the regions the objects occupy on screen. All text objects are aware of their position on the screen. The structure facilitates rapid searching for the



target object of a fixation. In addition to the layout information, the words of a document (the lowest-level text objects in the hierarchy) are associated with linguistic and lexical information obtained from the **Connexor Information Engine** (CIE) component, designed by Connexor Oy¹) and from the **British National Corpus** database (BNC, 2005). The document tree also maintains a history of fixations targeted for each of the text objects.

Each text object knows the region it occupies on the screen. The **object mask** is originally the smallest rectangle that encloses the object (Figure 6.6).

A fixation is mapped to a word if the fixation's coordinates locate inside the word mask. The unallocated space between object masks together with the inaccuracy characteristic to eye tracking complicates this simple principle. Due to the inaccuracy, the masks are not stable but vary as a result of reading path history. Mapping the fixations to their target objects is described more closely in Section 7.1.

The chronologically arranged fixation list does not provide direct access to the fixation history of any given word. That is why each of the word objects maintains the history of its "own" fixations in the document tree. So, when fixated upon, a word promptly knows its fixation history without forcing iDict to scan through the whole list of previous fixations from the fixation list. The linguistic features associated with the word objects are introduced next.

¹ For information on Connexor Oy, see <http://www.connexor.com/>.

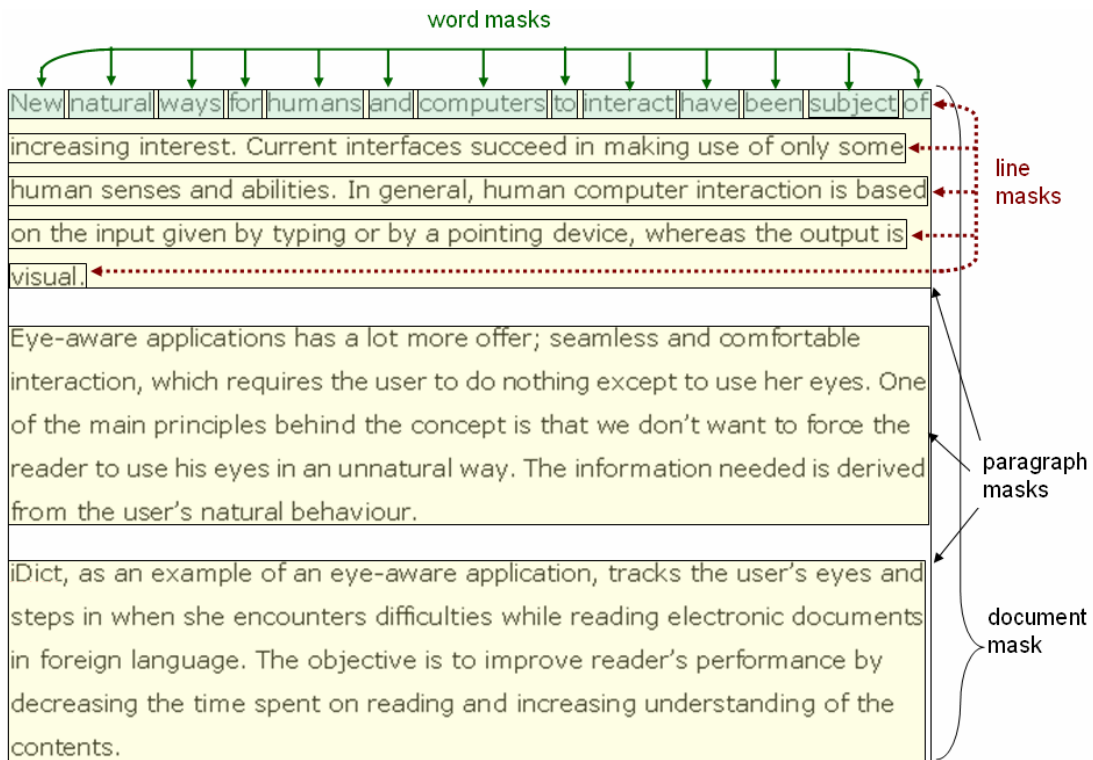


Figure 6.6: Text object masks.

6.4 LINGUISTIC AND LEXICAL PROCESSING OF A TEXT DOCUMENT

A design principle iDict is to enable the user to get the needed information at a glance, minimizing interference to the reading process. Therefore, one of the key issues is the quality of the gloss provided. Naturally, an inaccurate gloss interrupts the reading and confuses the reader when he or she tries to match the faulty semantic contents with the sentence.

iDict uses the CIE component, designed by Connexor (review Figure 6.4) to perform syntactic analysis of the text document. The CIE module is implemented on the basis of Connexor's Functional Dependency Parser, or "FDG parser" (Tapanainen & Järvinen, 1997), implemented as a COM (Common Object Model) server. The engine produces context-dependent linguistic information that we use to focus the query for the dictionaries and thus improve the quality of the gloss retrieved. In the following section, the linguistic analysis is described from a practical point of view: what information linguistic analysis provides for iDict and how the information is used.

6.4.1 Linguistic analysis

If the words of the text document were used in their original form and order as the query for the embedded dictionaries, the result of the query would in many cases be empty. For example, the citation form of a verb in dictionaries is usually the infinitive form and citation of a noun is in nominative singular form. Thus, the words in the text document must first be transformed into their base forms. For example, the lexical headwords for the dictionary lookups for the sentence “Making promises was easier than expected” should be “make”, “promise”, “be”, “easy”, “than”, “expect.”

Even when a word is transformed into its base form, the dictionary lookup for a single word is likely to yield erroneous translations, on account of lack of consideration of the context in which the word is used. The syntactic information produced via the syntactic analysis can be used for choosing a “best guess” for the right translation and sorting the alternative translations according to their probability of matching the context.

Morphological and syntactic information

The CIE component parses the text into sentences, and it determines the base form and also the word class (the part of speech), along with additional explicatory grammatical and syntactical information for each word in the sentence. By using the word class information, we are able to restrict the appropriate translation space substantially.

For example, consider the case in which a translation is wanted for “bear” or “park” in the following sentences:

Should all people have a right to keep and bear arms?

Should all people have a right to keep and arm bears?

Where can I park my car?

On the basis of the word class information, translations are not given for the noun “bear” (~ a “bruin,” a big animal) in the first, for the verb “bear” (~ to “carry”) in the second, or for the noun “park” (~ a “garden”) in the third sentence.

An example of the additional grammatical information sought (called a “subclass” in the CIE component) is the further specification of verbs as transitive or intransitive according to their capability of taking an object. Transitive verbs may be assigned an object, but intransitive verbs may not. The same verb can often adopt both roles, in different contexts, but the translation varies with the role adopted. For example, the verb “show” in the context of the first sentence below is transitive, but it is intransitive in the second.

Figure 4 shows the components of the system.

It does not show at all.

The transitive “show” translates into “näyttää, esittää, kuvata” in Finnish, whereas the intransitive “show” translates into “näkyä, näyttäytyä, vaikuttaa.”

Table 6.1 shows the word class information CIE supports (the tags presented in the list are used in an example sentence analysis later).

Main word classes		Subclass information	
A	adjective	V aux	auxiliary verb
ADV	adverb	V obj	transitive verb
CC	coordinative conjunction	V dat	dative verb
CS	subordinating conjunction	V refl	reflective verb
DET	determiner	N abbr	abbreviation
INF	infinitive marker	N prop	proper noun
N	noun	N sg	singular form
NEGPART	negative particle	N pl	plural form
NUM	numeral		
PREP	preposition		
PRON	pronoun		
V	verb		

Table 6.1: Word class information supported by CIE.

Identification of potential translation units

One of the substantial strengths of the CIE component is that it identifies the units of text that potentially can be translated as compounds or idiomatic expressions. The multiword expressions CIE is able to point out are given in Table 6.2.

NN	noun and noun compound
AN	adjective and noun compound
V-phrase	phrasal verb and particle
V-idiom	possibly idiomatic verb and object
P-idiom	possibly idiomatic prepositional phrase

Table 6.2: Compound and idiomatic expressions supported by CIE.

For example, in the sentence

It went on to the very end; I had to back off.

CIE identifies three multiword expressions: the V-phrase “went on,” the P-idiom “to the very end,” and the V-phrase “back off.”

If the reader were to need help with the phrasal verbs or with the idiom, without CIE’s multiword expression analysis iDict would end up giving translations separately for each of the words “go,” “on,” “very,” “end,” “back,” and “off.” Now an accurate translation can be given by consulting the dictionary using the CIE component’s normalized (see the next

section) form of the phrasal verb “go on” (“jatkaa” in Finnish), the four-word idiom “to the very end” (“loppuun asti”), and the phrasal verb “back off” (“perääntyä”).

6.4.2 Dictionary lookups

The queries are performed through the Connexor Lexical Module component. It standardizes the dictionary interface and makes it easier to integrate iDict with new dictionaries. As is CIE, CLM is implemented as a COM component.

Normalized base form

For the dictionary queries, iDict uses the normalized base forms of words provided by CIE. The normalized base form is the format used most commonly as the citation form in dictionaries.

Normalization of a single word transforms the word into its base form. Also the compounds and phrasal verbs are transformed into their base forms (e.g., “information societies” → “information society”; “went on” → “go on”). Most of the idiomatic expressions are retrieved from the dictionary in their text forms, but personal pronouns in idiomatic transitive verb expressions (V-idioms) are identified and transformed into the form usually found in dictionaries. For example, in the sentence

Could you please tune my piano?

CIE identifies the V-idiom “tune my piano” and standardizes it into the form “tune one’s piano,” which is then used in the dictionary query.

Dictionary query

The CLM component takes a query as input and returns the information it fetches from an embedded dictionary. Table 6.3 shows CLM input and output format.

CLM input (the query) <ul style="list-style-type: none">- identification of the dictionary to be consulted- normalized base form of the word(s) to be retrieved- syntactical and grammatical information on the word(s)- specifier (the text’s genre)
CLM output (the dictionary information) <ul style="list-style-type: none">- translation for the word(s)- grammatical information (word class and subclass information)- pronunciation information- synonyms for the translated word(s) (in the target language)- definition of the word(s) (textual definition)- example sentences (illustrating the context of use in the source language)- word frequency information- field reserved for future use

Table 6.3: Format of CLM input and output.

The query identifies the dictionary addressed, gives the word(s) for which the translations are fetched (in normalized base form), and passes on the syntactical and grammatical information retrieved through CIE analysis. The last piece of information, specifier, was included for considering the text's genre (i.e., its domain – technical, biological, psychological, etc.), but this information is not yet in use.

The module returns the seven fields of information listed in Table 6.3 for each translation retrieved from the dictionary. Obviously, the fields for which information does not exist in the accessed dictionary are empty. The information in the first six fields is displayed in the dictionary frame with a layout that imitates the format familiar from printed dictionaries (review Figure 5.3). The word frequency information¹ is used in combination with the word class information for choosing the most probable gloss for the user's needs. The information is also used to sort the optional translations in the dictionary frame, with preference given to the right word class and, within the word class, with sorting according to the frequency of the word.

Dictionary resources

Custom Dictionary was implemented to conform to the **Custom Dictionary Format (CNF)**, which contains the information that the CLM module is able to return. The format also facilitates the integration of new dictionaries: only the modification of the retrieved dictionary information to fit the CNF format is needed. Custom dictionaries were used for experimenting with *iDict*'s linguistic features. Additionally, the user of *iDict* can use the custom dictionary as a personal translation repository (for example, to add translations of new idioms to the dictionary).

Connexor converted two commercial dictionaries into the CNF format (the WSOY dictionary and the Sandstone dictionary). Unfortunately, neither of the dictionaries was ideal for *iDict*. The Sandstone dictionary was selected because it supports several language pairs, including English to Finnish, Italian, and German. However, this dictionary does not contain grammatical information that could be used to restrict the query (for example, word class information is not included). Because the use of linguistic analysis is a significant feature in *iDict*, Connexor also converted the WSOY (English-to-Finnish bilingual) dictionary into CNF format. It contains much information that was considered to be of use in demonstrating the capacity of *iDict*. The problem with the WSOY dictionary is that it is based on a printed dictionary, and parsing the information from the dictionary proved to be more laborious than was

¹ *CLM uses word frequency information based on a text document database containing 36,000 newspaper articles.*

expected. Also, the parsing often fails to extract a lot of the information that in principle would be very useful for iDict.

In addition to sorting out the “best guess” glosses, iDict would be capable of passing versatile translation information to the user in the dictionary frame. The ultimate quality of the translation provided is, naturally, dependent on the quality of the embedded dictionaries. We expect that dictionaries better supporting this kind of information – in an electronic form that would make it possible to fully exploit the potential of iDict – will be available in the future.

6.4.3 Example of linguistic processing of a sentence

When a document is opened in iDict, Document Manager (review Figure 6.4) first structures the text into the dynamic tree structure, in which every word is an object. Document Manager then sends the text to CIE and gets back analysis for the whole text. After this, Document Manager annotates each word with the appropriate linguistic information. Later on, if a translation for a word gets triggered (i.e., a dictionary lookup for the word is requested), the word, together with the attached linguistic information, is passed on to the CLM module.

The analysis given by CIE for the example sentence presented earlier

1 2 3 4 5 6 7 8 9 10 11 12 13 14

It went on to the very end; I had to back off.

(the words are labeled with their position in the sentence) is given in Table 6.4. Each line contains a text token, its normalized base form, the word class, and the possible subclass or categorization of a multiword expression. Lines end with a reference to the token’s position in the text. Each of the words in a sentence is represented in at least one analysis line. The analysis information is only a subset of the information that the FDG parser would be able to give; only the information considered valuable for iDict was included in CIE.

As described earlier, the example situation and the analysis provide lexical and syntactic information not only for each individual word but also for the three idiomatic expressions “went on,” “to the very end,” and “back off.” In addition, CIE identifies “very end” as a compound word formed by an adjective and noun

It :base it :PRON 1
went on :base go on :V-phrase 3 2-3
went :base go :V 2
on :base on :ADV 3
to :base to :PREP 4
the :base the :DET 5
very :base very :A 6
to the very end :base to the very end :P-idiom 7 4-7
end :base end :N 7
very end :base very end :AN-comp 7 6-7
; :pun 8
I :base I :PRON 9
had :base have :V 10
to :base to :INFMARK 11
back off :base back off :V-phrase 13 12-13
back :base back :V 12
off :base off :ADV 13
. :pun 14
:sent 15

Table 6.4: CIE analysis for the example sentence.

iDict attaches to each word in the document tree a chain of linguistic information of tokens (the tokens that include the word). The chain is ordered from the widest token to the linguistic information for the word itself. For example, the word “very” has attached the following chain of linguistic information for the tokens: “to the very end,” “very end,” and “very” (lines 8, 10, and 7 in Table 6.4). If the word “very” then gets triggered, the dictionary lookups are performed in the order of the list of linguistic information: first the active dictionaries are consulted (in the order specified by the user) for the token “to the very end”; if a translation is not found, a lookup is performed for “very end.” The last piece of linguistic information in the list is always for the one-word token. So, if all attempts to find translations for larger text units fail, the translation for the word itself is given. In this case, the word “very” is the last linguistic token in the chain.

6.5 TEST BED FEATURES

One of the main purposes in implementing iDict was for use as a test bed for studies of the use of eye input. That is why the prototype version had embedded a selection of functions that enable monitoring the application’s runtime processes. These test bed features are hidden in a debug menu that is usually invisible but can be made visible by a developer when needed. The test bed features include, for example, tools for checking the results of linguistic analysis performed for selected pieces of text and for changing parameters of the algorithms that interpret the reading process. The most important of these tools is the test environment for replaying

6.5 Test bed features

reading sessions (Figure 6.7).

The raw eye position data and the fixation data acquired from the eye tracker can be saved to external data files. The fixation data file can then be used for replaying the reading session, fixation by fixation, in the test environment. The gaze path is visualized via circles (fixations) and lines connecting them (saccades). The radius of the fixation circle is relative to the length of the fixation. The gaze path is displayed on top of the text document that was read in the original reading session. When the fixations are stepped through, the information associated with each fixation is displayed in a separate dialog window.

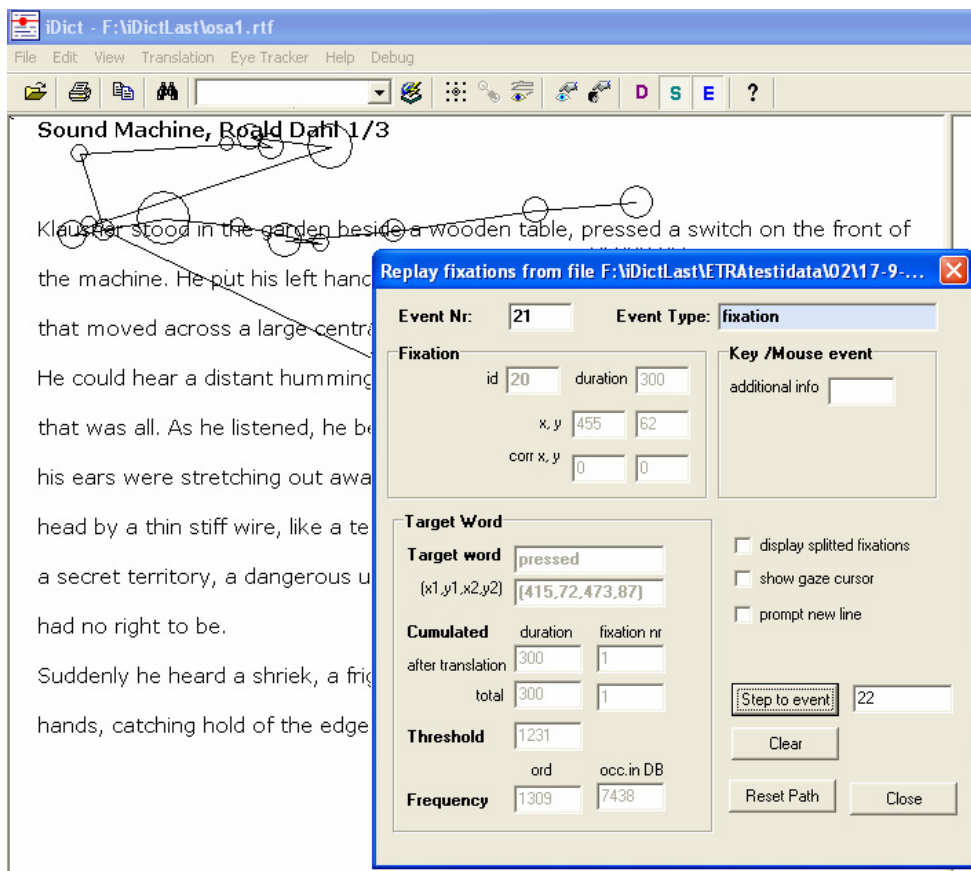


Figure 6.7: iDict test environment.

These test bed features were heavily used during the development of eye input interpretation algorithms. All of the figures that illustrate reading paths in the following chapters were produced with the test environment.

.....



Part III

Using Gaze Paths to Interpret the Real-Time Progress of Reading

- Chapter 7** Inaccuracy in Gaze Tracking
 - Chapter 8** Keeping Track of the Point of Reading
 - Chapter 9** Recognizing Reading Comprehension Difficulties
 - Chapter 10** Interaction Design of a Gaze-Aware Application
 - Chapter 11** Evaluation of iDict's Usability
-

reasons for errors in mapping a fixation on the word being processed during a reading session originate from at least the following three sources:

1. Measurement inaccuracies – the accuracy of measured point of gaze depends on the eye tracking device used and the success of the calibration performed.
2. Drift from calibration – inaccuracy also originates from imprecise compensation for head movements and from change in the size or shape of the measured characteristics of the eye.
3. The nature of the eye – even if no error is caused by the other two factors, we cannot be absolutely sure which word has the visual attention at any given point in time, since the reader can focus the visual attention without moving the eyes (as discussed in Subsection 2.1.1).

Hence, even if the development of technology will allow us to track reading of material in a smaller font size and with tighter line spacing, the inaccuracy problems in monitoring the gaze path during reading will still remain. A common size for text read from the screen is 11–14 pt with 1.5 times line spacing. For example, the height of a capital letter displayed in 11-point Verdana on a 17" screen, when 1024 x 768 resolution is used, is about 3.5 mm. Viewed from a distance of 60 cm, it covers a visual angle of 0.3° . The height of a single line of that text would be 6.3 mm, which ends up covering a visual angle of 0.6° . This means that actually, knowing the coordinates of a single fixation is not sufficient for determining which line of text the reader perceived, even with perfect eye trackers.

7.2 EXPERIENCES OF READING PATHS IN PRACTICE

Figure 7.1 represents a typical example of a recorded reading path. The gaze path appears to be a mess. In the upper right corner are fixations during which the reader seems to have fixated on empty space. Similar empty-space fixations occur frequently in the figure. We are very unlikely to spontaneously focus on an empty space, so the measured fixation locations must be inaccurate. If that is the case, what can we deduce on the basis of such inaccurate information? In this section we explicate how the inaccuracy can be deconstructed into more controllable sub-problems.

7.2 Experiences of reading paths in practice

A more systematic inspection of the fixation path in Figure 7.1 starts to make some sense of the mess. For example, there are successive horizontally progressing fixations that can be assumed to represent reading of a line. In some places, the fixations have piled up, which represents delayed reading – possibly difficulties in understanding.

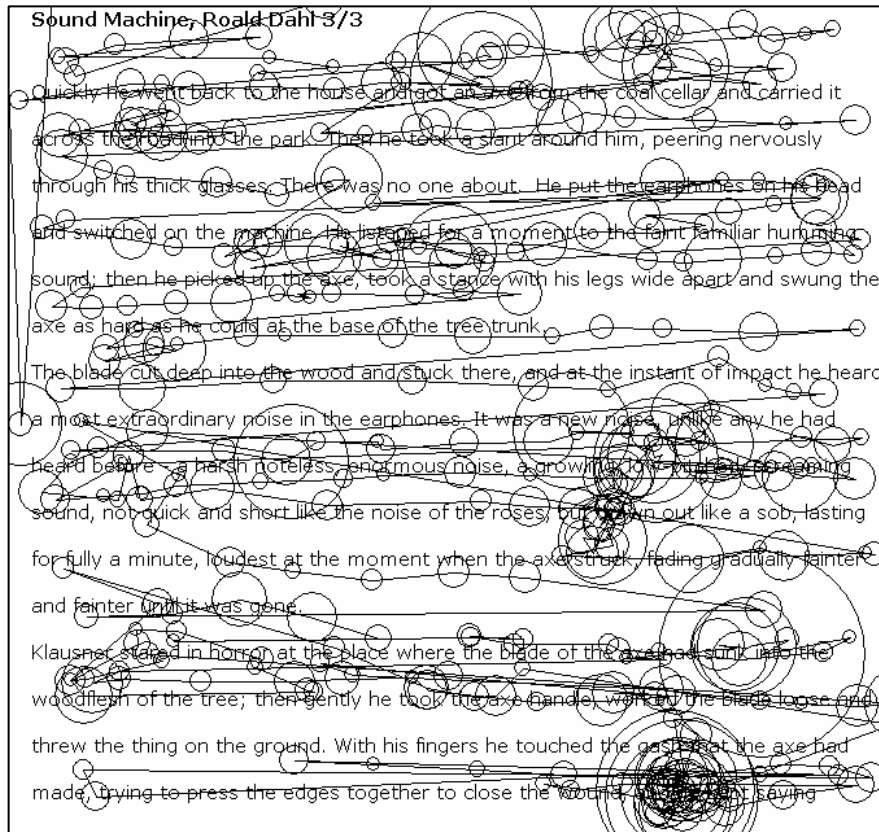


Figure 7.1: An example gaze path in reading a passage of text (recorded with iView X).

Figure 7.2 presents an example of a more successful recording of a reading session: the recorded reading path appears to have no significant accuracy problems.

Calibration has obviously been successful, and not much drifting from the calibration values has occurred during the reading session. Only at the bottom of the text window, for reading of the last line of the text, does there appear to be a vertical error in the measured locations of fixations. Judged from the leftmost and rightmost fixations on each line, the fixations' locations appear to have no significant errors horizontally, either.

Next, we give examples of vertical inaccuracy, followed by discussion of horizontal inaccuracy.

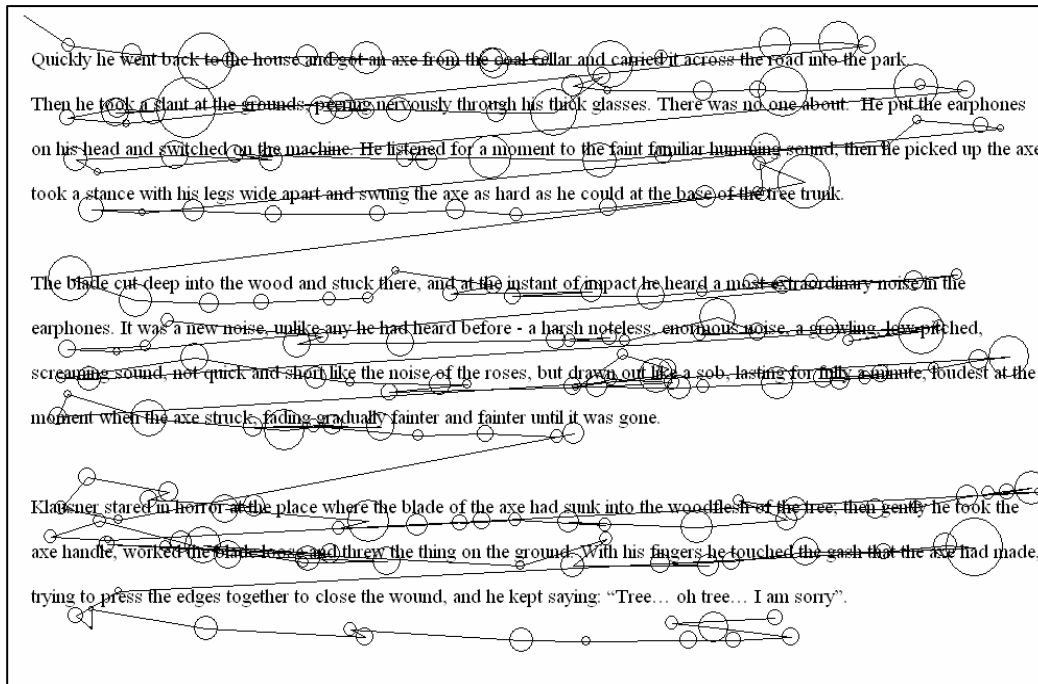


Figure 7.2: Successfully tracked reading session (recorded with EyeLink).

7.2.1 Vertical inaccuracy

Figure 7.3 shows an example in which tracking of reading of the lower line begins quite accurately but the measured fixation locations rise as reading proceeds. If the fixations were mapped straightforwardly to the closest word (to whose mask window the distance is shortest), all fixations from 9 to 18 would have been mapped to the upper line.

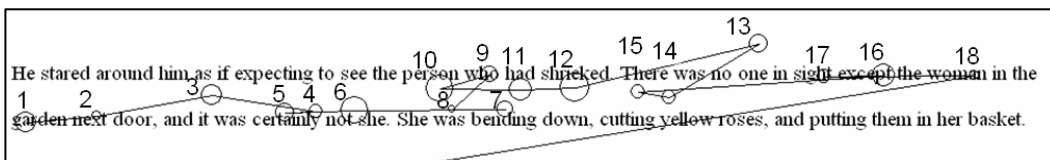


Figure 7.3: A raising reading path (EyeLink).

However, reviewing the reading paths of the previous and successive lines makes it obvious that the lower line was read with the successively numbered fixations 1–18. The line above was already read before fixation 1, and the reading of the next line after fixation 18 was easily identified in the post-analysis of the whole reading path. Fixation number 13 can be an exception, but if the reader really fixated on the word “one” in the upper line, it was probably a mistake and the reader returned to the original line with fixation 14.

Such ascending fixations during reading of a line were common, especially when EyeLink was used. This may be due to bad calibration in the region, to the reader’s facial muscle activity, or maybe to some changes

in the size or shape of the reader's pupil. In this case, it would be tempting to believe the last, because the error appears, interestingly, after a regression. Regressions reveal a need to return to check something already read and thus may be a sign of a greater cognitive load, which has been found to affect pupil size (Hyönä, Tommola & Alaja, 1995). Checking the hypothesis against the recorded data reveals that, indeed, the average pupil diameter is about 10% smaller during fixations 1-8 than during fixations 9-18. However, since the trackers we used do not give very reliable measurements for pupil size (e.g., there is no compensation for changes in distance), the hypothesis is not validated in the context of this dissertation. Nonetheless, it would be an interesting topic for further study.

Even though ascending reading paths were common in our experiments, in some cases the path was seen to drop during the course of reading a line (as during reading of the last line in Figure 7.2). In some cases, the rising (or dropping) path could also regain its accurate position, as is shown in Figure 7.4.

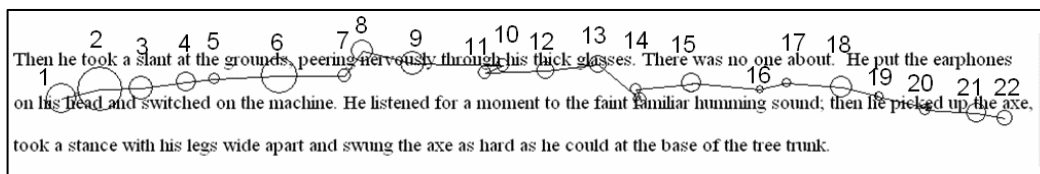


Figure 7.4: Resuming vertical accuracy (EyeLink).

Both of the trackers used during the development of iDict had their strengths and weaknesses in recording the reading paths. In principle, EyeLink supplies the fixation locations with greater accuracy, but fixing the head-mounted optics to the head in a stable enough manner causes problems. Because the eye cameras are attached to the head band, the band easily slips, causing a change in the positioning of the eye cameras in relation to the subject's eyes. Squeezing the band tightly against the subject's head makes wearing the device uncomfortable, and even that does not prevent errors caused by facial expressions that affect the subject's forehead and cause distortion from the original calibration. The fact that the drift may be either permanent or only temporary makes deducing the target words even more difficult to manage.

While vertical inaccuracy during reading of a line is more typical of the tracker with head-mounted optics, it cannot be said to be totally due to head band slippage. Similar paths were recorded also with the remote optics eye tracker (see Figure 7.5, and also Figure 7.1). In Figure 7.5, the context of the reading session allows us to ascertain that the line read with fixations 1-15 was on the third line of the clip.

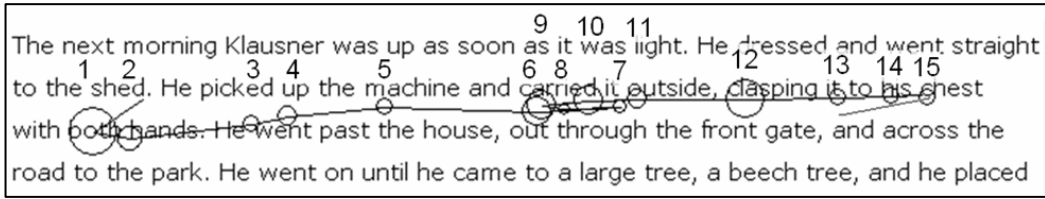


Figure 7.5: Ascending reading path (iView X).

The first fixations in the example reading paths in figures 7.3, 7.4, and 7.5 are accurate enough to enable mapping to the correct line. However, even the first fixations are often judged inaccurately. Figure 7.6 gives such an example: if interpreted in isolation from their context, the fixations would get mapped to the words in the first of the three lines.

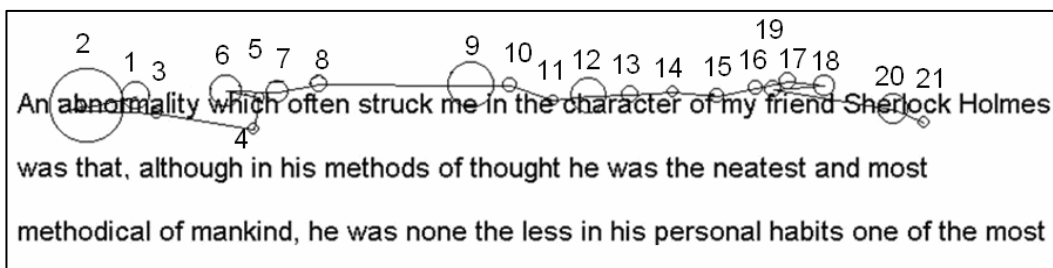


Figure 7.6: Global vertical shift of the whole reading path of a line (EyeLink).

However, a review of the gaze path that was recorded prior to the path above (Figure 7.7a), and of the path that occurred after (Figure 7.7b) the path above, reveals that the fixations in Figure 7.6 were actually targeted at the second line of the clip. Thus, the temporal order of the figures is 7.7a (reading path for the first of the lines), 7.6 (second line), and 7.7b (third line). Fixation 27 in Figure 7.7a is the first fixation in Figure 7.6, and fixation 21 in Figure 7.6 is the first fixation in Figure 7.7b.

These examples show that some kind of algorithmic compensation for the drift from calibration is inevitably needed. The fixations cannot be mapped to the words directly on the basis of the word masks. The application should be aware of the context of a fixation; i.e., the reading process should be monitored so that the facts of normal reading behavior can be used to make decisions to compensate for the errors in the measured fixation position. The inaccuracy compensation algorithms developed for iDict are described in Section 8.3.

7.2.2 Horizontal inaccuracy

Compared to vertical inaccuracy, horizontal inaccuracy is much more difficult to pinpoint from reading paths. As the review of the reading studies demonstrated, a lot of detailed knowledge of the reading process is available. For example, we know the probable landing position in a word, situations in which the reader is likely to make refixations to the

word, the assumable fixation durations in these situations, and the situations when the reader is likely to skip a word (O'Regan, 1981).

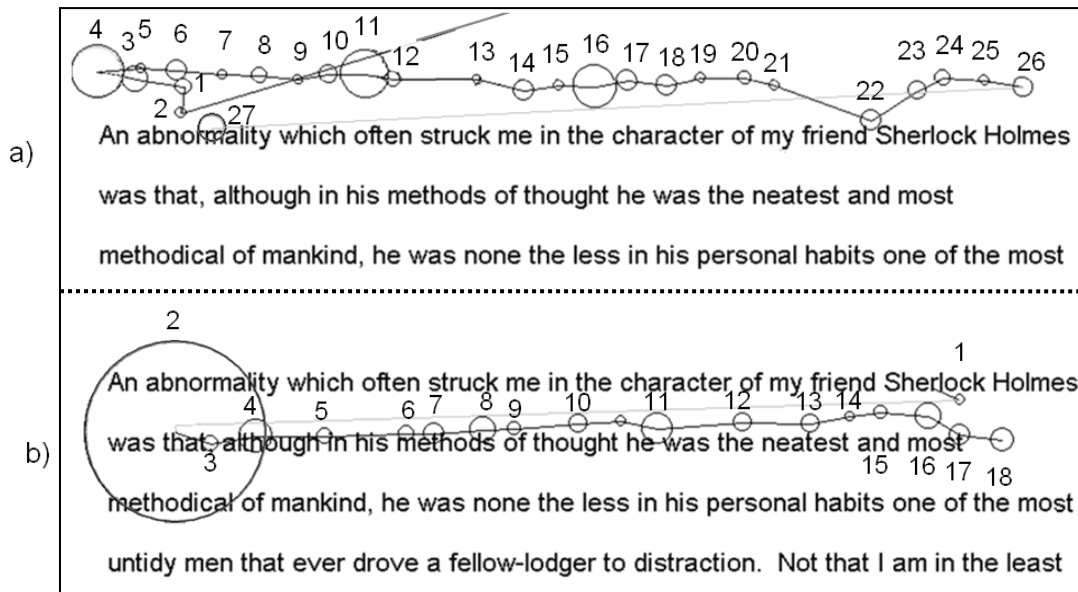


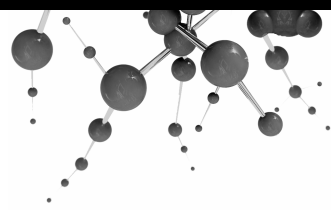
Figure 7.7: Reading paths prior (a) and after (b) the path presented in Figure 7.6 (EyeLink).

Using these findings for dynamic horizontal correction of the drift from calibration proved to be an impossible task. The findings apply to typical reading behavior for the average person. As such, they cannot be used in estimation of error in measurement of single fixation locations. As already seen in the vertical inaccuracy examples, the errors in the coordinates given by the tracker vary from one region of the screen to another. Together with the variance of individual fixations (in relation to the assumed positions of fixations as derived from the average behavior), this variability results in too much uncertainty for making definitive conclusions as to a plausible horizontal drift value.

Fortunately, our experience with a substantial number of tracked sessions showed that horizontal inaccuracy is less common than vertical inaccuracy. Even though there is no conclusive explanation for the phenomenon, it has been reported also by other researchers (e.g., Stampe & Reingold, 1995; Ohno, Mukawa & Yoshikawa, 2002). It may be due to characteristics of human vision or, alternatively (in which case, corrected in time), due to the tracking technology used. Another noteworthy observation from our experiments is that when horizontal error occurs it seems to be global (similar in all parts of the screen), unlike vertical error. Since there is no conclusive evidence that verifies or explains this observation, it may simply be that only the large, global errors were noted.

On the basis of the examples presented, we believe the application should be able to automatically compensate for inaccuracy in real time in order to successfully keep track of the **current text object** in focus in the context of

reading long text passages in a real-world environment. Monitoring the current line of reading is especially important from the vertical inaccuracy standpoint. In the next chapter, the algorithms designed for this purpose are introduced.



8 Keeping Track of the Point of Reading

The algorithms that were developed to compensate for inaccuracy affect the sizes and locations of text object masks dynamically, often resulting in overlapping object masks. That is why the order in which the object masks are considered for mapping an incoming fixation to a text object is significant. In this chapter, we first, before describing the algorithms for handling the inaccuracy problem, introduce the general rules applied in the mapping.

8.1 MAPPING OF FIXATIONS TO TEXT OBJECTS

The general findings on reading behavior (review Section 4.1) that guided the design of the mapping algorithms are the following:

- lines are read from left to right;
- almost every content word is fixated upon at least once;
- about 10–15% of saccades are regressions, often to the same line but sometimes to lines previously read;
- at the end of a line, the reading point is transferred to the beginning of the next line; and
- except in transferring to a new line, saccades to successive lines are almost nonexistent.

The last item in the list is more a general observation from our experiment and holds only when the reader is attentive. Usually, the cases where the reader makes fixations to lines beyond the line being read demonstrate either a loss of concentration in reading or that the reader is just skimming through the text without really reading it. More generally, in keeping track of the reading, the algorithms are optimized to find the target text object efficiently only if the reader attends to reading the text. However, the application should also be able to recognize the focused text objects when the reader scans the text in an atypical manner.

The basic concept of mapping a fixation to a text object was phrased (in Section 6.3) to mean that the fixation coordinates are inside the mask of the object. For now, we can forget the complexity derived from the inaccuracy because it will be taken care of when the masks for text objects are assigned, which will be described in the next sections of this chapter. In this section, the following notation is used in presenting the mapping algorithms.

The information on the point in the text where the reading is progressing is maintained in the **current text objects**:

p_c = current paragraph,

l_c = current line, and

w_c = current word.

In the notation, each index may vary from 1 to x , where x is the maximum index in its context. For example,

$p_c l_c w_x$ stands for the last word in the current line, and

$p_x l_i w_1$ stands for the first word in the i^{th} line of the last paragraph.

Accordingly, for the **freshly focused text object**,

p_f , l_f , and w_f is the lastly mapped text object – the target of the latest fixation.

The freshly focused text objects are separated from the current text objects because a focused text object does not always evoke updating of the current object. In our tests, single separate fixations, like the stray fixation in Figure 8.1 (or fixation 13 in Figure 7.3, and 22 in Figure 7.7a), were common. These stray fixations were targeted more commonly upward than downward from the current line; one explanation is that the readers made some kind of half-conscious checking of the previously read context. That is why the current line and current paragraph are not changed by stray fixations. This also holds for skimming of the text: the change in current word, line, and paragraph are evoked not before a second fixation is mapped to a new line.

8.1 Mapping of fixations to text objects

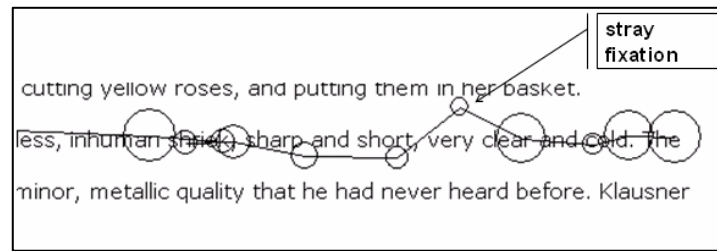


Figure 8.1: A stray fixation (EyeLink).

The notation $p_{prev}l_{prev}$ in the following algorithms maintains the information of the freshly focused line $p_f l_f$.

The position of the fixation received from the tracker is denoted with fix_c . The high-level pseudocode representation of the algorithm that maps the newly retrieved fixation to the corresponding text objects is given below. Both the text object masks for the text being read ($mask$) and the current and freshly focused text objects ($p_c l_c w_c$ and $p_f l_f w_f$) are used globally in these algorithms.

```

Map_fixation ( $fix_c, p_c l_c w_c, p_f l_f w_f$ )
# The procedure updates the current text objects  $p_c, l_c,$  and  $w_c$ 
1 if  $fix_c$  outside  $mask(document\_frame)$  # handle the possible eye-sensitive areas of the
2   handle\_other\_areas; exit # application window, outside the document frame
3 if  $p_c l_c w_c \neq NUL$  and  $fix_c$  inside  $mask(p_c l_c w_c)$ 
4   exit # refixation to the same word:
5   # current text objects not updated
6  $p_{prev} = p_f; l_{prev} = l_f$  # preserve the last successful paragraph
7  $p_f l_f w_f = \mathbf{Map\_word}(fix_c)$  # and line mappings
8 if  $p_f l_f w_f \neq NUL$  # if a new target word was found
9   if  $fix_c$  inside  $mask(p_c l_c)$  # check if the current text object should be updated
10     $w_c = w_f$  # fixation on the same line as the previous one
11 elseif  $fix_c$  inside  $mask(p_{prev} l_{prev})$ 
12     $p_c = p_f; l_c = l_f; w_c = w_f;$  # at least second fixation on a new line
13 else # first fixation on a new line, current text
14   # objects not updated
15 else  $p_f = p_{prev}; l_f = l_{prev};$  # the mapping failed; preserve the latest
16   # successful mapping
end of Map_fixation

```

Thus, each of the incoming fixations is mapped to a word object, and, if the fixation was not a stray fixation, also the current position where the reading is proceeding is updated. The algorithm contains a function *Map_word*, which performs the actual mapping of the fixation to the target text object as follows:

```

text_objects Map_word (fixc)
# The function finds (and returns) the target word (also line and paragraph) of the fixation
1 if pclcwc ≠ NUL and fixc inside mask(pclc)
2   for each wi from wc+1 to wx and from wc-1 downto w1 do
3     if fixc inside mask(pclcwi)
4       return (pc, lc, wi)           # target word was found from the current line
5
6 pflf = Map_line(fixc)           # else find the new target line
7 if pflf ≠ NUL                       # target line was found
8   for each wi from w1 to wx do
9     if fixc inside mask(pflfwi)
10      return (pf, lf, wi)       # target word was found from a new target line
11 return (pf, lf, NUL)           # target word was not found
end of Map_word

```

Hence, the mapping is done on the basis of word masks and working out of the “normal reading behavior” starting from the current word: the procedure first checks whether the fixation is inside the word mask of one of the next words in the current line; then the previous words are checked. If the target word is not found in the current line, the function *Map_line* is called for the changed target line. The function *Map_line*, in turn, after trying to map the fixation in the current paragraph first to the next line, then to previous lines, and last to the lines beyond the next line, asks for the changed target paragraph from the function *Map_paragraph*, if needed. The pseudocode representations of *Map_line* and *Map_paragraph* are analogous to that of the *Map_word* function.

Until now, the only definition for the mask objects has been the rough one given in Section 6.3: “The object mask is originally the smallest rectangle that encloses the object.” However, it was already remarked upon, at the beginning of this chapter, that the object masks are not static but, instead, are dynamically affected by the algorithms that aim to correct the inaccuracy in the tracked point of gaze. These algorithms are described in the next section.

8.2 DYNAMIC CORRECTION OF INACCURACY

Dynamic correction of the drift from calibration was first suggested by Stampe (1993) and by Stampe and Reingold (1995). A similar approach was presented by Hornof and Halverson (2002), who used “implicitly required fixation locations” (RFLs) to correct the systematic error of the measured position of gaze. The principle was that when – during an eye tracking session – we can reliably assume that a user’s visual attention is focused on a certain object, we use the information on the real position of the object and the measured point of the object to determine the probable error of the eye tracker. This information can then be used to correct subsequent eye tracking data.

However, adopting a similar method to correct the drift globally would not be wise, because the errors we detected were often clearly local. An example is shown in Figure 8.2. During the reading of the first lines, vertical drift apparently occurs (especially on the right), but when the last lines of the clip are read, the measured vertical coordinates of the fixations in the line endings are quite accurate. Therefore, we cannot use the observed error of the measured fixation point to globally correct the subsequent fixation locations. The recent WebGazeAnalyzer application, designed by Beymer and Russell (2005), has a similar approach to ours for keeping track of reading. Their application keeps track of horizontal gaze lines and tries to match them to the lines of the text document. However, WebGazeAnalyzer is designed for performing post-analysis, when the whole data set saved during a session is available. In our case, the mapping must be done in real time during reading.

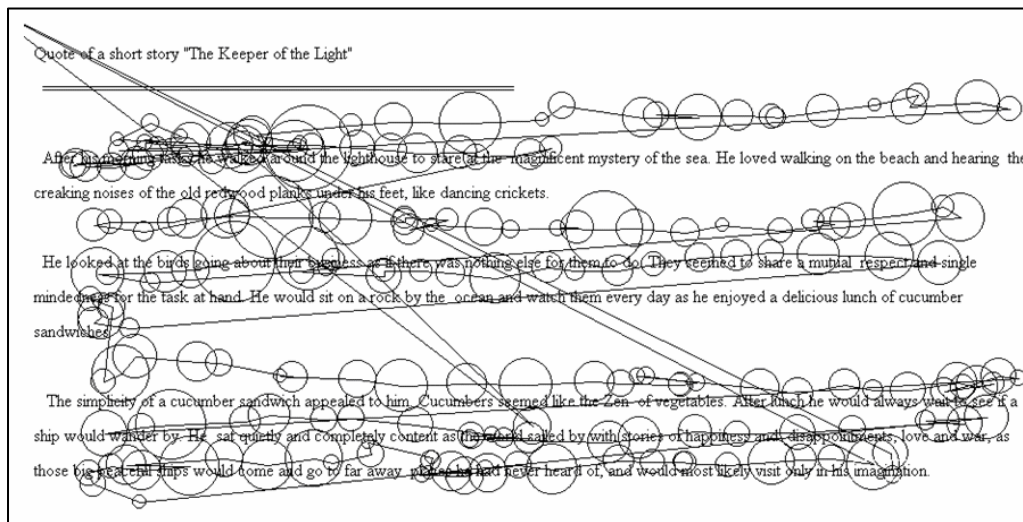


Figure 8.2: Example of local vertical shift (EyeLink).

When iDict diagnoses an error in a measured fixation location, the information is passed to the text objects and the correction is applied locally, only in a region in the proximity of the point where the error was discovered. Requisite for performing dynamic correction is that we, at some points during the tracking sessions, know with a high certainty the correct location of the tracked point of gaze.

In gaze-command applications where gaze is used to actively initiate actions on the screen, these “hot points” are easier to localize. In gaze-aware applications similar to iDict, the situation is more complicated when the user does not intentionally control the eye movements. An eye behavior pattern that can be used for localizing “hot points,” in the context of reading, occurs when a reader moves from one line to another. Such moves are called “return sweeps” in reading studies. Our goal is to detect the return sweeps in order to pass the information to the application as a

new line event. We shall use the two terms interchangeably. The pattern is examined in detail in Section 8.5. First, the principles of the drift compensation algorithms are described below.

From the application's point of view, the source of the error in the measured point of visual attention is not relevant. Essential to consider is that the error is not consistent; it changes over time and over the tracked target space. In other words, the accuracy drifts during the session. That is why we call the algorithms we have developed **drift compensation algorithms**, even though they partially compensate also for the inaccurate measurement of the focus of visual attention.

8.3 DRIFT COMPENSATION ALGORITHMS

The drift compensation algorithms developed for iDict operate on three levels. The first two levels are handled automatically, and the third (for use if the two levels of automatic correction fail) is left to be managed by the user. The three levels of algorithms compensating for the inaccurate tracking of a line can be briefly summarized as follows.

- | | |
|-----------------------------|---|
| 1. Sticky lines | <ul style="list-style-type: none"> - involves automatic vertical correction - affects the height of the current line mask temporarily - compensates for vertical drift across the line being read |
| 2. Magnetic lines | <ul style="list-style-type: none"> - is based on smoothly proceeding reading - involves automatic vertical correction - affects the locations of line masks persistently - compensates for vertical drift at the beginning of a line - is based on return sweeps |
| 3. Manual correction | <ul style="list-style-type: none"> - involves manual vertical and horizontal correction - affects the locations of line and word masks persistently - is based on the feedback given by the user. |

Below, each of the algorithms is described in detail.

8.3.1 Sticky lines - a vertically expanding line mask

The unoccupied space between lines is normally allocated evenly to the masks of the line above and below. However, when the current line of reading is known, the first precaution against vertical inaccuracy is that, while the line is being read, its mask is enlarged to include the full height of the unoccupied space between the lines' original line masks (above and below – see Figure 8.3).

The line masks are expanded horizontally, also. Reading research has shown that the field of perceptual span in reading is asymmetric. A reader is able to use the parafoveal vision more efficiently to the right than to the left of the foveal region of the visual field. According to Rayner (1995), a reader is able to identify as many as 15 letters to the right of the gaze point but only three or four to the left. The asymmetry of the field of perceptual span has been found to be tied to cultural background, as we noted in Section 4.2. Since our application was developed for languages that use the Latin alphabet, the horizontal expansion of line masks is greater on the left than on the right. Overshootings to the left upon movement to a new line are not common (this is discussed further in Section 8.4), so the expansion applied at the beginning of a line is not as great as the size of the perceptual span field suggests. Instead of expansions of 15 character widths at the beginning of a line and four character widths at the end, we use nine and three, respectively.

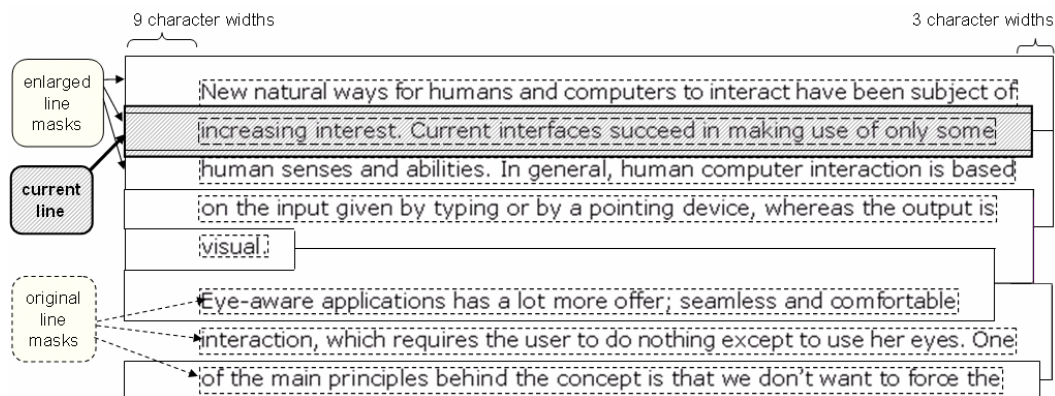


Figure 8.3: Vertically expanded masks and the dominating current line's mask.

The first and last lines of the text are handled as exceptions on the basis of the space above and below. If the space above (or below) the first (or last) line is empty, the line mask can have its height increased safely. In these cases, the upper boundary of the mask of the first line is increased (or the lower boundary of the last line decreased) by two line heights. Unallocated space left by short lines is also allocated evenly to the masks of lines above and below (not more than one line height in either direction, though).

A similar logic is applied for the paragraph masks. The current paragraph dominates in allocation of the free space around it. The mask assigned for the word objects is derived from the asymmetric character of the perceptual span field. The space between words is always joined to the subsequent word.

The other precaution against losing the current line of reading is to further expand the height of the mask of the current line as long as reading is

interpreted to be continuing along the line. Each fixation mapped to the words in the current line increases the height of the line mask. In order not to overlook the shifts in line that the reader does make, the expansion has to be restricted. In iDict, we found an incrementing of the top boundary (and a decrementing of the bottom boundary) of the line mask by about one tenth of the font height to be sufficient. With the trackers we used, a reasonable limit for the maximum expansion in one direction was 1.2 times the line height. The enlarged mask of the current line is transient; when the current line is changed, this (previously current) line reinstates its old mask.

Thus, the enlarging mask of the line currently being read helps with the problem commonly encountered with ascending and descending reading paths. For example, the wrongly mapped fixations (from 10 to 15) of Figure 7.5 now get mapped to the third line of the clip as they should be, as shown in Figure 8.4. The gray area denotes the growing height of the line mask as a result of each new fixation in the current line.

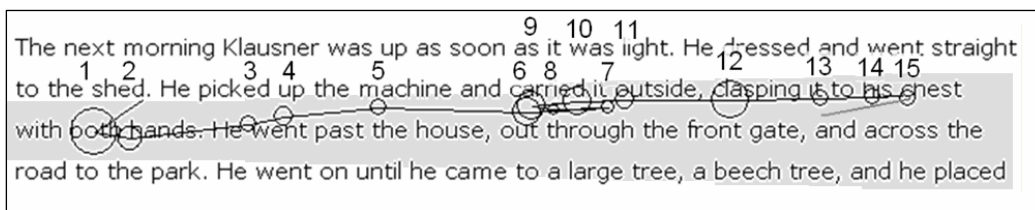


Figure 8.4: Constantly expanding mask of the current line.

Sticky lines compensate for the drift in accuracy if only two presumptions are fulfilled: the first fixations focused on a line are mapped to the correct line, and the occasions when the smooth progress of reading of a line is disrupted are identified. Before considering how to handle the first of these presumptions, we describe the two modes we use for identifying disrupted flow of reading: **Scan Mode** and **Smooth Progress Mode**.

Scan Mode – detecting the behavior of scanning

Dramatic discontinuations of fluent reading are taken care of with the mapping algorithms given above, in Section 8.1. The algorithms are optimized to find the word in focus in cases where reading proceeds typically. Nonetheless, the word in focus is found also in the case of atypical reading behavior, even if not so effectively.

However, since the mask of the current line has some degree of precedence over the masks of other lines, the temporary mask enlargements for the current line must be cancelled when the reader seems to cease smooth reading and begins to scan through the text. We found that the change from intensive reading to scanning was easy to detect from the reading paths on grounds of (vertical) saccade height. In reading,

horizontally long saccades are common, but large vertical transitions indicate that the reader is making stray fixations, not attending to reading.

In inspection of the reading paths, it seemed to be safe to set reading detection to Scan Mode when the height of a vertical saccade is over three line heights. When Scan Mode is active, the temporary changes made in the mask of the current line are cancelled. Then, none of the lines dominates in allocation of mask space, since there are no longer expectations for any particular line to be in focus.

Smooth Progress Mode - preventing too sticky lines

The growing line masks may cause insensitivity to when the reader changes line, as in the quite common pattern of regressing to a previous line. If the line mask has grown to cover the space of the previous line's mask, the regression goes unnoticed. That is why the algorithms must keep track of divergent reading behavior. Similarly to Scan Mode, Smooth Progress Mode is set to be on when the reading appears to be progressing smoothly along the same line. Setting this mode to be inactive is a less radical move than setting the algorithms to Scan Mode. Reading may still be continuing normally. When Smooth Progress Mode is turned off, the spread line mask of the current line shrinks to its original size, still occupying the neighboring white space.

We studied various reading paths in order to determine when to assume that reading is progressing smoothly along the line. The following simple condition turned out to work adequately. A saccade height of less than a font height alone is enough to keep the algorithms applying Smooth Progress Mode. However, we noted earlier that the reading paths include short stray fixations, in which the vertical transition often exceeds the font height. That is why the reading is considered to progress smoothly even if saccade height exceeds the font height - if the saccade length is shorter than 12 characters and the absolute value of the saccade's angle is less than 0.3 radians (i.e., $\text{abs}(y_shift) / \text{abs}(x_shift) < 0.3$).

8.3.2 Magnetic lines - relocation of line masks

The first of the presumptions for sticky lines is that the first fixations are mapped to the right line. Obviously, this assumption is not realistic without further consideration of the inaccurately measured point of visual attention. Vertical inaccuracy may occur also at the beginning of the line, as the examples in figures 7.6 and 7.7 demonstrate.

New line events are used to perform automatic vertical relocation of the newly entered line's mask. We will concentrate on an examination of new line events in the next sections (sections 8.4 and 8.5), showing that in transfer to a new line, the long regressive saccade often causes inaccuracy in the reader's first fixation at the beginning of a new line, both vertically and horizontally. This is why relocation of the mask of a newly entered line is not performed on the basis of the first fixations alone. The vertical relocation of the line mask is updated according to the average of the vertical coordinates of the first three fixations on the line. Actually, the new vertical position of the line mask is computed using only two fixations; the first fixation, for entering the new line, is excluded. The relocation of the line is persistent: if the reader makes regressions to previously read lines, the mapping of words is performed using the relocated line masks.

It was previously concluded that the inaccuracy not only is local to some regions of the tracked space but can also change during a reading session. The "magnetic lines" algorithm takes inaccuracy into account cautiously by spreading the observed inaccuracy only for those line masks in close proximity to the current line. When an offset of dy pixels between the measured fixation positions and the location of the current line is observed, also the masks of the neighboring lines (both above and below, four lines in total) are relocated. The immediate neighbors' masks are shifted by $dy/2$ and the next line masks beyond the neighbors by $dy/4$ pixels. If the updated line masks overlap with the original (possibly reoriented) mask of the neighboring line mask, the relocation may have an effect beyond the two neighboring lines in each direction. The relocation should not lead to a situation in which a mask of the original line mask overlaps with the neighboring line mask. In these cases, the line space area (the space between lines) is used to temper the push effect of mask relocation.

The cautious spreading of the repositioning of line masks takes into account that the observed inaccuracy does not necessarily globally affect the interpretation of measured fixation coordinates. However, even if the error were global, the correction would be carried further by each of the subsequent line repositionings.

8.3.3 Manual correction

If the automatic algorithms fail to compensate for the drift of the measured fixations' locations, the reader may perform an explicit drift correction using the arrow keys. This implies that the user knows which line and word are considered to be the current ones (by getting proper feedback from the application). The issue is discussed in the context of interaction design in Chapter 10.

For example, in Figure 8.5, the reader has finished the reading of the first line with fixation 2. When the reader is searching for the beginning of the next line, the word “society” is fixated upon on the first line before reading of the new line commences with fixation 4.

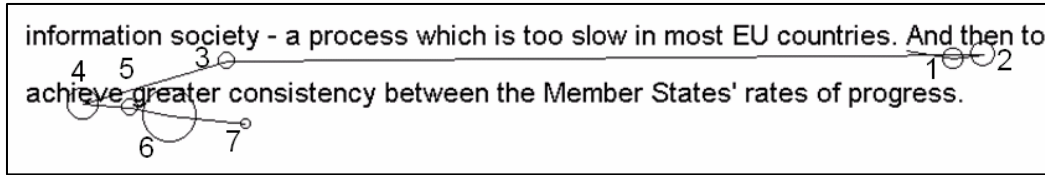


Figure 8.5: An example of a return sweep (iView X).

Regardless of the extensive amount of reading research performed, there seem to be no studies concentrating on examining on a detailed level how the transition from one line to another happens. The reason for that is understandable. Since most reading research is performed in the field of psychology, the emphasis has been either on understanding the underlying oculomotor processes or on using eye movements (together with linguistics and psycholinguistics) for revealing the reader’s cognitive processes during reading. Transferring from line to line is not very interesting in either of these contexts. On the contrary, it usually distracts from the processes being studied, as a confounding factor. The vast majority of the empirical reading studies in which the eye movements are recorded present the stimulus text on one line. The research usually concentrates on local phenomena: eye behavior while one is reading a few consecutive words or a single sentence. So, in most cases one line is enough. Presenting a one-line stimulus also conveniently avoids the problems with vertical accuracy.

Even if no comprehensive studies have been conducted in the area of interest to us, observations of reading paths in the transition to a new line have been reported by many researchers. These observations are similar to those we made in our own experiments. For example, Just and Carpenter (1980) refer to a study performed by Bayle in 1942, in which it was observed that

[...] the return sweep is typically too short: the eye often lands on the second word of the new line for brief amount of time and then makes a corrective saccade leftward to the first word in line.

Similar undershooting with large saccades is observed in other visual tasks also (O’Regan, 1990). Siebert et al. (2000) make a note that readers sometimes skip a line by mistake. Rayner (1998) reports that the first and last fixations often fall at a distance of five to seven letters from the ends of the line. The first fixation on a line has been observed to be longer (Rayner, 1977) and the last fixation shorter than an average fixation (Rayner, 1978).

8.5 ANALYSIS OF NEW LINE EVENT GAZE PATTERNS

To better understand when to launch the new line event, we analyzed the eye movement data from a reading test. We wanted to find out (1) how many saccades the readers make when they transfer their focus from line to line, (2) the length of the saccades made when moving to a new line, and (3) the positions of the first and last fixations on a line.

In the test, 10 participants read three blocks of text, each of which had 10 lines. Five of the participants were male, and five were female. The participants had an average age of 24 years. EyeLink was the tracker used. The font used to display the double-line-spaced text on a 19" screen (with a resolution of 1024 x 768) was 12-point Times New Roman. A complete description of the test setup is given in Section 9.1. The test was originally set up for determining the triggering threshold for providing a gloss. However, since the experiment setup involved reading without any interruptions, the data are also valid for the new line event analysis. Gaze data from one of the reading sessions in a longer test was analyzed.

Before analyzing new line event patterns, we must introduce four more concepts:

1. The **first NLE fixation** is the last progressive fixation before reading starts to transition to the next line (e.g., fixation 2 in Figure 8.5).
2. The **last NLE fixation** is the last regressive fixation that ends transition and starts the reading of the next line (e.g., fixation 4 in Figure 8.5).
3. **Transition fixations** are the fixations between the first and last NLE fixations (e.g., fixation 3 in Figure 8.5).
4. **Transition saccades** are the saccades needed to take the eyes from the first NLE fixation to the last NLE fixation on the next line (e.g., saccades 2-3 and 3-4 in Figure 8.5)..

8.5.1 Identification of new line events

Return sweeps were manually winnowed out of the eye movement data. The gaze patterns turned out to be relatively easy to identify visually from the reading paths. An elementary requirement applied was that the point of gaze had to be transferred from one line to the next line in the text. Thus, a gaze pattern where the reader made a regression to a previous line was not regarded as a new line event. An example of a regression to a previous line is shown in Figure 8.6. The beginning of the second line is read with fixations 1-8. The reader then regresses to the first line (fixations 9-13) before returning to the second line.

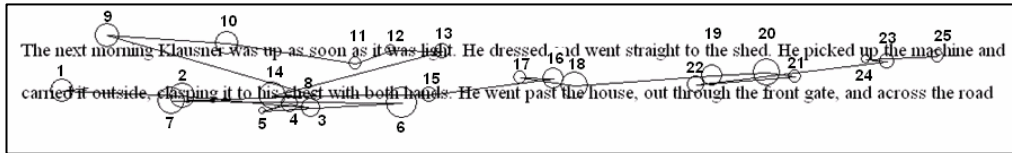


Figure 8.6: Returning to read a line after a short regression to the previous line (EyeLink).

In manual screening, an additional requirement was imposed for new line events: the transfer should leave from the end (last quarter) of the current line and end at the beginning (first quarter) of a new line. Consequently, saccade 13–14 in Figure 8.6 does not initiate a new line event. Similar gaze patterns, where the reader returns to check something from the previous line, are common, even though this path contains more saw-edged regressions than are typical.

These rules led to identification of 100 instances¹ of new line events from the data. Since the text passage contained 10 lines of text, if the 10 test participants had read the lines with the minimal number of new line events, there should have been 10×9 transfers from a line to a new line. Two of the readers left the last line unread, though (the last line was comprised only of two words).

The 12 “extra” events were generated when many of the readers, after transferring to the target line, returned to read the end of the previous line. An example of such case we will see later, in Figure 8.12. These patterns – let us call them **reinforced** new line events – may reflect the reader’s intention to maintain the continuity of the sentence after it was broken by the line break. Returning back to read the previous line again may also be an implication of a review of a reference or of a revision of a misconception.

Six events of this type were encountered with one of the readers, who had (according to his own, subjective rating) weaker skills in reading English than other test participants did. So, it may be that the reinforced new line event pattern is more common to weaker readers. However, such events were identified with other readers, too (see Table 8.1), so we have to take them into account regardless of the reader’s skills in reading the text. One of the other participants had two reinforced new line events, and four of them had one. Three of the participants read the lines with the optimal number of new line events.

¹ Coincidentally, the number of new line events gives the observer a convenient opportunity to project the subsequent references as numbers of new line events directly in the form of percentages.

NL events	Participant										total
	1	2	3	4	5	6	7	8	9	10	
normal	9	9	9	8	9	9	9	9	9	8	88
reinforced	1	0	6	0	1	1	1	0	0	2	12
total	10	9	15	8	10	10	10	9	9	10	100

Table 8.1: The number of different new line events identified.

8.5.2 Number of transition saccades in new line events

The readers used up to four transition saccades in moving from one line to another (see Figure 8.7). The most typical new line event contains two transition saccades; i.e., the readers transferred to the next line using one transition fixation. Three-transition-saccade new line events were common, too, but one- and four-transition-saccade events were both quite rare.

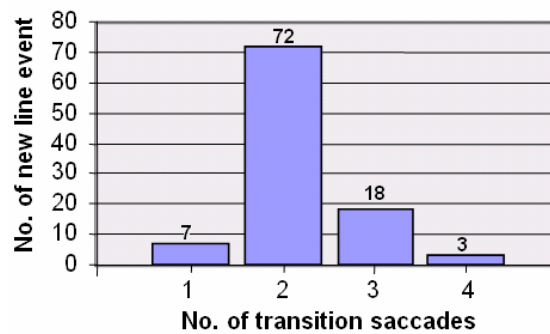


Figure 8.7: Number of transition saccades in new line events. Numbers of 1-, 2-, 3-, and 4-saccade transitions among the 100 new line events analyzed.

Figure 8.8 illustrates the distribution of one-, two-, three-, and four-transition-saccade new line events by subject. It reveals that two subjects had four-transition-saccade new line events and there were one-transition-saccade new line events for half of the subjects. If rare new line event types are seen with only some of the readers, this could indicate that readers have individual styles in transferring to a new line.

The fact that the subjects who made four-transition-saccade transfers did not have rapid one-transition-saccade transfers at all indeed suggests that some readers tend to make transfers more slowly, using more transition saccades than others do. Nonetheless, the differences appear to be very small. No new line events with five or more transition saccades were found – not even from the reading path of the reader (P3) who had poorer skills in English and read the text much more slowly, using considerably more fixations to read the text than others did.

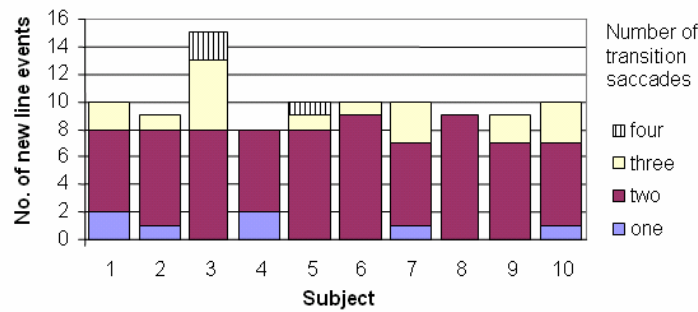


Figure 8.8: Number of transition saccades in new line events by participant. Individual numbers of 1-, 2-, 3, and 4-saccade transitions among the analyzed 100 new line events.

The first observation that contributes to the function being designed to detect new line events in real time is that four is the maximum number of transition saccades we have to watch out for. In addition, we need to know how long the transition saccades are.

8.5.3 Transition saccade length

In the discussion that follows, “**transition length**” denotes the horizontal transition during a new line event pattern: the summative length of the horizontal transition from first NLE fixation to last NLE fixation. The unit used to measure the transition length is the relative length of the launch line. For example, a transition length of 0.9 represents horizontal saccades that transfer the reader’s focus by 90% of the launch line’s length.

For 99 of the 100 new line events studied, the transition length was more than 0.8 times the launch line’s length. Thus, in identification of new line events, the limit can be safely set to 0.8 times the launch line’s length. Figure 8.9 shows more precisely how the total transition length is summed from the lengths of each of the transition saccades¹.

In most new line events, the first transition saccade is the main saccade and takes the reading from the end of the launch line to the near vicinity of the beginning of the target line on the left. Inspection of new line event patterns confirms the earlier observations on corrective saccades. Mostly, the short second and third transition saccades correct undershooting when the reader searches for the beginning of the target line.

However, there is also a smaller number of new line events in which the transitions do not follow this pattern. In some cases, the reader makes a short precursory regressive saccade before establishing longer saccades to the left. Also, there are a few cases in which the first transition saccade

¹ There were only three new line events that consisted of four transition saccades. All of these fourth transition saccades were very short, less than 0.05 times the launch line’s length.

8.5 Analysis of new line event gaze patterns

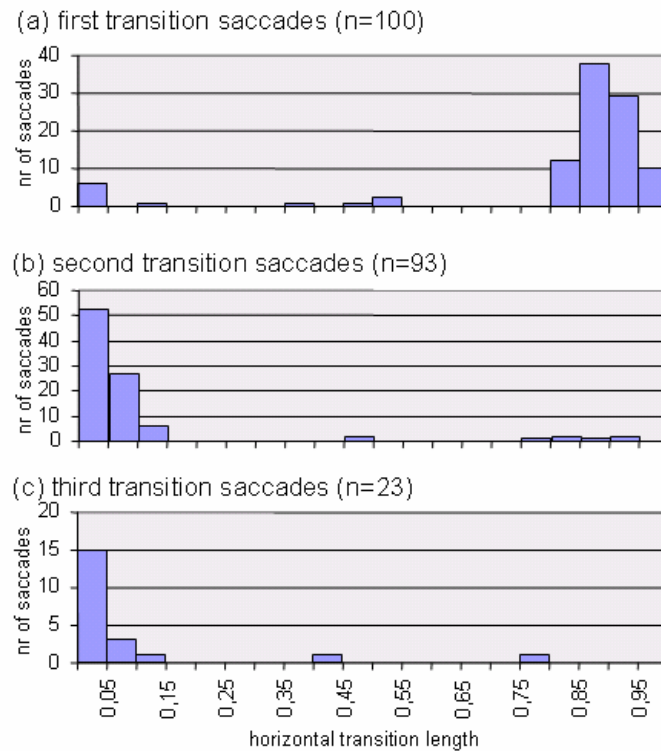


Figure 8.9: Distribution of transition saccade lengths (expressed in relation to the launch line's length).

lands horizontally somewhere in the middle of the launch line.

In most cases (89 out of 100), the first transition saccade alone was horizontally more than the limit of 0.8 times the launch line's length, and in almost all (99 out of 100) new line events the length of three successive regressive saccades exceeded the limit. This suggests that even performing a quick and dirty check of the single saccade lengths for regressive saccades ends up screening most of the new line events effectively. However, using the three last saccades sharpens the accuracy at a low cost. Summing the three last saccade lengths and comparing the result to the limit value is a fast operation, provided that the data structure maintaining the list of fixations allows rapid access to previous fixations and saccade lengths.

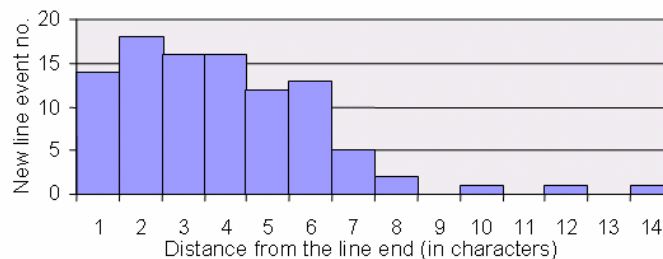
Thus, when a regressive fixation is encountered, the algorithm should check the cumulative saccade lengths for up to three previous saccades in order to alert the application to a new line event. Identifying new line events on the basis of transition saccade lengths alone easily results in accepting incorrect gaze patterns. Locations of first and last NLE fixations are additional indicators that we can use to make the identification more accurate. They can be used to screen out erroneously identified patterns.

8.5.4 First and last NLE fixation locations

Figure 8.10 shows horizontal distances of last and first NLE fixations from the line ends. The distance is expressed in characters. Using the mean character width of the text in the line as the measuring unit for the fixation landing positions renders the analysis independent of the font used in displaying the text.

The distances of first NLE fixations from the line endings in 99¹ new line events are displayed in Figure 8.10a. The average distance was four characters, with the standard deviation 4.4. The average distance from line beginnings of last NLE fixations (Figure 8.10b) in new line events was also four characters. A lower standard deviation value, 2.4, indicates that last NLE fixations are a little more steadily focused at the beginning of the new line than are first NLE fixations at the line endings. We chose to take only the last NLE fixation distances under control in the new line event detection algorithm. An area 12 characters wide was set as the bounds for the target of last NLE fixations; i.e., a new line event should end with a fixation to the first 12 characters on the target line. Also a fixation to the left of the beginning of the target line can be the last NLE fixation.

(a) launching positions of first NLE fixations on launch line (n=99)



(b) target positions of last NLE fixations on target line (n=100)

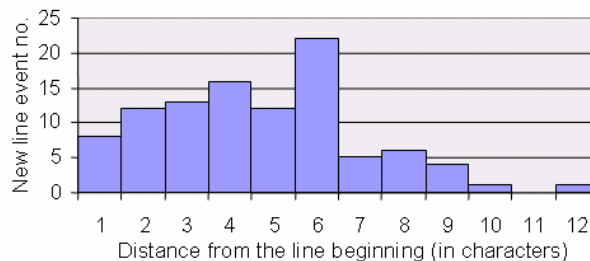


Figure 8.10: First and last NLE fixation locations.

(a) Distance of first NLE fixations from the line endings.

(b) Distance of last NLE fixations from the line beginning (expressed in characters).

¹ In one new line event, the distance from the line's end was 38 characters.

Fortunately, Smooth Progress Mode already handles these gaze patterns. A long saccade to the end of the previous line probably turns Smooth Progress Mode off, and hence the line mapping algorithm when performed without expanded line masks does find the right line for the fixation. Thus, we just have to keep guard for smooth progress also during repositioning the new line, and the magnetic lines process must be interrupted when the smooth progress is discontinued.

Nevertheless, the requirements for saccade height during the repositioning should be looser, because of the corrective saccades in the beginning of the line. Instead of one font height and 0.3 radians for horizontally short saccades, the limits for saccade height and angle limits (for horizontally short saccades) are set to the target line's line height and 0.5 radians during the use of magnetic lines.

8.6 NEW LINE DETECTION ALGORITHM

The above analysis resulted in the following function (next page), which is able to detect new line events in real time. A summary of the parameters derived above is provided at the beginning of the function. The *id* (ordinal number) of the last fixation retrieved from the tracker is passed to the function in the call.

The function is called whenever the eye tracker sends a regressive fixation that is focused on the beginning of a line (left of the observed limit: 12 x the mean character width).

boolean NewLineEvent (*id*)

parameters used to detect new line events in real time

```

1  sacc_cnt = 3;           # number of transition saccades to be inspected
2  min_h_shift = 0.8; # length of the transition during NLE (relative to launch line length)
3  min_v_shift = 0.2; # minimum and maximum heights of the transition...
4  max_v_shift = 2.2; # ... during NLE (relative to line height)
5
6  h_shift = v_shift = 0
7  set line_2 to the line on which fixation(id) was targeted
8  set line_1 to the line preceding line_2
9  if line_1 = NUL or line_2 = NUL
10     return false
11
12 for each tmp_fixation from fixation(id) downto fixation(id - sacc_cnt - 1) do
13     if tmp_fixation = NUL return false
14
15     set first_nle_fixation to the fixation prior to tmp_fixation
16     if first_nle_fixation = NUL return false
17
18     add horizontal saccade length (first_nle_fixation - tmp_fixation) to h_shift

```

8.6 New line detection algorithm

```
19   add vertical saccade height (first_nle_fixation – tmp_fixation) to v_shift
20   if (h_shift > min_h_shift * length of line_1 and
21       v_shift > min_v_shift * height of line_1 and
22       v_shift < max_v_shift * height of line_1)
23       return true
24 return false
end of NewLineEvent
```

We now have described how a fixation position acquired from the eye tracker is mapped to the focused word. Due to the inaccuracy of eye trackers (which is evidenced in inaccurate fixation positions), the search for the focused word is performed on two levels: (1) the fixations are mapped to “floating” text objects on the basis of their temporal mask locations, and on the other level (2) the text object mask repositioning in the application window is determined on the basis of the user’s reading behavior. On both levels, the algorithms’ design has its origin in knowledge of typical eye behavior during reading. However, from the feasibility standpoint, it is essential that the algorithms be able to handle atypical reading paths, also. We cannot assume that a reader is concentrating fully on reading the text without breaks, or that he or she is always motivated enough to read a text line by line.

8.7 COPING WITH ATYPICAL READING PATTERNS

It is interesting to note that originally, in the idea paper for iDict (Hyrskykari et al., 2000), we figured that, in addition to identifying when the reader is encountering difficulties, we would have to develop algorithms that are able to distinguish among the reader’s three states: “scanning, reading, and dormant gazing,” as we put it. We soon renounced the idea because it seemed unnecessary for the application to know these states in the context of determining when to give translations to the reader. Now, two of the states are recognized after all (in Scan Mode and Smooth Progress Mode), but more to cope with the inaccuracy than to recognize the need for automatic help. The state of “dormant gazing” (when the user stares at the screen with “blank eyes”) is still left unidentified. This state is very difficult to separate from that in which the reader concentrates on processing a problematic word. In the context of iDict, waking up the user with possibly unnecessary glosses is not necessarily annoying.

Considering the following two examples of atypical reading paths should be an aid in following how the algorithms presented in this chapter work. The first path is the same as presented in Figure 8.6, but is repeated here for ease of reference.

8.7.1 Examples of following atypical reading paths

In Figure 8.11, the post-analysis of the reading path makes it apparent that, after reading the second of the lines, the reader makes a regressive fixation (fixation 9) to the first line. Reading of the second line is recovered with fixation 14.

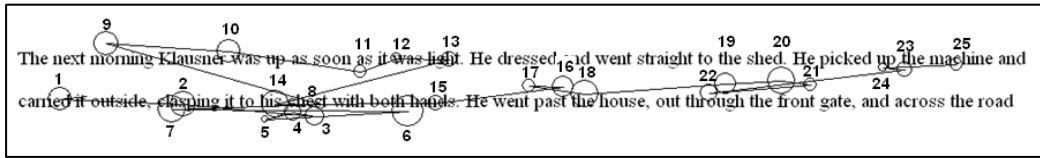


Figure 8.11: Regressive fixations to the previous line (EyeLink).

Our algorithms stay in Smooth Progress Mode during the first eight fixations. During these fixations, the line mask of the second line spreads by 8/10 of the line's height, but the saccade between fixations 8 and 9 breaks the smooth reading (vertical shift of a long saccade by more than the font height).

At this point, the line mask of the current line regains its original size. Thereby fixations 9 to 13 are mapped to the first of the lines. During those fixations, the line mask of the first line spreads by 5/10 of the line height. Saccade 13-14 again breaks the smoothly progressing reading of the first line and shrinks the spread mask, resulting in mapping of fixation 14 to the second line. Fixations 14-25 all keep the algorithm in Smooth Progress Mode and expand the line mask of the second line so that fixations 23, 24, and 25 are mapped to the second line even though their position is inside the first line's original mask.

Figure 8.12 is an example of a very complicated reading path. In this case, the regression to the previous line is followed by an additional new line event. In fact, in this case it is hard to map some of the fixations to the correct line with absolute certainty. However, the wider context of reading assures that reading of the second line was started with fixation 1 and – after regression to review the first line – reading of the second line, from the beginning again, resumed with fixation 17. Also, since the fixations on the second line seem to be measured too high throughout the line, the regression to the first line is not made at fixation 6 but by fixation 8.

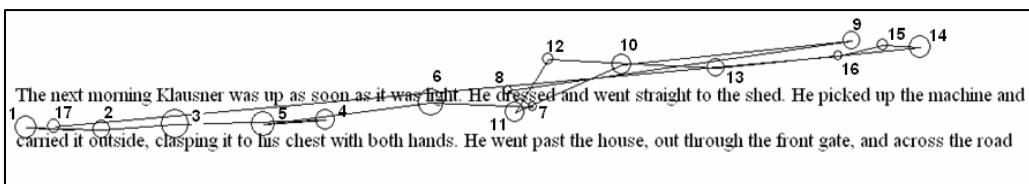


Figure 8.12: Regression to previous line followed by new line event (EyeLink).

Fixations 2 and 3 pull the line's mask (magnetic lines) by about three quarters of the font height and also the first line's mask with half of that correction. That is why fixations 1–7 are mapped to the second line. This is the case even if the vertical transition from fixation 5 to fixation 6 exceeds the current line's font height and turns Smooth Progress Mode off, reducing the expanded line mask height achieved during the five fixations. The sharp (about -0.7 radians) regression from fixation 7 to fixation 8 resets the expanded line mask again, resulting in fixation 8 getting mapped to the first line. Fixation 9 is interpreted as a stray fixation, and the first line remains the current line during fixations 10–16. Fixation 17 raises a new line event alert, and the ensuing reading is mapped to the second line, just as it should be.

8.8 PERFORMANCE EVALUATION FOR DRIFT COMPENSATION ALGORITHMS

We set up an experiment to test the performance of the mapping and drift compensation algorithms in practice.

Since the drift compensation relies heavily on line tracing, we – in addition to general performance – were interested in how the line spacing would affect the performance. It is reasonable to presume that the benefits of dynamic drift compensation vanish when the line spacing grows high enough, to the point where the line height exceeds the average inaccuracy of the eye tracking.

8.8.1 Test setup

Six participants (four male and two female) read three text documents, each of which contained about 250 words and 18–20 lines. All of the subjects had good or very good skills in English. The texts were displayed on a 19" screen with a resolution of 1024 x 768. The font used in each of the texts was 11-point Verdana. One of the texts was displayed with single line spacing, one with 1.5 line spacing, and one with double line spacing. The line spacing for the texts was counterbalanced to eliminate the possible effects of text contents. To give an impression of the visual layout of the texts used, we display the first part of the text in each line spacing condition in Figure 8.13.

Even though the experiment was conducted using iDict, the feedback and tracking of reading were turned off. Hence, iDict did not interfere with the reading at all; it was used only as an instrument for recording the eye movement data. iView X was the tracker used in the experiment. The tracker was calibrated before each of the three reading sessions. Eye movement information was recorded from each of the reading sessions. Since bad calibration would have affected the results, each of the reading sessions (i.e., three for each subject) was followed by a calibration control

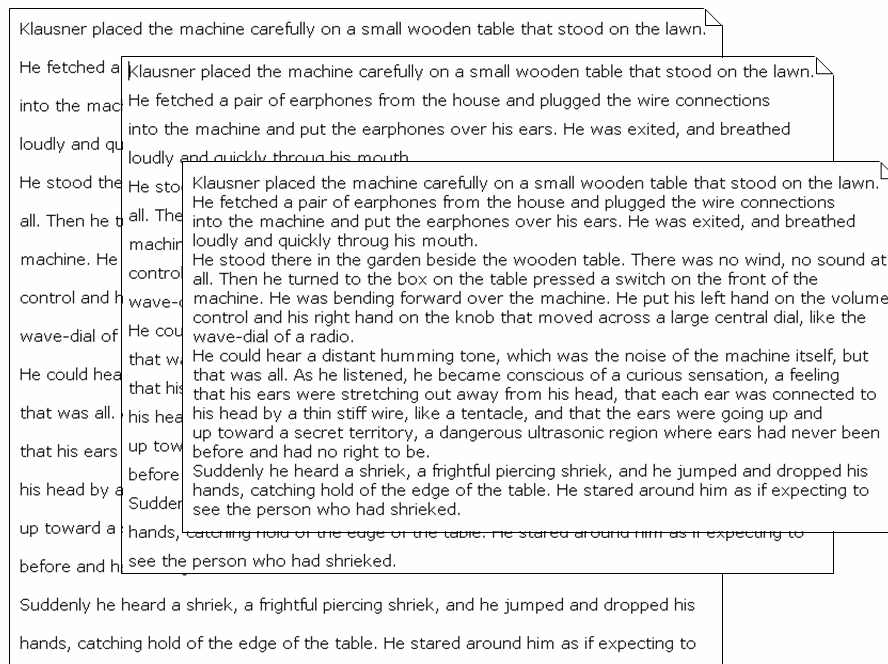


Figure 8.13: The first of the three texts displayed with single, 1.5, and double line spacing.

and recalibration was performed when needed.

8.8.2 Analysis of the reading paths

The total number of fixations recorded in the reading experiment for single-, one-and-a-half-, and double-spaced texts was 1,380, 1,470, and 1,484 fixations, respectively.

First, the fixations recorded during the reading sessions were algorithmically mapped to the closest words, and then the drift correction algorithms were used for obtaining the mapping that iDict would have produced. After this, the correct lines for each fixation were manually determined. The examples in given above in this chapter demonstrate that the right line of a fixation can be determined with a high reliability when the context, the history, and the future of the reading path are known. However, finding the correct mappings manually is a laborious task, and that was why the number of participants in the experiment had to be restricted to six. As a result of knowing the correct mappings, we were able to obtain a “hit percentage,” the percentage of the fixations that were mapped onto the right line with and without the drift correction algorithms.

8.8.3 Results

The hit percentages for each test participant and line spacing condition are presented in Table 8.2.

Each line spacing condition resulted in a better average hit percentage with the correction algorithms than without them. For the single-spaced text, only 39% of the fixations were correctly mapped when the algorithms were not used, and the hit percentage rose to 53% when the algorithms were applied in the mapping. The corresponding hit percentages for the texts with 1.5 and double line spacing were 56% rising to 86% and 76% rising to 78%, respectively.

The improvement was statistically significant for text read with 1.5 line spacing: ($F(1.5) = 9.2$, $p < 0.05$). However, inferences made on the basis of statistical analysis performed for data from an experiment with only six participants are highly unreliable. Thus, the results are considered below with this in mind.

Overall, the results were satisfactory. The algorithms performed best for the most commonly used line spacing (1.5). For three of the participants (3, 4, and 6), the algorithms performed almost perfectly in the case of 1.5 line spacing: for each of the readers, only isolated fixations were mapped onto a wrong line. Achieving a 100% hit percentage is unrealistic in any case. Even though the fixations can in most cases be manually mapped reliably to the right line, the example path given in Figure 8.12 shows that in some

	line spacing 1.0		line spacing 1.5		line spacing 2.0	
	algorithms		algorithms		algorithms	
	not used	used	not used	used	not used	used
1	51%	58%	73%	91%	95%	82%
2	38%	42%	80%	83%	72%	75%
3	30%	40%	54%	98%	84%	98%
4	40%	50%	31%	98%	45%	71%
5	63%	74%	41%	50%	75%	45%
6	9%	56%	57%	97%	83%	97%
	39%	53%	56%	86%	76%	78%
	Average					

Table 8.2: Performance of the drift correction algorithms. Fixation hit percentages for six readers with three differently line-spaced texts, with and without using the mapping and drift correction

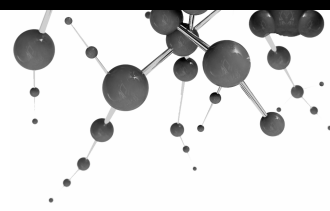
cases deciding upon the right target for single fixations involves uncertainty.

It should also be remembered that iDict was not active during the experiments. Manual correction by a user would have raised many of the individual hit percentages dramatically, due to the fact that once the algorithm has a wrong start for reading a line it often stays on the wrong course for a longer time. An extreme example of this can be seen in the

reading session of the fifth reader in which double line spacing was used¹. Correspondingly, the algorithms would in many cases yield substantially higher hit rates with even a single corrective action on the part of the user. It is interesting to see that, on average, even without manual correction, the algorithms performed reasonably and the hit percentage rose considerably in most cases.

The surprising fact that the correction algorithms resulted in a better average outcome with 1.5 line spacing than with the double-spaced text can be explained mainly by the results of the tracking session of the fifth reader with double-spaced text. Still, the overall improvement achieved with the algorithms was less with the single- and double-spaced text than in the one-and-a-half-spacing condition. Likely explanations for this can be given. Tracking the reading of a double-spaced text (with 11-point font size) seems to hit the limits of the eye trackers' accuracy. If the vertical error in measuring the fixation locations is not more than the line height, the effect of the drift correction algorithms vanishes and can, in fact, be counterproductive, if the algorithm tries to main an incorrect interpretation of the current line while a non-intelligent algorithm would do a better job. In the case of single-spaced text, the algorithms were able to improve the mapping accuracy, but not as much as when the text was 1.5 times line-spaced. A closer look at the reading paths suggests that the reading paths for the single-spaced texts were less smooth. That is, there were occasions when the reader had trouble staying on the right line. The algorithms are designed to take into account the normal regressions that a reader often performs in order to check words already read, and the algorithms failed more readily in tracing the line of reading because the reader lost track of the line being read, resulting in an exceptionally high number of regressions. This assumption is congruent with the spontaneous comments of some of the participants after the test. They stated that single-spaced text was hard to read.

¹ *The sixth reader with single line spacing is an example to the other direction: when the algorithm had a correct line in the beginning, it was able correct the mapping for a longer time.*



9 Recognizing Reading Comprehension Difficulties

One of the original goals of iDict was to provide the user with help as automatically as possible. The gaze paths in which the user expects to get a whole dictionary entry for a word in the dictionary frame are easy to detect simply on the basis of the fixation locations. The critical question is how to decide when the reader seems to be in need of a gloss for a word (or a phrase) in the text frame. We should be able to detect the situations in which the reader seems to have difficulties comprehending the text being read.

We will first supply exact definitions for the eye behavior measures that we found (in our review of the reading research, summarized in Subsection 4.2.3) to indicate comprehension difficulties (sections 9.1 and 9.2). We report on an experiment that was performed to find the measures most suitable for automatically triggering help for a reader in iDict (Section 9.3). The data recorded in the experiment are used for constructing a threshold function for triggering the help function (Section 9.4). The chapter concludes with testing of this threshold function in the context of iDict (Section 9.5).

9.1 READING COMPREHENSION AND EYE MOVEMENT MEASURES

On the basis of previous research, we found five items whose measurement could allow us to detect when a reader is having difficulties understanding the text. The measures we found worthy of closer examination are (1) **first fixation duration**, (2) **gaze duration**, (3) **total time**, (4) **number of fixations**, and (5) **regressions**.



We will analyze each of these in detail below. Each assigns a score for the words in the text as calculated from the stream of fixations recorded during reading.

9.1.1 Definitions for the measures

Consider a text as a sequence of n words $T = \langle w_1, w_2, \dots, w_n \rangle$. Each element w_i in the list T represents an instance of a word in the text. Thus, the words w_i and w_k may represent lexically the same word even if $i \neq k$.

Suppose now that m consecutive fixations are recorded during reading of the text T . These fixations form a sequence $F = \langle f_1, f_2, \dots, f_m \rangle$. For each fixation f_k , $k = 1, 2, \dots, m$, we denote by $w(k)$ the index of the corresponding word instance in T , and by $d(k)$ the duration of the fixation (in milliseconds).

For example, $w(k) = 5$ if the k^{th} fixation is mapped to the fifth word in T .

The exact definitions for the five measures we examine are the following:

1. **First fixation duration** for the word w_x ,

$$ff(w_x) = d(k), \text{ such that } w(j) \neq x, j = 1, \dots, k-1, \text{ and} \\ w(k) = x$$

is the duration of a fixation when the reader enters the word w_x for the first time.

2. **Gaze duration** for the word w_x ,

$$g(w_x) = \sum_{k=a}^{a+b} d(k), \text{ such that } w(j) \neq x, j = 1, \dots, a-1, \text{ and} \\ w(a) = w(a+1) = \dots = w(a+b) = x, \text{ and} \\ w(a+b+1) \neq x \text{ or } a+b = m$$

is the cumulative sum of fixation durations in the word when the word w_x is entered for the first time.

3. **Total time** spent on the word w_x ,

$$t(w_x) = \sum_{w(k)=x} d(k),$$

sums the durations of all fixations mapped to the word during reading. Unlike $g(w_x)$, $t(w_x)$ includes also the regressive fixations to the word w_x .

4. **Number of fixations** to the word w_x ,

$$n(w_x) = \left| \{k \in [1, m] \mid w(k) = x\} \right|,$$

is the total number of fixations mapped to the word w_x .

5. **Regressions** to the word w_x ,

$$r(w_x) = \left| \{k \in [2, m] \mid w(k-1) > x, w(k) = x\} \right|,$$

is the number of inter-word regressive fixations mapped to the word. In other words, each fixation that enters a word from a word appearing later in the text increments the $r(w_x)$ score of the word where the fixation landed.

All of the measures listed can be computed in real time. Regressions can be thought to reflect problems either in the position from which the regression launches or in the position at which the regression lands. However, in our context, we use the measures in real time to discern the need for help. There is no point in providing help for a word the gaze is **leaving**, as any gloss judged necessary at that point would be given to a place that does not have the reader's attention. So, "regressions" covers only regressive saccades **entering** a word.

9.1.2 Measuring reading comprehension in non-ideal conditions

The findings of correlation between the scores for these five measures and comprehension difficulties (review Subsection 4.2.3) have been made in extremely controlled reading experiments. Commonly in experiments, head movements have been restricted with, for example, a chin, neck, or forehead rest or bite bar, and the drift from calibration has been controlled with repetitious recalibrations. Also, the stimulus text presented has often been limited to one line of text, thus circumventing problems with vertical inaccuracy. When the user's body movements are not intrusively restricted by external physical means, inaccuracy in the measured point of gaze invariably results. It can also be assumed that a situation where the user reads longer text passages is cognitively different from the situation in experiments studying the effect of comprehension difficulties while one is reading a sentence or a couple of sentences. When reading a longer passage of text, the reader has to integrate the new content into the preceding text, which may result in, for example, more regressions or slower reading.

In addition, most of the reading research experiments have concentrated on studying how reading comprehension difficulties are manifested in eye movements when one is reading sentences written in one's native language. Consequently, the emphasis has been on understanding syntactically complicated structures, like garden path sentences (mentioned in Subsection 4.2.3). In reading of text written in a foreign

language, syntactically complicated structures are, of course, a problem for a reader.

However, before one can understand the structure of a sentence, the first step is to understand the meaning of the words in the sentence. iDict is able to provide dictionary lookups for words and phrasal expressions; help in parsing syntactically challenging sentences is beyond the goals of the application. Thus, at this stage, the emphasis in the studies was on detecting comprehension difficulties on word and phrase level. The texts used in the experiments were excerpts from common prose text. Syntactically complicated sentences were not intentionally included.

We performed a series of experiments aiming to find out which of the measures are robust enough to be used in an interactive gaze-aware application, similar to iDict, when a user is allowed to behave more naturally.

9.2 EXPERIMENT ON USING THE MEASURES IN NON-IDEAL CONDITIONS

We conducted several pilot tests to refine our ideas of how the tests should be carried out. The observations in the pilot tests and the resulting experiment setup are described, next.

9.2.1 Experiment setup

The first observation from the pilot tests was that it is not a trivial task to motivate experiment participants to concentrate on the text they are reading. iDict is supposed to help readers who are motivated to understand what they read. In experiment conditions, the internal motivation is easily lost; the readers easily start to “mimic reading,” scanning through the text without actually comprehending what they read.

Also, finding out which words the reader is having problems with is difficult. In early experiments, the readers were allowed to indicate their need for help by pressing a button, and then to ask for help from the test supervisor, who acted in a “human dictionary” role. However, it was soon noticed that the test participants should not be allowed to speak out, press buttons, or give any other indications of the points at which they encountered a word for which they wished to get help. Attending to secondary tasks during a test immediately corrupts the eye movement data.

A third observation from the pilot experiments was that – especially when the head-mounted eye tracker was used – the participants reported being conscious that their eye movements were being recorded and that this might have affected their reading behavior. This observation accounts in

part for the skim reading noted above; the participants often tried to “perform well in the reading task” by reading faster than they would normally.

Special care was taken to avoid these problems in the experiment. The experiment setup is described in more detail below.

Participants

Ten students, five male and five female, participated in the experiment as a voluntary part of their course credit for the Introduction to Usability course. Their ages varied from 19 years to 35, the average being 24. Three of the participants wore eyeglasses, and two of them had contact lenses. Nonetheless, the calibration succeeded well for all participants. All participants had Finnish as their native language, and they had learned English at school as either their second or third language. Nine of the subjects considered their skills in English good and said that they read English quite fluently. One reported his skills in English as not very good and that reading English is toilsome for him.

Stimulus texts and motivation

Two means were employed for ascertaining that the subjects were concentrating on reading. The text was chosen carefully, with the participants’ characteristics borne in mind, and comprehension of the text was controlled after reading of the text blocks. The text was an extract from Roald Dahl’s short story “Sound Machine”; it generated an interesting, tense setup that enticed the subjects to read further. All of the subjects considered the text interesting, and some of them even asked for a reference for the book, because they wanted to read the whole story.

The story was divided into three blocks of text, presented in the Times font, in 12-point text with 1.5 line spacing on a 19" screen (with a resolution of 1024 x 768). Each of the blocks contained about 250 words. We motivated the subjects to comprehend what they read by telling them in advance that, after reading each text block, they would have to give a verbal review to the experiment supervisor. To avoid situations in which they would concentrate on memorizing the text, they were told that glancing over the text while giving their report was permitted.

Procedure

One hour was allocated for each subject to perform the experiment. For all but one subject, the time was sufficient. Before starting the experiment, the subjects filled out a personal information form and were informed of the overall procedure for the experiment.

To reduce the participants’ consciousness of the eye tracking, they were told it is important that they try to forget that their eye movements were

being recorded and that they try to read normally, just as they do in a normal situation. Additionally, reading of the first text was treated as a rehearsal for the two remaining readings. We mounted the tracker for the reader and rehearsed calibration of the tracker right at the beginning of the experiment, even though the first text was read without recording the eye movements. That was also told to the subjects, so they got used to the tracker and the situation while they knew that their eye movements were not being monitored. The tracker was not recalibrated during a reading session, but recalibrations were performed at the beginning of the two remaining reading sessions.

Each reading session was followed by a verbal review, in which the participant was asked to reread the text and indicate the words whose meanings he or she was not sure of. The request was phrased as “point out the words that caused you problems understanding the sentence. I mean the words about which you would have wanted to get automatic help or at least the words for which you would not have considered the help needless.” We made notes of both the list of problematic words for each participant and the participant’s success in giving the review.

In addition, the third reading session was followed by a request to write down a translation of the last block, so that we could check later whether the text was really understood. After the translation was written, we interviewed the subjects.

Apparatus

In this experiment, the fixations were recorded with the EyeLink system.

9.2.2 Overview of the data

Since the aim of the experiment was to find out which of the measures copes best with the non-ideal conditions, the fixation data were mapped to the focused words only by using the drift compensation algorithms; no manual correction was performed. In the context of analyzing new line events (described in Section 8.5), the mapping of fixations to the words was confirmed manually. In that experiment, the aim was to determine how this particular event during reading affects the reading path. This time we already knew that in ideal eye movement recording conditions the candidate factors do provide a reflection of comprehension difficulties.

The interview performed after the test revealed that the tracker did not disturb the reading much, at least not while the subjects were reading the last two text blocks. Several of them reported after the experiment that by the time they were reading the third text they had become interested in the text itself, wanted to find out what happens next, and had forgotten all about the eye tracking.

The average time used for the experiments, without the time used to write

the translation, was 24 minutes; the average time that the eye tracker was mounted was about 15 minutes. Writing down the translation of the third text block took another 20 minutes, usually.

The eye movement data recorded during reading of the third text block were analyzed. The block of text contained 252 words. Time spent reading the text varied from one minute to about five and a half minutes (see Figure 9.1). However, most of the readers spent less than two minutes on the reading. The time spent by participant P3 (338 s) differed substantially from that of the others. He was the one who considered his skills in English to be poor. Average reading time without P3 was 89 seconds.

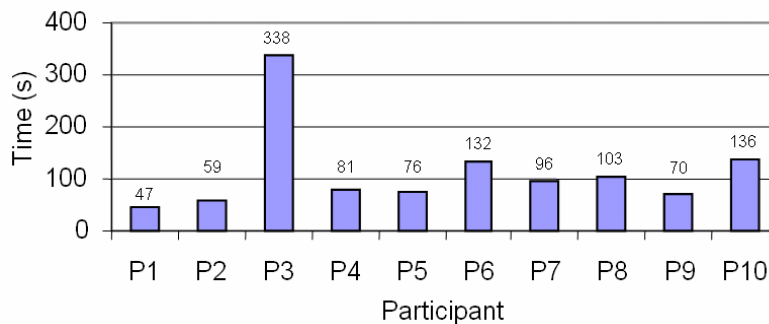


Figure 9.1: Time spent on reading the analyzed session by each participant.

In the experiment, the words of the analyzed text T were divided into two disjoint sets: **problematic words** (W^P) and **familiar words** (W^F).

The breakdown by problematic and familiar words was determined individually for each participant on the basis of their own reports right after reading of the text for the first time. We know (e.g., Rayner, 1995; Underwood & Radach, 1998) that some of the words do not get fixated on at all during reading; about one third of the words are originally skipped (Brysbart & Vitu, 1998). As we have noted, function words are skipped more often than content words. The average skipping percentage for function words has been measured to be as great as 80% of all function words, whereas only 15% of content words are skipped (Just & Carpenter, 1980; Rayner & Duffy, 1986; Reichle et al. 1998). In our experiment, it was interesting to notice that this held true even though the observations cited above were made for reading of text written in the reader's native language. The readers skipped 30% of all words.

Below, we will denote the subset of **fixated problematic words**¹ as W^P

¹ In our experiment, W^P was the same set as W^P for each reading session. In other words, none of the participants reported such a word to be problematic as would not have caused fixation during reading of the text for the first time.

and of fixated familiar words as W^F .

Requesting the subjects to pinpoint the problematic points in the text right after the text was read proved out to be a good choice. At that time, the participants still remembered how they reacted to different words during the first reading. Written translations were less useful than we had anticipated, because the translation situation was so different from the first-time reading. At this point, the subjects were reading the text through for the third time, and they had much more time to think about the meanings of words and sentences while they were writing the translation down. Some of the subjects made that remark themselves in the interview at the end of the experiment.

The total number of words identified as problematic was 88. However, 36 of these were pointed out by participant P3. The high number of problematic words that he pointed out after the session made his recollection of problematic words unreliable. Even though his information was interesting and revealed how the eye behavior of a reader with poor language skills differs from that of readers who considered their skills in the language good, analysis of how his comprehension difficulties are manifested in different eye movement measures cannot be considered very reliable. This is why participant P3 is omitted from subsequent analysis of the data. The rest of the participants were able to point out the problematic words quite convincingly; they accounted for 52 problematic words (Figure 9.2) from the analyzed reading session.

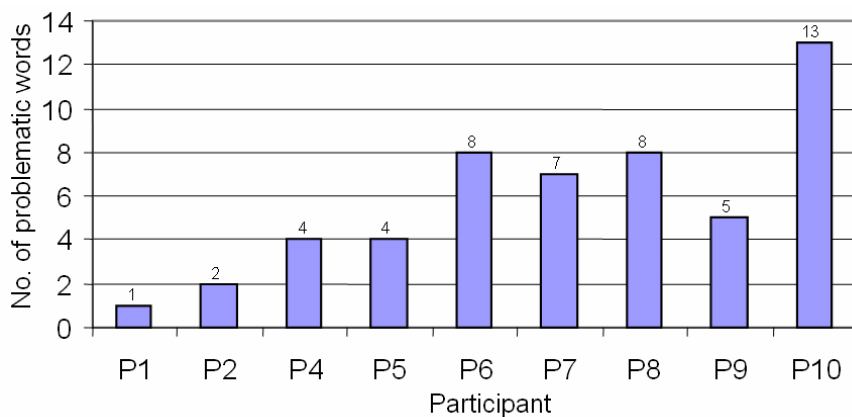


Figure 9.2: Number of problematic words identified by the participants.

9.2.3 Scores for different measures in the experiment

The eye movement data recorded in the third reading session were analyzed after the experiment. The candidate eye movement measures were analyzed both for the problematic words and for the familiar words. Below we introduce the average scores for each measure, by participant.

The means of personal average scores are computed; we call them mean scores for short. Thus, for clarity, we will subsequently use the term “average” (denoted as $\overline{measure}$) to refer to within-subject averages, and the term “mean” ($\mu_{measure}$) to refer to the mean over the whole data set (computed as the mean of the within-subject averages).

When appropriate, also the standard deviations for the measures are computed, and, similarly, to differentiate the personal standard deviation of a measure (within-subject) and the mean standard deviation for all of the data (between-subjects, computed as the mean of personal standard deviations), we adopt the following terms and notation. The personal standard deviation is denoted by s and the mean of personal standard deviations by σ .

First fixation duration for a word

As defined above, W^P is the set of problematic words a participant pointed out, and $W^{P'}$ is the subset of the fixated words of W^P . The average first fixation duration for problematic words for a participant is defined as

$$\overline{ff(W^P)} = \frac{\sum_{w_k \in W^P} ff(w_k)}{|W^{P'}|}.$$

The definition indicates that we should exclude the skipped words from the average. Since Just and Carpenter (1980), debate has continued as to whether skipped words should be taken into account in the calculation of averages such as the one above. The usual advice given is to consider what is sensible in the experiment in question. The often-presented justification for including the skipped words is that, while not fixated upon, they are probably perceived and cognitively processed, affecting the fixation durations for neighboring words. However, since there is no reliable method for parceling out the fixation duration effect for a skipped word in the various measures, we found it safer to leave out the skipped words.

The average first fixation duration for the words familiar to the participant is arrived at in the manner used for the problematic words

$$\overline{ff(W^F)} = \frac{\sum_{w_k \in W^F} ff(w_k)}{|W^{F'}|}.$$

First fixation duration averages computed for each of the participants, in Figure 9.3, reveal that when the measurements are done in non-ideal conditions, the problematic words are not distinguished from the familiar words; for four of the participants, the average fixation duration for familiar words was even longer than that for problematic words.

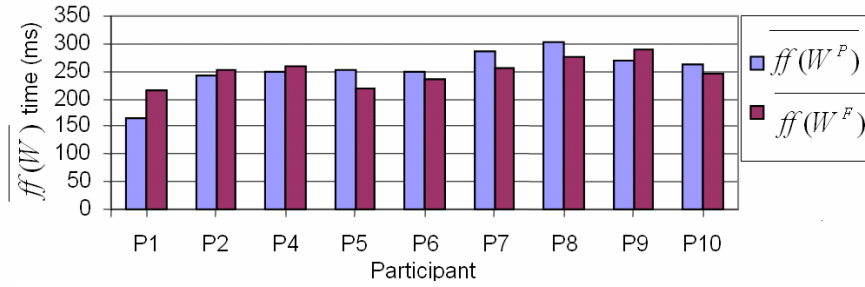


Figure 9.3: The average first fixation duration. Personal first fixation averages for problematic and familiar words.

The mean first fixation duration over the whole data set is denoted as μ_{ff} . In the data, the mean first fixation durations for problematic and familiar words were 253 ms (μ_{ff}^P), and 250 ms (μ_{ff}^F), respectively. Thus, there was no useful difference in first fixation duration between the problematic and familiar words.

Gaze duration for a word

The average gaze duration for a participant is defined for problematic words as

$$\overline{g(W^P)} = \frac{\sum_{w_k \in W^P} g(w_k)}{|W^P|},$$

and for familiar words as $\overline{g(W^F)}$, similarly.

The average gaze durations for problematic and familiar word are shown in Figure 9.4. Gaze duration performed somewhat better than first fixation duration; for seven of the participants, $\overline{g(W^P)}$ was longer than $\overline{g(W^F)}$. Participants P1 and P2, for whom average gaze duration was shorter for problematic words than for familiar words, were the ones with the fewest problematic words in the text (only one and two of them, respectively – review Figure 9.2). However, also for, e.g., P10, who reported 13

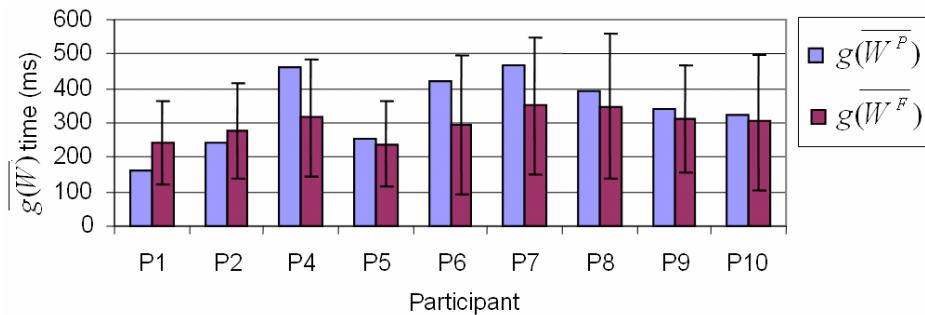


Figure 9.4: The average gaze duration. Personal gaze duration averages for problematic and familiar words, together with gaze duration standard deviation bars for familiar words.

problematic words, the difference in average gaze duration for problematic and familiar words was very small.

The mean gaze duration for problematic words (μ_g^P) was 341 ms, and the mean gaze duration for familiar words (μ_g^F) was 297 ms.

Even though the mean score is higher for problematic than for familiar words, the inspection of standard deviations reveals that the success of gaze duration is not convincing, either.

In Figure 9.4, also the personal deviation bars for duration of gaze on familiar words are displayed. Standard deviations for problematic words are not relevant, due to the small number of problematic words for some of the participants (review Figure 9.2). Mean standard deviation σ_g^F , the average of personal deviations for familiar words, was 169 ms. The standard deviation s_g^F for each of the participants encompasses the average gaze duration for problematic words. This means that familiar words cannot really be separated from problematic words on the basis of gaze duration.

Total time spent on a word

The third measure calculated on the basis of fixation durations is total time. The average total time for problematic words for a participant is defined as

$$\overline{t(W^P)} = \frac{\sum_{w_k \in W^P} t(w_k)}{|W^P|},$$

and for familiar words as $\overline{t(W^F)}$, similarly.

Figure 9.5 shows that for all participants the average total times were higher for problematic words than for familiar words.

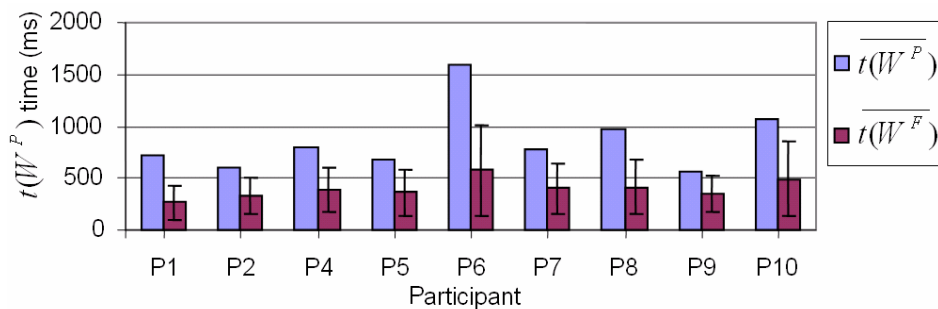


Figure 9.5: The average total time. Personal total time averages for problematic and familiar words, together with total time standard deviation bars for familiar words.

The mean total time for problematic words (μ_t^P) was 864 ms, and the mean total time for familiar words (μ_t^F) was 398 ms.

In addition, for all participants, $\overline{t(W^P)}$ was long enough to reside well outside the standard deviation range of total time spent on familiar words. The mean standard deviation for familiar words, σ_t^F , was 248 ms.

Number of fixations for a word

An increasing number of fixations for a word has also been found to correlate with comprehension difficulties. The average number of fixations for problematic words for a participant is defined as

$$\overline{n(W^P)} = \frac{\sum_{w_k \in W^P} n(w_k)}{|W^P|},$$

and for familiar words it is $\overline{n(W^F)}$, similarly.

As was the case with average total time scores, for all participants this measure yields higher averages for problematic words than for familiar words (Figure 9.6). However, for three of the participants (P2, P8, and P9), $\overline{n(W^P)}$ falls slightly outside the range of s_n^F .

The mean number of fixations for problematic words (μ_n^P) was 3.2, and the figure for familiar words (μ_n^F) was 1.7. The mean standard deviation for familiar words (σ_n^F) was 1.0.

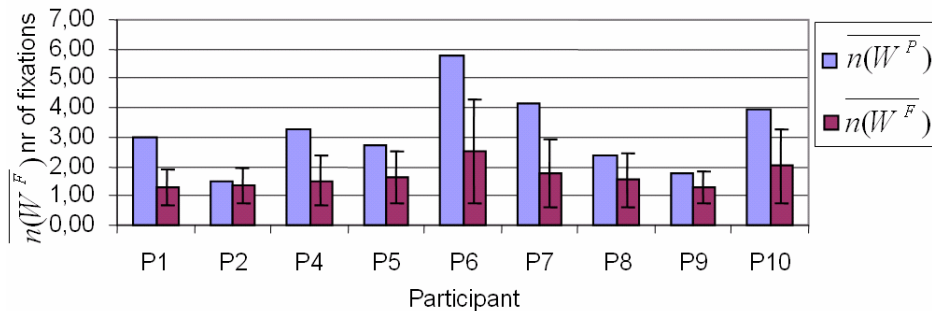


Figure 9.6: The average number of fixations. Average number of fixations for problematic and familiar words, together with the fixation number standard deviation for familiar words.

Regressions to a word

The total number of fixations in the eye movement data analyzed (without P3) was 2,737. Our implicit definition for a regressive fixation (in the context of defining $r(w_k)$, the regression measure) was that a fixation is regressive if the previous fixation was mapped to a word that appears

later in the text. Similarly, a regressive saccade takes the gaze from a word to a preceding word. Therefore, horizontally backtracking saccades do not necessarily yield a regressive fixation (for example, in the case of a new line event). Nor is a fixation mapped to the same word as the previous fixation (an in-word fixation, even if there is regression horizontally) a regressive fixation. In our data, 17 percent of fixations (458) were regressive fixations. The interpretation of the $r(w_k)$ score is that, for example, if the score is 2 there were two regressive fixations mapped to the word w_k .

The average number of regressions for a participant is, for problematic words, defined as

$$\overline{r(W^P)} = \frac{\sum_{w_k \in W^P} r(w_k)}{|W^P|},$$

and for familiar words as $\overline{r(W^F)}$, similarly.

The range of values for the regression measure is small, starting from 0, and having only a few scores higher than 3. Average and standard deviation figures for problematic and familiar words by participant are, analogously to the previously discussed measures, displayed in Figure 9.7.

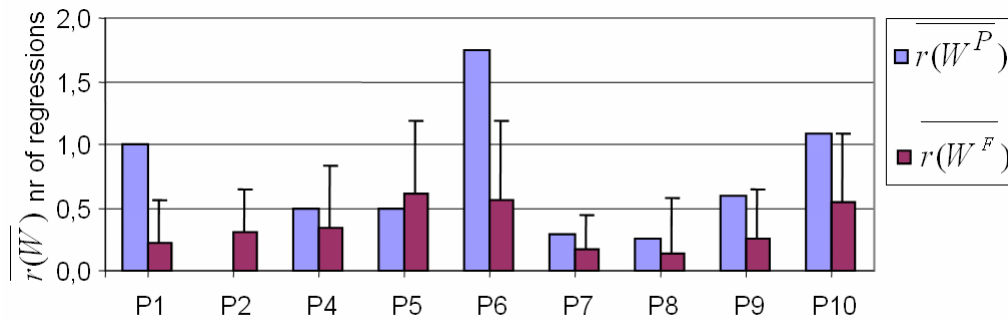


Figure 9.7: The average number of regressions. Personal averages for regressions to problematic and familiar words, together with standard deviation bars for regressions to problematic and familiar words.

For participants P2 and P5, $\overline{r(W^F)}$ is higher than $\overline{r(W^P)}$, and in most cases the range of standard deviation s_r^F exceeds the average number of regressions for problematic words. The mean number of regressions to problematic words (μ_r^P) was 0.6, and to familiar words (μ_r^F) it was 0.3.

A visualization of the distribution of regressions for problematic and familiar words gives us a better idea of the relationship between regressions and reading of problematic words. Figure 9.8 displays the distribution for words having one, two, or more $r(W)$ scores, for problematic and familiar words.

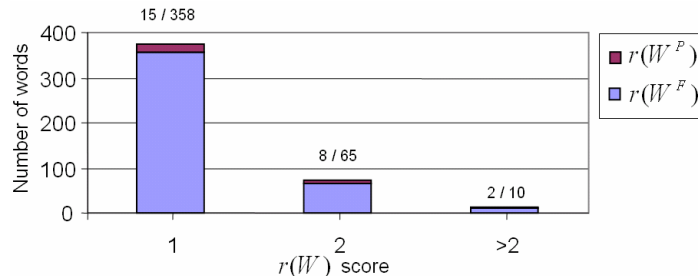


Figure 9.8: The distribution of regressions. Distribution of regressions to problematic and familiar words.

The distribution is computed from the whole set of eye movement data analyzed (recorded for the nine participants covered in this discussion). The figure illustrates that the number of regressions to a word does not screen out the problematic words from the recorded eye movement data.

9.2.4 Discussion and conclusions

The analysis above reveals that some of the eye movement measures that we have examined perform better in sub-optimal eye tracking conditions than others do. All but $\overline{ff}(W)$ showed a trend of increasing scores for problematic words as opposed to familiar words (the mean of the average scores computed was higher for words belonging to W^P than for words in W^F). However, total time was the most robust of the measures – it was the only one for which the scores of $\overline{t}(W^P)$ were higher than $\overline{t}(W^F)$ with s_t^F added for each of the participants.

Number of fixations performed a little better than gaze duration did, in the sense that for each participant $\overline{n}(W^P)$ was higher than $\overline{n}(W^F)$. We wish to find a condition for triggering the help function that would work in a wide range of eye tracking environments. Number of fixations is sensitive both to the eye tracking equipment (the time frequency of recorded samples) and to the algorithm used for detecting the fixations. While the measure performed moderately well in segregating the problematic and familiar words in the experiment with EyeLink (which had a relatively high sample rate of 250 Hz), it would probably perform much worse if the tracker had a sample frequency of 50-60 Hz.

Since each candidate measure, apart from first fixation duration, tended to show higher mean scores for problematic than familiar words, we could try creating a composite function using multiple measures. An appropriately

constructed composite triggering function could perform better than any one measurement.

However, if we consider what total time actually indicates, we note that it is already a kind of composite function of most of the other elements. Thus, we should actually not find it surprising that it performed so well compared to the other measures. Both gaze duration and number of fixations end up increasing the total time spent on a word. The same goes for regressions.

This is why we selected total time as the basic indicator for iDict to use in determining when the reader probably is in need of a gloss for a word. In the following section, we will consider more closely how the total time should be used in the iDict context. What is a suitable threshold for triggering the gloss? How could we add to the accuracy of gloss triggering? Should the threshold vary across readers? Could we use some characteristics of a given piece of text for increasing the fidelity of the gloss provided?

9.3 TOTAL TIME AS A BASIS FOR DETECTING COMPREHENSION DIFFICULTIES

In this section, we will first, using the data recorded in the experiment described above, determine an appropriate value for use as the threshold at which the total time spent on a word triggers a gloss. We continue by analyzing the data in order to find out whether we should personalize the threshold for the readers. After that, we explore whether we can use word frequency and the length of a word to improve the accuracy of the automatically given glosses.

9.3.1 Total time threshold

What is the threshold for $t(w_k)$ (subsequently denoted as th) after which the gloss for a word w_k should be triggered? The aim is to segregate familiar and problematic words on the basis of the $t(w_k)$ score accumulating for the word. In the previous experiment, the mean total time, μ_i^P , was substantially higher (864 ms) than μ_i^F (398 ms). So, in order to avoid help being triggered for familiar words, th should be higher than μ_i^F , but how much higher?

Using the mean standard deviation, σ_i^F , as the unit for the scale to increase th takes into account the variation of the gaze behavior. Approximately two thirds of the data points lie within one standard deviation of the mean (e.g., Howell, 1987, p. 41). Thus, if the threshold is computed as the sum of the mean total time and the standard deviation of total time ($th = \mu_i^F + \sigma_i^F$), it would end up triggering about one third of the

familiar fixated words.

The variation in total time is high (in the experiment reported above, $\mu_t^F = 398$ ms and $\sigma_t^F = 248$ ms), so it is impossible to achieve the ideal scenario – i.e., to find a threshold that would end up triggering only the problematic words. Let us construct the threshold as a sum of mean total time and mean standard deviation multiplied by some constant (a **threshold factor**, denoted as a) as follows.

$$th = \mu_t^F + a\sigma_t^F.$$

When this is applied to the data recorded in the experiment, we can see (Figure 9.9) the number of problematic and familiar words that would have triggered the automatic gloss with different values of th (as a function of the threshold factor, a).

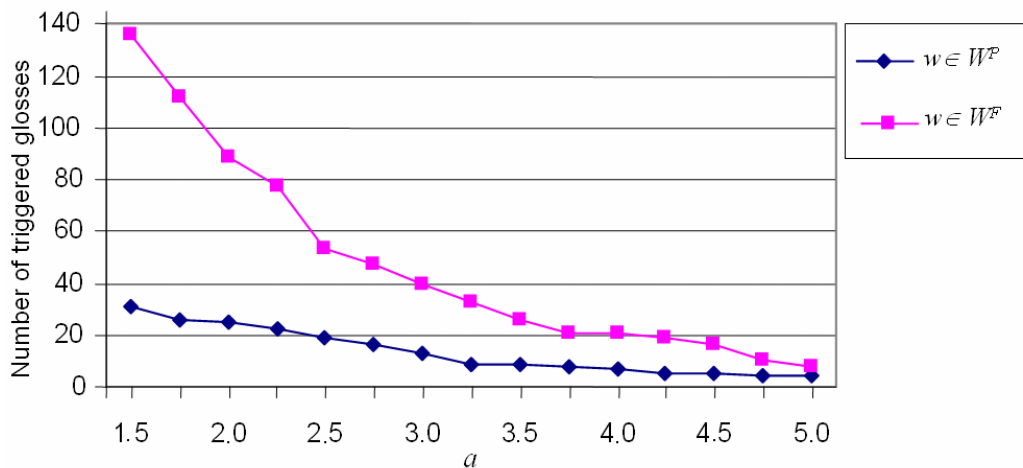


Figure 9.9: Total time threshold and glosses triggered. Number of glosses for problematic and familiar words as a function of the threshold factor, a .

We should remember that in the experiment (from which the data are used to hone the total time threshold) the participants did not get automatically triggered glosses. The data indicate the readers' natural reading behavior; delay on words was affected only by their comprehension of the sentences read. However, when testing the application with the automatically triggered glosses, we noticed that readers semi-intentionally tended to prolong their fixations on problematic words once they became acquainted with the application's behavior. That is why it is more important to concentrate on finding a threshold that does not give too many **false alarms** – i.e., false positives, triggered familiar words. Of course, the threshold should still result in triggering of a fair number of glosses for problematic words, but because in the real situation the total times for problematic words tend to be longer, we do not worry so much about their total times remaining under

the triggering threshold.

In Figure 9.9, the number of false alarms decreases more sharply until reaching the point of the threshold value computed with an a value of 2.5. At that point, th is 1018 ms. The number of false alarms at that point would have been 53, and the number of correctly triggered glosses would have been 19.

Since the decrementing of false alarms seems to temper at the point of $a = 2.5$ and the decrementing of correctly triggered glosses is small but steady (i.e., there are no distinctive points in the decreasing curve), we choose to use 2.5 as the value for the threshold factor, a , for computing th .

To summarize, applying the general threshold

$$th = \mu_i^F + a\sigma_i^F, \text{ with } a = 2.5,$$

computed from the mean total time measurements to the data from our experiment produces the th value 1018 ms and would end up triggering alerts for

- 36.5% of the problematic words (19 out of 52) and
- 2.4% of the familiar words (53 out of 2,216).

The above search for an appropriate threshold value for total time legitimates a question regarding the use of the mean values of total time scores for computing the threshold. Why not just find out the best threshold time th in milliseconds?

The reason for expressing th in terms of μ_i^F and σ_i^F is that doing so gives us the possibility of setting the threshold according to the particular circumstances of reading at hand. It has been detected that, for example, the complexity of the text affects the eye behavior (e.g., Frazier & Rayner, 1982; Rayner & Pollatsek, 1989). Presumably, also the reader's skills in the language used affect reading. Next, we will use the data from our experiment to find out how much the personalization of th would affect the number of glosses triggered.

9.3.2 Personalizing total time threshold

The values of $\overline{t(W^F)}$ across readers varied from 271 ms to 577 ms (the mean being 398 ms). This suggests that instead of using a general th , computed from the mean of the total time figures, it might be worth setting the threshold individually for each reader, computed using the personal $\overline{t(W^F)}$ and s_i^F values.

Figure 9.10 shows a comparison of the number of false alarms if the

general versus the personalized th value were applied to the data, with varying threshold factor values. As assumed, the general threshold triggers more false alarms than the personalized threshold does. For example, for the threshold chosen above (with $a = 2.5$), the difference would be 12 words (0.5% of the familiar words). Hence, the difference is not large but is still coherent.

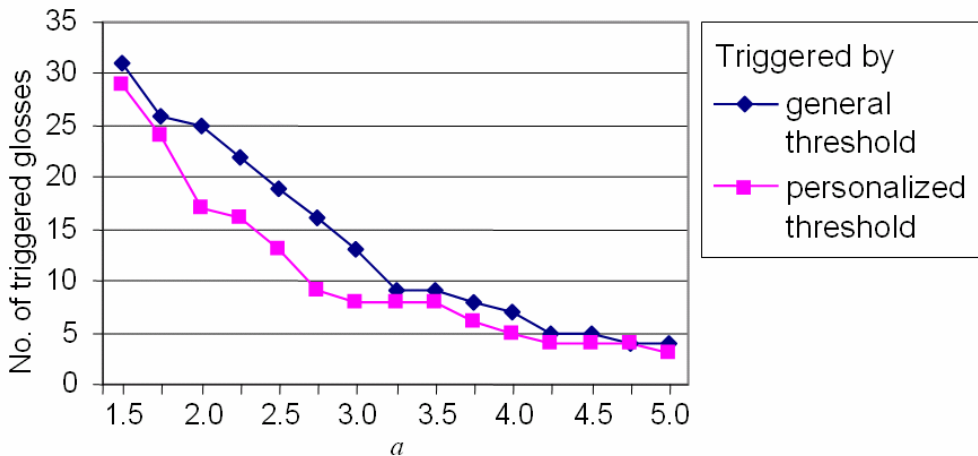


Figure 9.10: Personalized threshold and false alarms. Number of incorrectly triggered glosses with general and personalized th values as a function of a .

Personalization does have the desired effect, although not big, on the number of false alarms. How does it affect correctly triggered glosses?

The number of correctly triggered glosses with the general and the personalized th (when applied to the experiment data) values is shown in Figure 9.11. Personalizing the total time threshold values would decrease also the number of correctly triggered glosses. For the chosen threshold ($a = 2.5$), the difference would be six words (11.5% of the problematic words). Again, the effect is not big, but this time it is an undesirable one.

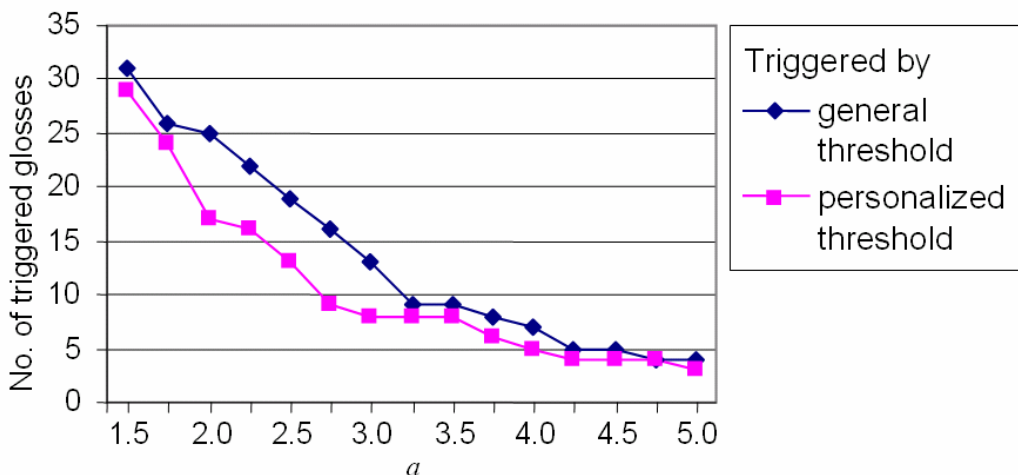


Figure 9.11: Personalized threshold and correctly triggered glosses. Number of glosses triggered with general and personalized th values as a function of a .

Thus, by replacing the general threshold with a personal threshold we did not obtain substantial improvement in the triggering accuracy. This may be a consequence of the homogenous test situation. The text was the same for all readers considered, and they all had quite good skills in English. However, even though the nine readers reported having good skills in English, their skills must differ to some extent. Perhaps just a few categories (e.g., “beginner,” “fair,” “good,” and “fluent”) would do; the better the reader, the shorter the threshold time. According to the analysis in the previous section, 1000 ms (rounded from 1018 ms) should be a suitable total time threshold for good readers.

Could we use some features of the text document for improving the triggering accuracy? Naturally, the readers need help more often with low-frequency (i.e., rare) words than with high-frequency words. We also know that gaze duration for a word correlates positively with the word’s length (e.g., Rayner, 1998).

9.3.3 Word frequency and word length

In strictly controlled conditions, the word frequency’s effect on a word is that the lower the frequency of the word the longer the reader spends reading the word (e.g., Inhoff & Rayner, 1986; Rayner & Duffy, 1986). This is in addition to, as noted above, the correlation of word length with time spent on the word.

Word frequency

However, the word frequency effect is not strong enough to be of use in the non-ideal conditions studied here. We did decide to use the information on word frequency, but the other way around. Since the mapping of fixations to the corresponding words is not optimal, due to the inaccuracy discussed previously, the reader’s eye movements may trigger glosses for familiar words that probably are false alarms. We could block the glosses for the most frequent words. However, we cannot draw a clear line between the words for which a reader may want to receive a gloss and those where one is not needed; the familiarity of words differs with the individual.

That is why we did some testing to see how setting individual thresholds for words according to their frequency would affect the triggering accuracy. We used the British National Corpus database (BNC, 2005) to determine the frequencies of the words in the text. BNC lists the number of occurrences for 6,318 of the most frequently used words in a 100-million-word database. Figure 9.12 shows the cumulative coverage of the 6,318 most frequent words in the database.

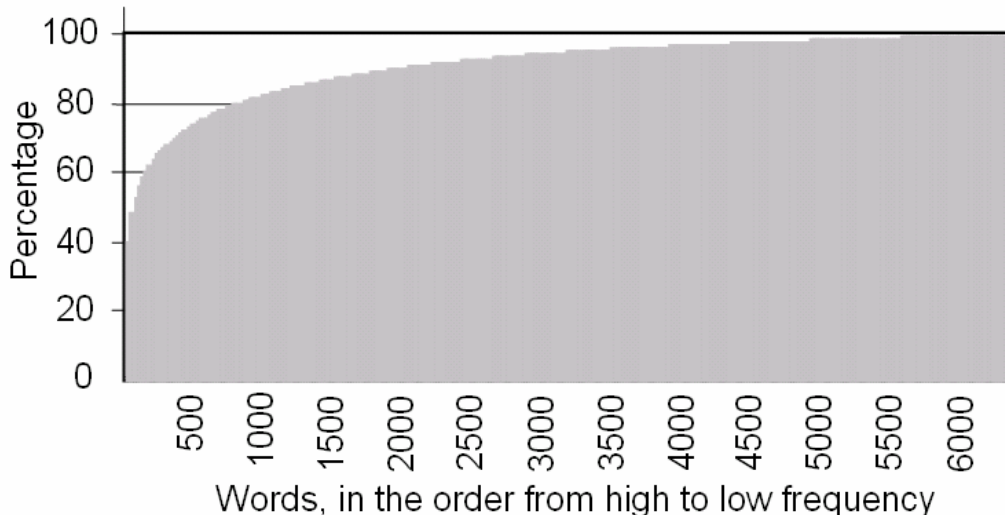


Figure 9.12: Distribution of words in BNC. Cumulative coverage of the 6,318 most frequent words in BNC's 100-million-word database.

For example, the most frequent word ("the") occurred 6,187,267 times in the database. Occurrences of the most frequent words in the database dominate the distribution of words; each of the 99 most frequent words occurs more than 100,000 times in the database. These together cover almost half of the instances of all the words in the database. These together cover almost half of the instances of all the words in the database. The least frequent words on the list still occurred more than 800 times each in the database; as an example, one of them is the word "voucher," which occurred 867 times in the database. The first 6,000 words in the list cover about 85% of the total number of all words in the database.

We will call a word's order number in BNC the word's frequency number, $freq(w)$. So, for example, $freq("the") = 1$ and $freq("voucher") = 6000$. The number was then used to set an individual threshold for each of the words. The 100 most frequent words in the list (i.e., those with $freq(w) \leq 100$) were given the total time threshold th_h (the total time threshold for high-frequency words, see Figure 9.13). The words with $freq(w) > 6000$ and words that were not included on the BNC list at all were given the total time threshold th_l (the total time threshold for low-frequency words). Thresholds for the remaining words were linearly scaled to the range between th_h and th_l .

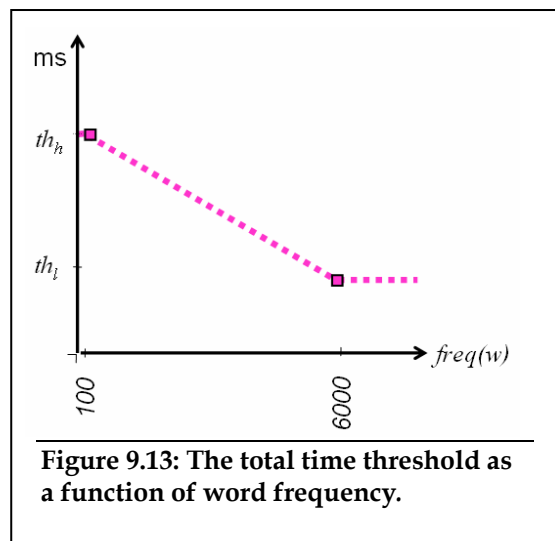


Figure 9.13: The total time threshold as a function of word frequency.

The total time threshold for word w_k when word frequency is taken into account is computed according to the formula

$$th^{wf}(w_k) = th_h + b(freq(w_k) - 100), \text{ when } 100 < freq(w_k) < 6000.$$

Here 100 is the number of high-frequency words assigned the threshold th_h , and the slope b depends on the thresholds given for high frequency and low frequency according to the expression

$$b = \frac{th_h - th_l}{100 - 6000}.$$

Figure 9.14 shows the number of glosses triggered for problematic and familiar words when the words are assigned individual thresholds according to the formula presented. The numbers are expressed as a function of th_h where th_l is kept constant. The threshold for low-frequency words (th_l) was set to the value 1000 ms, which we found to be a suitable threshold above (Section 9.3.2).

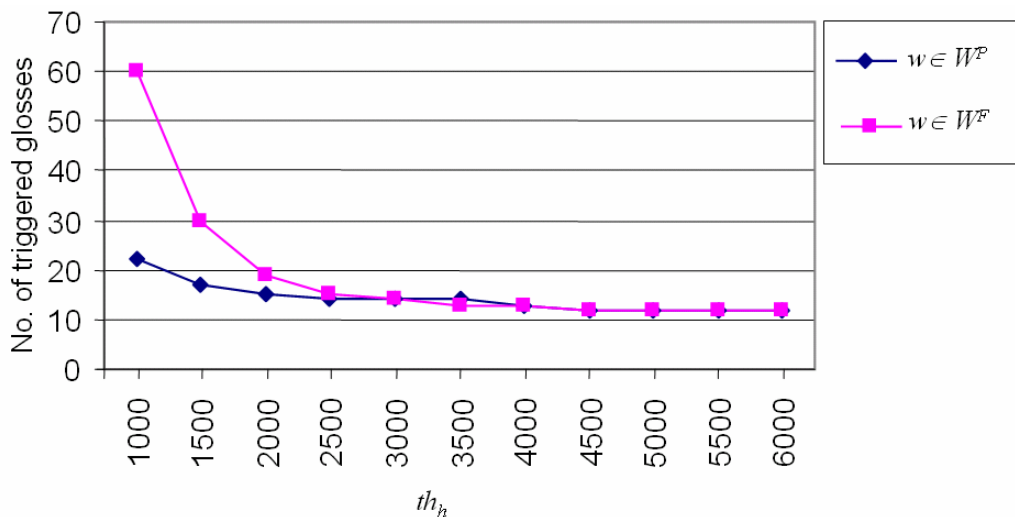


Figure 9.14: Glosses triggered, with a varying total time threshold. Number of glosses triggered for problematic and familiar words when words are assigned individual total time thresholds according to their frequency.

When th is raised from 1000 ms to 2000 ms, the number of false alarms decreases rapidly. After that, the decrementing gets slower. Also, the number of correctly triggered glosses decreases with the th value's move from 1000 ms to 2000 ms, but much more slowly than that of false alarms. Thus, by increasing the threshold for the most familiar words to 2000 ms, we screen out false alarms to a level where only about the same number of familiar as problematic words trigger a gloss.

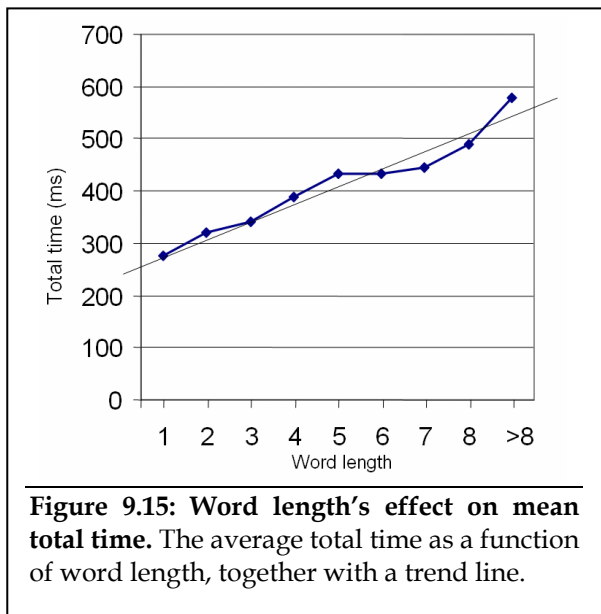
The assumption was that raising the thresholds for high-frequency words would reduce the false alarms produced by the inaccuracy of the measured point of gaze. This was verified by the observation that the false

alarm effect decreased when the mapping of fixations to corresponding words was confirmed manually.

When applied to the data from our experiment, the $th^{wf}(w_k)$ values would end up triggering

- 28.8% of the problematic words (15 out of 52), and
- 0.9% of the familiar words (19 out of 2,216).

The number of false alarms is very satisfactory. The number of correctly triggered translations could be higher; one more parameter that we could use to improve that rate is word length, examined next.



Word length

It has been shown that the length of a word affects the gaze durations for the word (Rayner, 1998). Figure 9.15 shows that in our data the effect also applies to the mean total times. In the figure, the mean total times are shown as a function of word length (in characters). Mean total times are computed for familiar words. Problematic words are omitted because in their case the word length is not the prime reason for the increased time spent on the word.

The equation for the trend line drawn in Figure 9.15 (computed via the least squares method) is

$$y = c * x + k,$$

where $c = 33$ and $k = 249$ ms. The average length of words on the BNC list is seven characters. Let us now set the total time threshold, which takes lengths of words into consideration, for a word w_k to be

$$th^{wl}(w_k) = th^{wf}(w_k) + c * (length(w_k) - 7).$$

The threshold for seven-character words receives the value we assigned earlier, but each additional character increases the threshold by 33 ms; correspondingly, for shorter words, each character decreases the threshold by 33 ms. When the th^{wf} thresholds were replaced with the th^{wl} thresholds and the new thresholds were applied to our data, the change in the number of glosses triggered was almost nonexistent. The number of glosses for problematic words did not change at all (15 glosses), and the

the threshold for a word.

By replacing the general threshold with personal thresholds, we caused the number of false alarms to increase slightly. We noted that the fact that the effect was not bigger may derive from the homogenous test setting. When the readers had good skills in English and the text was normal prose (not, for example, complex text from a specialist field), personalizing the threshold did not pay off. In this work, the subsequent tests were performed using a general threshold. However, by defining the threshold as a function of total time scores, we retain the ability to automatically adjust the threshold to the specific circumstances of the reading.

The word frequency effect helped to filter out false alarms while not dramatically decreasing the number of correctly triggered glosses. Scaling the thresholds for high- and low-frequency words to a range of 2000 ms to 1000 ms (corresponding threshold factors were 6.5 and 2.4, the latter rounded from 1018) ended up triggering

- 28.8% of the problematic words (15 out of 52) and
- 0.9% of the familiar words (19 out of 2,216).

The effect of word length on total times seemed not to be strong enough to improve the triggering accuracy.

Thus, the resulting function for use in iDict for computing the total time threshold for a word w_k was formed using the general total times and word frequency, as follows

$$th(w_k) = th_h + \frac{th_h - th_l}{100 - 6000} (freq(w_k) - 100), \text{ where}$$

$$th_l = \mu_t^F + a_l \sigma_t^F \text{ and}$$

$$th_h = \mu_t^F + a_h \sigma_t^F ,$$

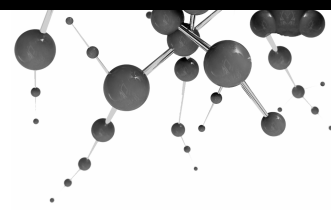
$$\text{when } 100 < freq(w_k) < 6000.$$

The constants derived by optimizing the triggering accuracy in the reading path data recorded in our experiment were $th_h = 2000$ ms ($a_h = 6.5$) and $th_l = 1000$ ms ($a_l = 2.4$). With substitution into the function, the derived threshold function reduces to the form

$$th(w_k) = 2000 \text{ ms} - 0.17(freq(w_k)) \text{ ms},$$

$$\text{when } 100 < freq(w_k) < 6000.$$

We have now explained how iDict keeps track of the point of reading (Chapter 8) and how it decides whether help is needed at that point (Chapter 9). Next, we turn to the question of how the help should be given when it is needed.



10 Interaction Design of a Gaze-Aware Application

In the introduction and when reviewing attentive interfaces in Chapter 4, we indicated that eyes are potentially an extremely valuable source of additional input information in attentive applications. On the other hand, the two previous chapters revealed the problems of using real-time eye input for adapting the application's behavior to the user's intentions.

We have observed that many of the problems can be avoided, or at least mitigated, through cautious interaction design decisions. In this chapter, we discuss the lessons learned in designing, implementing, and testing iDict. Many of the interaction issues are common to all gaze-aware applications. The discussion aims to help developers of gaze-aware applications avoid common pitfalls. We present our observations as guidelines for designing similar applications.

10.1 NATURAL VERSUS INTENTIONAL EYE MOVEMENTS

The original goal of iDict was to provide help proactively, on the basis of natural eye movements (Hyrskykari et al., 2000; Hyrskykari et al., 2003). However, when testing the application, we found that users quickly adapt their gaze intentionally, or semi-intentionally, to get the application to react. Still, we think that the basic idea of making use of the user's natural eye movements is realized in iDict. Nonetheless, the fact that the users are aware of the ongoing eye tracking blurs the boundary between natural and intentional eye movements. This should be taken into account, and in some cases can even be exploited, in the design of a gaze-aware application.

We encountered the issue of the murky distinction between natural and intentional eye movements even in our preliminary tests, when some readers reported that they could not get a gloss for the word even if they “tried to get iDict to respond.” A closer look at the eye movement data showed that the readers had an ability to make surprisingly long continuous fixations when they concentrated on staring at a word. The fixations reported by the eye tracker could last from 1000 ms to even more than 2000 ms. In early versions of our application, a fixation was not processed by the application until it had ended. Consider a situation in which a reader makes two or three “natural” fixations on a word. The fixation lengths vary from 200 to 400 ms, thus possibly not yet exceeding the total time threshold for the word. Then, the reader intentionally starts to stare at the word, inducing an unnaturally long fixation for the word, but the application does not receive the fixation data, and total time for the word does not increase as long as the fixation continues. Readers in this situation got the impression that the application froze; they stated that “the application does not respond to my request.” That is why we could not settle with the fixations the tracker provided to us and therefore used the raw data instead, choosing to update the word’s total time after every 500 milliseconds even if the fixation still continued.

The rest of our eye-input-related observations are divided into three categories. They contribute to three more general design guidelines that should be taken into account by designers of gaze-aware applications. These principles are by no means new, but the fact that they apply for applications that in most cases are proactive by nature is interesting. The principles are (1) appropriate feedback, (2) controllability, and (3) unobtrusive visual design. Each is discussed in more detail below.

10.2 APPROPRIATE FEEDBACK

What is the role of feedback in gaze-aware applications, which often are proactive by nature – at least to some extent? Should we hide the reasoning behind the automatically triggered actions from the user? The fundamental issue in proactive computing is to decrease the burden the user carries when interacting with computer-based applications; proactive environments aim to anticipate our needs and act on our behalf (Tennenhouse, 2000). Non-command interfaces (Jacob, 1993; Nielsen, 1993) have a parallel goal of drawing the user’s attention away from the interfaces so that it can be directed to the task itself. The “transparent interface” concept is frequently used in the same sense. For example, Ishii (2004) characterizes a transparent interface as a matter of the user’s focus of attention and consciousness, as follows:

10.2 Appropriate feedback

A transparent interface (or tool) is one that does not get in the way, allowing users to concentrate on the task at hand.

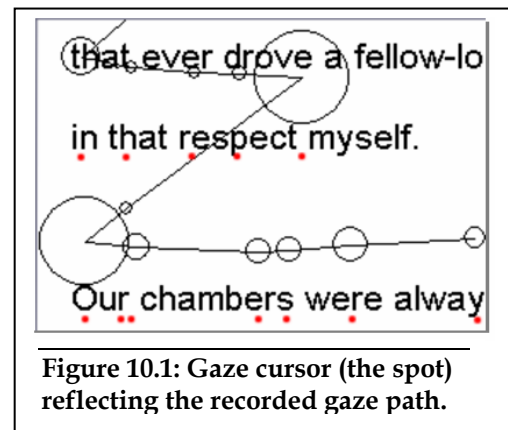
This may lead us to conclude that, in order to make the interface transparent, we should hide the operation of the application from the user. However, our experience, as in the example with prolonged fixations, demonstrated that users become confused if they do not understand the basic principles of the automatically triggered actions. In particular, when an unexpected, or erroneous, action takes place, understanding why it happened would help the user to accept the action.

10.2.1 Feedback on measured gaze point

In iDict, the primary feedback is the triggered gloss, but inaccurately measured fixations do get mapped to a wrong word, resulting in wrong glosses. iDict offers three different feedback options for monitoring the eye movement interpretation. The user can choose which one of them, if any, is activated.

Gaze cursor

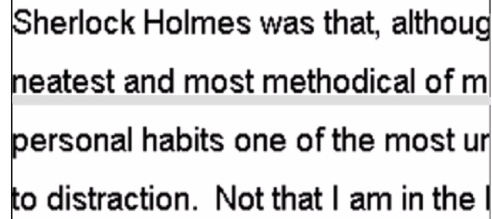
The obvious feedback is to show a small gaze cursor, which renders the measured point of gaze on the screen. This gives the reader an opportunity to intentionally “look off” (for example, to the right of the actual target) to get the target word construed correctly. It seems clear that a straightforward implementation of this feedback is not acceptable. The gaze cursor is seldom precisely where the user is really looking, and trying to control the measured point distracts from the process of reading. Furthermore, the constant movement of the visualized gaze cursor distracts the reader quite a lot, as noted already by Jacob (1993). Since in iDict we track the line of reading, we used the information to make gaze cursor movements steadier, and we tied the vertical coordinate of the gaze cursor to the presumed line of reading (Figure 10.1).



This generated an illusion that the gaze cursor locked on to the line the user was reading. Without tracking of the line currently being read, the gaze cursor was very unstable and provoked the reader to follow its movements; now, the gaze cursor (the spot in Figure 10.1) appears to follow along smoothly.

Line marker

The second form of feedback the reader may choose is a line marker, which is a faint gray underline below the line of reading – or below the



Sherlock Holmes was that, although
neatest and most methodical of m
personal habits one of the most ur
to distraction. Not that I am in the l

Figure 10.2: Line marker helping the reader to stay on line.

line that iDict assumes to be the line of reading (Figure 10.2). We presumed that the line marker is a more sensitive way to show feedback. It generates less visual noise since it changes only when the line of reading changes. Some of the test users reported that the line marker does not disturb the reading process; on the contrary, it helps the reader to remain on the right line. The line marker is

particularly helpful when the reader moves to the next line – and also after checking of a dictionary entry for a word, guiding the reader to return to the right line.

Target word

The line marker gives the user feedback on the accuracy of mapping of the measured fixations to the line of reading, but not feedback on horizontal accuracy. The gaze cursor does carry the latter data, but this information is actually of too fine a granularity for the reader. If preferring to see the feedback in greater accuracy than on line level, the user would probably be interested in the word the gaze is mapped to at the moment, not the exact position of the measured fixation point.

The third feedback option indicates the mapped word to the reader by changing its color by just an observable amount, creating a mental picture of “pushing the word by gaze.” The threshold for triggering the “push” can be set according to the total time accumulated for the word. For example, the word may be defined to change its color just a little before a gloss for it is given. Thus, needless visual noise is minimized.

In fact, our primary observation regarding the feedback is that we should minimize the visual noise by trying to avoid feedback for which the user has no use. For example, one can choose for feedback in iDict to be given only when the user does not get glosses for the word desired and thus has to make manual correction to adjust the traced track of reading.

10.3 CONTROLLABILITY

Another issue that should be paid attention to is controllability: with proactive applications, the user often experiences a loss of control. The user does not know what is happening, why it is happening, and whether there is anything to be done to affect it.

10.3.1 Control over when the gloss appears

As discussed above, the feedback provided helps the reader to understand what happens, but that is not enough. Also, the user should be able to

supplied glosses but also eliminates the benefit of getting the gloss without any additional effort. Still, it saves the user from having to grab the mouse and move the cursor onto the word. The user can keep the hands on the keyboard, possibly using the arrow keys to correct the mapping of the word, and press the spacebar to obtain the gloss.

10.3.2 Control over the dictionary entry

What is an acceptable level of proactivity when dictionary entries are retrieved and displayed in the dictionary frame?

Dictionary entry – for which word?

What does the reader expect to see when moving the eyes to the dictionary frame? There are two plausible possibilities. The word for which the dictionary entry is given may be either the last word the reader focused (fixated) on or the word for which the most recent gloss was given.

When we tried the first solution, the effect was that the reader was never quite sure for which word the dictionary lookup would be given. The reader is not necessarily fully aware of which word the eyes were on at the moment of the decision to look for additional help. Moreover, the last fixation may sometimes be mapped to a wrong word due to inaccuracy problems. Without feedback concerning fixated words, the words for which the reader receives dictionary entries may seem somewhat random.

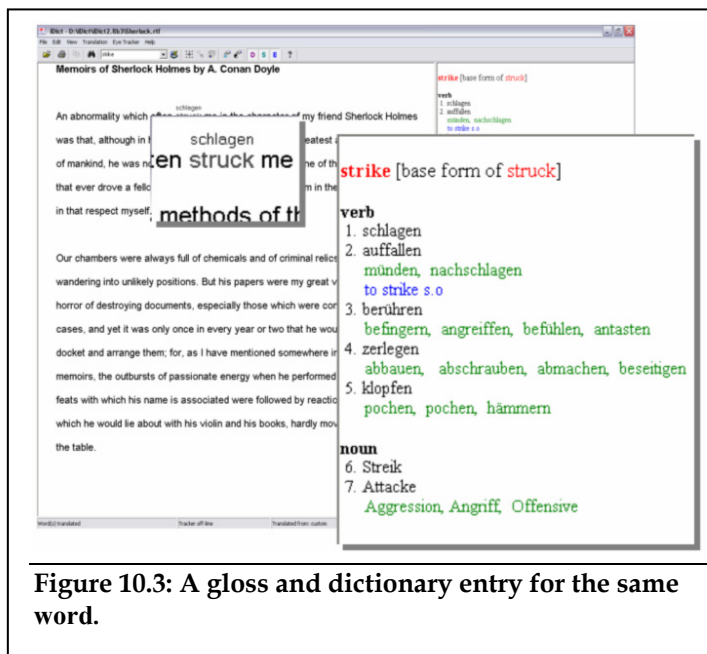


Figure 10.3: A gloss and dictionary entry for the same word.

As before, the principle of understandable behavior was the principal guideline for designing this feature, and we chose the second solution in iDict's implementation. When the gloss and the dictionary entry are given for the same word, the reader knows what to expect when looking in the dictionary frame. The reader has the feeling of being in control of the contents of the dictionary frame. A reader first triggers a gloss for a word

in the text and after that asks for more details if still in need of them.

In order to avoid needless visual noise, the dictionary frame is updated (to show the new entry) only when the reader turns his or her eyes to it;

otherwise, its contents remain stable.

Fixating on empty space

At first we wondered why it was so difficult for the readers to trigger the first dictionary entry by looking at the dictionary frame. By examining the gaze path, we realized that instead of looking in the frame, the users had their gaze drawn by the frame borders. Consequently, iDict did not react. In the beginning, there was nothing in the frame to look at; it is difficult to fixate on an empty space. The problem disappeared when we initialized the dictionary frame at the beginning of each session by displaying in the frame the prompt “Look here to get a dictionary entry.”

10.4 UNOBTRUSIVE VISUAL DESIGN

In considering the third issue, unobtrusive design, we emphasize that the costs of wrong decisions in giving the glosses (Horvitz & Apacible, 2003) can be minimized through careful visual design. The human visual system is sensitive to changes in the visual field (Bartram et al., 2003). Studies on **change blindness** have shown that, in order to consciously perceive a change in the visual field, the observer’s focus of attention should be at the location where the change takes place (Simons & Rensink, 2005). On the other hand, motion (Franconeri & Simons, 2003) – especially onset motion – has been shown to attract the observer’s attention (Abrams & Christ, 2003; Franconeri & Simons, 2005). In applications performing actions proactively, we must take special care to avoid situations that needlessly distract from the user’s main task (in our case, reading).

10.4.1 Visual design decisions in iDict

In iDict, the gloss is shown right above the word or phrase that appears to be problematic. This action is designed to be as unobtrusive as possible, to avoid extra visual noise. Correspondingly, removal of the gloss is designed to occur imperceptibly, without needless flashing or flickering.

The user can specify how many glosses are to be visible at a time. If, for example, 10 glosses are chosen to be visible, the most recent one is displayed in black, but the nine preceding glosses fade to gray in time. Sometimes the reader may want to recheck a gloss that was provided earlier. Total time for the word does not accumulate while its gloss is visible, thus preventing needless and distracting redrawing.

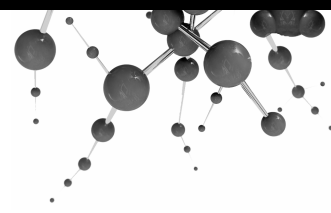
When reviewing the videos of their reading sessions, many of the test users reported that they did not notice the glosses that they did not expect. The change blindness we mentioned above and the fact that the glosses were designed to appear smoothly without visual noise together account for this. The location of the unnecessary gloss is often outside the focus of

the user's visual attention, since the reading has already continued beyond the point at which the system erroneously decided to take action. Additionally, Henderson and Hollingworth (1999) found that a change is more likely to go unnoticed if it occurs during a fixation orienting away from, rather than toward, the point of change.

There have also been studies indicating that change blindness is not affected by the focus of visual attention on its own; the relevance of the changed information to the task being performed also has a strong effect on the perception. Triesch, Ballard, Hayhoe and Sullivan (2003) found that a change in the appearance of an object, even if it is under visual focus, may go unnoticed if the changed attribute of the object is not relevant for performance of the primary task. Roda and Thomas (2006) make a similar deduction on the basis of their interpretation of Grossberg's Adaptive Resonance Theory (ART); they state that "intentions reflect expectations of events that may (or may not) occur" and "the user's attention will be focused on information that matches their momentary expectations." In the context of iDict and unnecessarily given glosses, that a user is not expecting a gloss to occur can easily account for the reports of users not perceiving needless glosses.

When giving the user visual feedback on eye movements, we have to be careful. The eyes are very fast, and also eye trackers sometimes erroneously report stray (often very short) fixations. For example, we noted that readers sometimes made fast visits to previous or succeeding lines and then resumed reading the initial line. Giving the user feedback right at the first stray fixation to indicate a change in current line, even if the reader really did make the fast visit instead of the tracker noting the fixation erroneously, gave the reader the impression of the interface being labile. Simply delaying the feedback on changed line at least until another fixation was targeted to the new line made a substantial difference. Such smoothing of feedback was used also for changing the dictionary entry in the dictionary frame. A single, short (under 150 ms) fixation does not yet activate a change of dictionary entry in the frame.

As our overall conclusion for this chapter, we can state that when one is using natural gaze as input for an application, many of the problems resulting from imperfect tracking and interpretation of gaze behavior can be tempered at a fundamental level with carefully designed interface design solutions.



11 Evaluation of iDict's Usability

The term “usability” is widely agreed (ISO 9241-11, 1998) to comprise three distinct aspects: effectiveness, efficiency, and satisfaction. Effectiveness is the accuracy and completeness with which users perform tasks using the system. Efficiency can be considered to be the relationship between effectiveness and the resources consumed in performing tasks. Satisfaction reflects the users’ subjective reactions and overall attitude to using the application. The aim of the evaluation of iDict’s usability was to take all three angles into account. To measure effectiveness, one should ideally use iDict in the real context for a long time. Such an extensive experiment is outside the scope of this thesis. Instead, we will measure how accurately the necessary help was given to the readers in our experiment. To measure efficiency and satisfaction, we settle on using subjective ratings provided by the test readers.

The first experiment was designed to measure how effectively readers get the help they need when assistance is triggered only by gaze via the threshold function developed in Chapter 9. If we were able to track the gaze point flawlessly, a user wouldn’t need the feedback of the tracking being performed. Since that is not the case, we wanted to know whether the feedback mode used has an effect on the accuracy of the assistance given. In the first experiment, we also asked the readers about their subjective experiences of the efficiency of iDict: whether the triggering sensitivity is appropriate and the application useful to them. We also asked about their preference from among different feedback modes.

Some of the applications described in Chapter 3, like Magic pointing (Zhai et al., 1999), combine the benefits of gaze and manual input. Also, in their experiment with EyeWindows, Fono and Vertegaal (2005) found that using gaze only for selecting the desired window and then letting the user

activate the window via a key press worked better than pure gaze interaction. The second evaluation experiment was designed to examine the effect on iDict's usability when the role of gaze was varied. The effectiveness of different input modes was considered. The participants were also asked for their subjective preference and their opinion of the efficiency of the different input styles used.

The first experiment and its results concerning iDict's measured effectiveness are analyzed in Section 11.1. The efficiency experienced in using iDict in the first experiment is reported upon in Section 11.2. The second experiment, comparing user satisfaction levels when gaze played different roles in the application, is covered in Section 11.3.

11.1 EFFECTIVENESS - ACCURACY IN GETTING THE EXPECTED HELP

In previous experiments, described in chapters 8 and 9, iDict was not operational; the readers didn't get assistance for the problematic words. In the experiment described next, we tested how the threshold function performed when iDict was fully operational.

Total time threshold th_l (the threshold used for low-frequency words) was set to 1000 ms, and the threshold for high-frequency words, th_h , was set to 1500 ms. Unfortunately, the experiment was designed on the basis of preliminary results from the experiment described in Chapter 9, which suggested that word frequency does filter out incorrect glosses but did not give us an answer concerning the proper value for the threshold for frequent words. We now know that using 2000 ms as the value for th_h would have been a better choice. According to the analysis completed in Chapter 9, replacing the value 1500 ms with 2000 ms would have reduced the percentage of incorrect glosses by about 0.5%.

11.1.1 Assumptions studied in the experiment

In preliminary tests, we had observed that the eye behavior changes – consciously or unconsciously – when the readers know to wait for a gloss to appear. That is why we assumed that now, with the users aware of iDict's operation, we would get substantially better triggering percentages for problematic words than the 29% we obtained by analyzing the data from the previous experiment (summarized in Section 9.3).

We also hoped that the percentage of false alarms would not rise to much higher than 1.4% of the familiar words (0.9% + 0.5% due to too low th_h) and were a little concerned about that; both checking and processing the automatically received dictionary glosses potentially cause delays and additional eye movements that could result in further false alarms. Also, the spatial accuracy of the eye tracker used in the experiment was lower than that of the eye tracker used in the previous experiment (EyeLink).

Usability considerations for an iDict-like application dictate that the application be used with a remote tracker, even at the expense of accuracy. That is why we used the iView X tracker for the experiment described in this section.

11.1.2 Experiment setup

The test was carried out in the same manner as in the previous experiment. As mentioned, the major difference was that iDict now really supplied the automatic dictionary glosses for the readers. The other, smaller differences between the test arrangements are explained below.

Participants

Six male and six female students, 12 participants in all (P1–P12), took part in the experiment. None of them had participated in the previous experiment, and none had used iDict before. Their ages varied from 20 to 41, the average being 25 years. Seven of them wore eyeglasses, and one used contact lenses.

Stimulus text and motivation

The text contained 641 words and was divided into three blocks. With small modifications, the text was the same as in the previous experiment. The layout was altered because in the previous experiment some of the participants complained that using the Times font type made reading from the screen unpleasant. Therefore, we switched to a sans-serif font type, Verdana. Since sans-serif fonts appear bigger in size, we reduced the font size to 11 pt. The texts were still displayed on a 19" screen with a resolution of 1024 x 768.

Procedure

At the beginning of the experiment, the participants were introduced to iDict and were allowed to practice its operation freely. After the rehearsal, the three text blocks were presented sequentially and the eye movement data for the reading of all three texts were recorded. During reading, the screen was now also recorded on video. As before, the problematic words were pointed out by the readers after each block was read, but, in addition, this time the video of the whole session was reviewed with the participant at the end of the experiment. This way, we hoped to get more reliable data on the points in the text that the participant regarded as problematic: the readers had an opportunity to review the glosses they got and to double-check whether the help received was expected or given needlessly.

Each of the texts was presented using a different feedback mode: for one text (A), no feedback was given (except the gloss); for another (B), the stabilized gaze cursor was used; and with the third (C), the line marker was present. The feedback modes were counterbalanced so that the order in

which they were used was fair to all modes. Subjective experiences of the performance of iDict and the preference from among the feedback modes used were sought after the whole session.

Apparatus

iView X was the eye tracker used in the experiments.

11.1.3 Results concerning triggering accuracy

From the three text blocks the participants pointed out a total of 310 problematic words, and of these they got help (a gloss for the word) for 281 words. In addition to the correctly triggered help, the participants got 178 false alarms. This means that in a real-life situation iDict triggered help for

- 91% of the problematic words (281 out of 310) and
- 2.4% of the possible false alarms (178 out of 7,382).

Table 11.1 summarizes the data for each participant. The first column (C1) of the table displays the participant's subjective assessment of his or her language skills (as a score on a scale of A to D). Column C2 presents the number of problematic words pointed out by each participant when the block was reviewed after reading and the material was double-checked from the video after the session.

Columns C3 and C4, respectively, contain the number of correctly triggered glosses and the number of false alarms (that is, the glosses the user regarded as needless). Columns C5 and C6 contain the same information as C3 and C4 do, but as percentages: the percentage of correctly triggered words from among the problematic words and the percentage of the familiar words that were falsely triggered.

Verifying the assumptions

The relatively small percentage (2.4%) of false alarms was a pleasant surprise; we had presumed that processing the glosses received during use of the application affects the reading path and increases the number of words that get accidentally triggered. To some extent that was true, but not on a larger scale.

The presumption that substantially more correctly triggered glosses would appear was right. Instead of the 29% obtained in the experiment in which we were developing the threshold function, the readers now got help in 91% of the situations when they wanted to get it. As presumed, the readers quickly learned to prolong their gaze in order to get the help; that clearly happened with many of the test readers. Actually, what is more interesting is what happened in the 9% of the cases when they did not get the desired help. A closer look at the data in Table 11.1 reveals some

11.1 Effectiveness – accuracy in getting the expected help

answers.

	C1 language skills	C2 problematic words	C3 correctly triggered	C4 false alarms	C5 correctly triggered	C6 false alarms
P1	A	13	6	8	46%	1.3%
P2	B	26	19	4	73%	0.7%
P3	A	11	9	3	82%	0.5%
P4	B	15	15	5	100%	0.8%
P5	D	115	112	52	97%	9.9%
P6	B	35	35	20	100%	3.3%
P7	B	18	18	17	100%	2.7%
P8	B	11	11	7	100%	1.1%
P9	C	20	12	19	60%	3.1%
P10	A	12	12	1	100%	0.2%
P11	B	21	20	36	95%	5.8%
P12	B	13	12	6	92%	1.0%
		310	281	178		

Table 11.1: Measured triggering accuracy in the experiment.

Explaining the results

iDict’s performance varied a lot between participants. For example, in the worst case, that of P1, the reader received help in only 46% of the cases in which she would have accepted a gloss. On the other hand, five of the participants (P4, P6, P7, P8, and P10) got help whenever it was needed.

The variance can be explained in part by the different strategies the participants adopted in reading. Some of them reported that they read as they would read normally – if the help did not show up, they put forth no additional effort to get it – whereas some of the others clearly wanted to get the help because they knew it was available. At the beginning of the test, all of the participants were informed of how iDict works. However, the test supervisor might have slightly encouraged use of the first strategy mentioned, by phrasing the task as, “After calibration, you can just start reading like you normally would.”

We suppose that some differences were caused by the inaccuracy of the measured point of gaze. If the calibration was off and translations were triggered for wrong words, some participants quickly learned either to “look off” and intentionally trigger the translation for the word they wanted or to correct the measured point of gaze with the arrow keys. Some participants did not bother to do that. If this explanation for the high percentages in column C5 were always true, a high value in that column should imply a positive correlation between columns C3 and C4 (i.e., for many of the correctly given glosses, there would also be preceding false alarms). This explanation does hold for many participants (e.g., P5, P7,

and P11) but not for all. For example, for P3, P4, P10, and P12, the measured point of gaze seems to have been quite accurate (a high percentage of correct hits but still only a few false alarms).

11.1.4 Feedback used and triggering accuracy

The above analysis of the triggering accuracy was performed on the data gathered for all of the text blocks read. However, each participant read the three blocks using a different feedback mode (no feedback, gaze cursor, and line marker). Did the feedback mode used affect the triggering accuracy?

As noted above, the average percentage of correctly triggered glosses when computed from the whole data set is 91%, and 2.4% of the words were incorrectly triggered. The results concerning correctly and incorrectly triggered glosses in the three sets of feedback conditions are given in Table 11.2.

	in numbers		as percentages	
	correctly triggered	false alarms	correctly triggered	false alarms
A (no feedback)	87 out of 99	76 out of 2465	88%	3.0%
B (gaze cursor)	108 out of 114	38 out of 2450	95%	1.6%
C (line marker)	86 out of 97	63 out of 2467	89%	2.6%

Table 11.2: The effect of different feedback modes on triggering accuracy.

The triggering accuracy was better when the user was provided with feedback, and, additionally, displaying a stabilized gaze cursor for the user resulted in better accuracy than displaying a line marker did. The same result is shown both with correctly triggered glosses and with the false alarms. The users seem to be better able to get the desired gloss when the feedback on the ongoing tracking is given with the gaze cursor. The feedback offers benefit for the reader also in terms of avoiding needless glosses, and the gaze cursor proved to be the most beneficial feedback mode.

When within-subject repeated measures oneway analysis of variance ANOVA was used to investigate the effect of visual feedback, it had no significant effect on the correctly triggered glosses. For the error percentages ANOVA showed a significant effect $F = 4.5$, $p < 0.05$. However, the pairwise post hoc Bonferroni corrected comparisons were not statistically significant.

11.1.5 Language skills and triggering accuracy

The participants' subjective assessments of their own English skills on a scale from A to D were given in Table 11.1 (C1). Analogously to the situation in the previous experiment, the participant (P5) who had worse

skills in English was easy to spot from the data. She got the highest proportion of false alarms (9.9%). One might assume that she was annoyed with the unnecessary glosses, but that was not true. She was one of the participants who reported that “I would definitely use the application if it were available.” In the next section, subjective experiences of using the application are described in greater depth.

11.2 EFFICIENCY – SUBJECTIVE EXPERIENCE OF IDICT PERFORMANCE

We mentioned in Section 11.1.2 that participants in the experiment were questioned after the experiment about their subjective experiences of iDict’s performance and their level of preference for the various feedback modes. The following three questions were assigned to the participants:

1. *How did you find the triggering sensitivity? Did you get a gloss (1) too easily, (2) at the right time, or (3) too slowly?*
2. *Did you find the application useful? (1) I would definitely use the application if it were available, (2) the application performed well enough for me to occasionally use it if it were available, or (3) the application did not perform well enough for me.*
3. *Which of the feedback modes did you like best? Put them in order of preference using the notation “A” for no feedback, “B” for gaze cursor, and “C” for line marker.*

11.2.1 Subjective experiences of triggering accuracy and iDict’s usefulness

The answers to the three questions are given in Table 11.3 (Q1–Q3). The language skill information is contained in this table also (C1), since considering it in relation to the answers given may in some cases be interesting.

When asked whether she felt that the glosses were triggered in a timely fashion (Q1), one participant (P1, who got help for only 46% of the problematic words) said she thought that the application reacted too slowly. She said she read normally and did not intentionally prolong the gaze on problematic words. One participant (P3) felt that the application was too eager in providing help. She was a student of English translation studies and felt that the reader should have more time to figure out the meaning of the word before the gloss is provided. Some of the participants reported that they actually had to delay their gaze somewhat consciously but felt that this did not bother them. They thought that increasing the sensitivity would have increased the number of false alarms, too. coded as “2+” in the table. Thus, 10 out of 12 participants felt that the sensitivity in recognizing the reader’s difficulties was good and they would not have wanted to change it in either direction.

	C1 language skills	Q1 triggering sensitivity	Q2 usefulness	Q3 feedback preference
P1	A	3	2	BCA
P2	B	2+	2	ACB
P3	A	1	2	CBA
P4	B	2	2	ACB
P5	D	2	1	BCA
P6	B	2	2	BAC
P7	B	2	2	BAC
P8	B	2	1	ACB
P9	C	2+	2	CBA
P10	A	2+	1	ACB
P11	B	2	2	CBA
P12	B	2	2	BAC

Table 11.3: Subjective opinions of iDict.

Especially for the readers who learned to take advantage of the application, it worked well. They did not think that they had to go to additional effort in order to receive the gloss. On the contrary, they felt that they could nicely control the appearance of a gloss. For some of the readers, the application performed extremely well. One of them reported after the test that “The application worked like a thought. Whenever I started to wonder about the meaning of some word, I got the gloss to tell me what it was. It was neat.” Some of the test readers reported that the unwanted glosses did not bother them, but, on the other hand, one of them (P3) did say, “I was afraid of spending so long on a sentence that the applications would start to give me assistance even if I didn’t want it.”

It was interesting to note that the two participants, P1 and P3, who would have wanted to change the triggering sensitivity – each in a different direction – both categorized their skills in English as “very good” (A). This affirms the assumption we made earlier, that the way the reader wants the application to react depends on personal preferences more than on reading skills. That participant P5, who assessed her skills as “poor” (D), would not have wanted to change the triggering sensitivity supports this observation.

As an answer to the second question (Q2), three of the participants reported that “I would definitely use the application if it were available.” The rest of the participants (nine of them) reported that “the application performed well enough for me to occasionally use it if it were available.” No one chose the third option, that “the application did not work well enough for me.”

It is likely that participants testing a new application are inclined to give answers that please the developers. Even with this possible bias, we can

conclude that the threshold function worked reasonably well in the application.

11.2.2 Preference for the different feedback modes

When the feedback modes were compared on the basis of their effect on the accuracy of triggered glosses, gaze cursor (mode B) performed better than the other modes.

However, when asked for preference from among the modes, the gaze cursor did not stand out clearly from the other modes.

The preferred feedback options (Q3) were distributed quite evenly among all modes (Figure 11.1). The spread of opinions about feedback

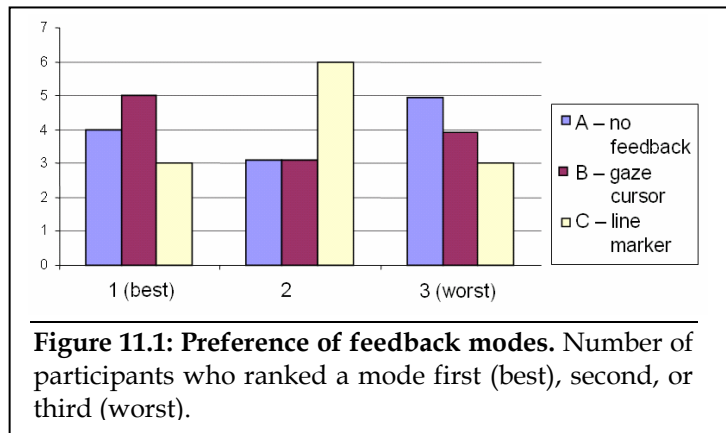


Figure 11.1: Preference of feedback modes. Number of participants who ranked a mode first (best), second, or third (worst).

modes does not entitle us to rank any of the feedback modes above others; clearly, what is the best feedback mode depends on the individual or the experiment was too short to allow the readers to form a firm opinion of the modes used. Nonetheless, the experiment proved that none of the modes clearly outshines the others where user experience is concerned.

11.3 SATISFACTION – COMPARING GAZE AND MANUAL INPUT

The main goal of the first evaluation experiment was to find out how accurately iDict is able to provide help for the reader when gaze alone is used to trigger the help function. The last experiment was set up to study how the new input modality compares with manual input; in previous studies, combined gaze and manual input were found to be successful. The second evaluation experiment, described in this section, compared the usability and user experience of iDict with the following input modes.

- (A) Mouse-only condition: gloss was triggered by mouseover event, and gaze was not used at all.
- (B) Combined condition: the word to be triggered was determined on the basis of the word with focus, but the gloss was triggered by a key press.
- (C) Gaze-only: gaze was used both for determining and for triggering the word (as was done in the previous experiment).

When the first condition applied, the user used the mouse to control which gloss was given. In order to design the condition to be similar to

gaze triggering, we set it up so that triggering was not performed with a mouse click but by moving the mouse cursor over the word for which the reader wanted to have a gloss. Similarly, the entire dictionary entry was displayed when the user moved the cursor into the dictionary frame.

With the second condition, gaze tracking was used for selecting the word for which the gloss was to be given, and the reader could choose to ask for the gloss by pressing the keyboard's spacebar. With this setup, the user should be aware of the word for which the gloss is to be given when he or she presses the spacebar. That is why, in this setup, the word with focus was indicated to the reader via a change in visual appearance (graying of the target word, as described in Subsection 10.2.1). To avoid distracting visual noise, the graying was performed only if the gaze was on the word for a prolonged time (> 1000 ms). We hoped that this would give the reader the impression that a prolonged gaze "pressed" a word active, such that a gloss could then be evoked for the active word by pressing of the spacebar.

For the third condition, we had to decide what kind of feedback to use in this experiment. Since none of the tested feedback modes was proven clearly superior, and since the feedback is actually needed only when the gloss given is not desired, we decided to implement one more feedback mode. In the third condition, the feedback of the tracked point of gaze was not given unless the user made manual correction indicating that the gloss was given for the wrong word. Vertical error (an incorrectly tracked line of reading) was much more common, and locating the line marker is easier than locating the small gaze cursor. That is why we chose to show the reader the line marker when corrective manual key presses were performed, to inform the reader of the line on which the tracked point of gaze was at the moment. After that, any horizontal error (more rare) was easy to spot because the erroneous gloss was probably given for a nearby word on the indicated line. The line marker disappeared when the reader ceased to make corrective key presses and continued reading.

The rest of the chapter outlines the experiment's setup and main results. A complete and detailed report on this experiment is given by Koskinen (2006).

11.3.1 Assumptions studied in the experiment

This time, there were three different input modes, which we wanted to compare. Even though the main motivation was to measure the subjective user experience in different conditions, also the triggering measurements (percentages of correctly triggered glosses and false alarms) were computed. This was done to compare the effectiveness of different conditions and also to verify the results of the previous experiments.

In comparing different conditions, the ideal is for the participants' experience with each of the measured conditions to be the same. However, in this case it was obvious that finding participants equally practiced with using mouse and gaze as an input method could not be found. That is why we assumed that the experiment would favor the mouse-only condition.

We were also interested in finding out whether the input mode affects the number of words the readers report to be problematic, and the number of correct and erroneous glosses received. In principle, the input mode should not affect the number of problematic words. If substantial difference between conditions was found, it would indicate that the experiment setup fails to identify the problematic words. One reason for that might be that in some conditions the readers are more reluctant to lean on the help available. On the other hand, giving the reader an active role through the use of the mouse could encourage a different behavior, where words are clicked to find their dictionary translations, even when the meaning is known to the reader. Thus it was difficult to predict what the experience would be in the different conditions.

11.3.2 Experiment setup

The experiment setup followed the procedure of the previous experiments.

Participants

There were 18 test readers participating in the experiment. They were students from the course Introduction to Interactive Technology. Their ages varied from 20 to 27 years, and nine were male and nine female. To ensure easy, better-quality calibration, the participants were selected to ensure that none of them used eyeglasses when using a computer. Also, their skills in English were assured to be decent through selection from among only potential participants who had at least good marks in English in their previous studies. Most of them (13) subjectively judged their skills in English to be good, four of them considered these skills excellent, and one assessed her skills as moderate. None of the participants were familiar with iDict. Only three of them had a little experience with eye tracking, having once participated in an experiment in which eye tracking had been used. So, all of them can be considered to be unpracticed with eye tracking.

Stimulus text and motivation

As before, the texts used in the experiment were excerpted from one short story¹ such that the plot continued from one text block to the next, creating

¹ This time, the short story selected was Arthur C. Clarke's "A Walk in the Dark."

motivation to find out what happens next. The motivation to comprehend the text was again increased with the obligation to give a verbal review of the text after each block.

Because this time the intention was to compare different input modes and all the users were obviously experienced mouse users, it was unavoidable that the mouse condition (A) was to have superior advantage in being a familiar style of interaction. Nonetheless, to ensure that the users understood the operation of the unfamiliar interaction modes, we this time divided the text into six blocks instead of three. For the first, third, and fifth block, the participant was allowed to rehearse the forthcoming input mode, and the second, fourth, and sixth blocks were the ones from which the data were gathered¹. We took care to balance the infrequent words and expressions in the different blocks, but, in addition to that, the conditions were counterbalanced to neutralize the possible effect of the order in which the input modes were used in relation to the text blocks. The text blocks contained 236, 228, and 216 words; thus, each participant read a total of 680 words while information was being recorded.

The blocks were all presented as 11-point Verdana text with 1.5 line spacing. The texts were displayed on a 17" screen with a resolution of 1024 x 768.

Procedure

After introducing a participant to the principal idea of iDict, we gave a brief explanation of the three input modes to be used. When gaze was used for the first time, the calibration was performed, and before starting the reading the calibration was confirmed with a test window². A recheck of calibration was performed also before the second gaze-aware condition.

After each condition, the participant gave a verbal review of the contents of the tracked block and pointed out the problematic words, as was done in the previous experiments. In order to measure the subjective assessment of usability for each of the input modes, we also asked the participant to fill in the SUS questionnaire form (the System Usability Scale; Brooke, 1996), which contained 10 statements concerning the usability of the tested conditions. At the end of the experiment, each participant was asked the order of preference of the three input conditions.

¹ In previous experiments, the participants practiced with the application only at the beginning of the experiment.

² In the test window, there were 12 points. The experimenter asked the reader to look at each of the points. If the measured point of gaze seemed to significantly differ from some of the points for focus, recalibration for that part of the screen was performed.

Apparatus

Tobii 1750 was the eye tracker used in the experiments.

11.3.2 Results for different input conditions

Below, we first take a look at the use of the SUS questionnaire and report the ratings given by users. Second, we review the results concerning the subjective preference of the different input conditions. Finally, we report how the effectiveness of different input modes in this experiment compared to what we found in the first evaluation experiment.

Subjective assessment with the SUS questionnaire

Instead of using a questionnaire containing questions we designed ourselves, we chose to use the SUS questionnaire, which is designed to cover a variety of aspects of system usability with only 10 questions presented for a test participant's consideration. The participants give their subjective opinion of how much they agree with the statement, by using a five-point scale ranging from "strongly disagree" to "strongly agree." The 10 statements can be seen in Table 11.4.

In addition, by assigning scores¹ 0 to 4 for the 10 statements, we can compute a composite SUS score ranging from 0 to 40. The SUS score reflects the overall usability of the system, or, in our case, of different conditions. The SUS score given for

- condition A (mouse-only) was 34.56, with a standard deviation of 4.5; for
- condition B (gaze and mouse) was 29.94, with a standard deviation of 4.8; and for
- condition C (gaze-only) was 29.89, with a standard deviation of 4.9.

Thus, the mouse-only condition received the best SUS score. That was an expected result, since using the mouse is the familiar means of interaction. The SUS score was about the same for the gaze-only condition and for the combined condition. The number of ratings given for each of the statements is itemized in Table 11.4.

If we look at the ratings by statement, the only statement for which the mouse-only condition was not ranked as best was statement 5. The gaze-only condition was considered the most successful in integrating the various functions of the system. When the SUS scores are viewed by test reader, 14 of their scorings were best for the mouse-only condition, three

¹ For the positive statements (S1, S3, S5, S7, and S9), the points given increase from 0 (strongly disagree) to 4 (strongly agree), and for the negative statements (the rest of the statements), the number of points given drops from 4 to 0, correspondingly.

were the same for gaze-only and mouse-only conditions, and the SUS score for the gaze-only condition was the best for one of the test readers.

The SUS score for the mouse-only condition was significantly better than that for the combined condition ($p < 0.001$) and also for the gaze-only condition ($p < 0.001$). However, the scores for all three conditions were over 20, meaning that the test readers experienced them all positively.

	strongly disagree	disagree	no opinion	agree	strongly agree
	S1. I think that I would like to use this system frequently.				
A	0	0	2	9	7
B	0	4	6	6	2
C	0	3	5	7	3
	S2. I found the system unnecessarily complex.				
A	12	5	1	0	0
B	8	7	3	0	0
C	5	10	2	1	0
	S3. I thought the system was easy to use.				
A	0	1	0	6	11
B	0	1	3	8	6
C	0	0	3	8	7
	S4. I'd probably need support of a technical person to be able to use this system.				
A	11	6	0	1	0
B	6	9	1	2	0
C	7	5	2	4	0
	S5. I found the various functions in this system well integrated.				
A	0	0	5	9	4
B	0	1	9	7	1
C	0	0	4	9	5
	S6. I thought there was too much inconsistency in this system.				
A	12	4	2	0	0
B	7	9	2	0	0
C	6	10	2	0	0
	S7. I'd imagine that most people would learn to use this system very quickly.				
A	0	0	0	8	10
B	0	0	0	10	8
C	0	0	2	10	6
	S8. I found the system very cumbersome to use.				
A	11	5	1	1	0
B	3	12	1	2	0
C	2	11	3	2	0
	S9. I felt very confident using the system.				
A	0	0	2	8	8
B	0	4	2	9	3
C	0	1	7	8	2
	S10. I needed to learn a lot of things before I could get going with this system.				
A	15	3	0	0	0
B	13	3	1	1	0
C	10	6	1	1	0

Table 11.4: SUS questionnaire results. The number of each of the ratings given for the SUS questionnaire statements (S1-S10) by the 18 test readers. The conditions were: A = mouse-only, B = combined, and C = gaze-only.

Since the SUS questionnaire results can favor the familiar condition, we wanted to also obtain a straightforward subjective opinion of the conditions from the test readers. That is why we asked them to rank the different input conditions depending on which one they would prefer to use.

Subjective input mode preferences and assessment of efficiency

This question as well indicated a preference for using the mouse (Figure 11.2). The mouse-only condition received the highest number of top rankings: eight of them. However, it is interesting that more than half of the participants did not rank the mouse-only condition first: 10 participants ranked either the combined or the gaze-only condition first. There was no difference between the gaze-only and combined gaze and mouse condition in this respect. This experiment supported the observation made in the previous tests. Even though inaccuracy in tracking the point of visual attention and interpreting the gaze behavior decreases the value of gaze-aware applications, some users had reported even in our previous experiments that they experienced gaze-aware interaction as enjoyable and very natural.

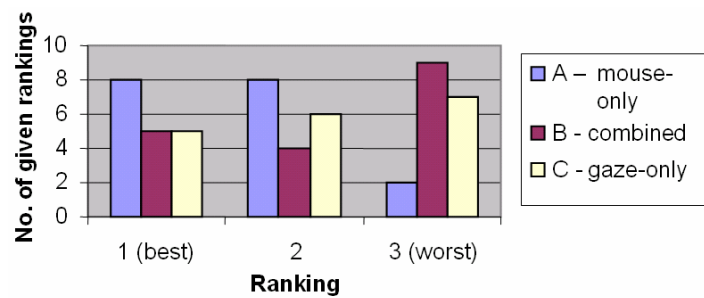


Figure 11.2: Preference of input conditions. Number of participants who ranked a condition first (best), second, or third

Efficiency of different input conditions

In order to verify the results reported in Section 11.1, we recorded the number of problematic words together with the number of correctly triggered glosses and false alarms, as was done previously for each condition.

First, we analyzed the time used for reading and the number of problematic words reported for each input condition. There was no significant difference between the conditions in either of them. The average reading times for conditions A, B, and C were 130 s, 130 s, and 139 s, respectively. The total numbers of words reported as problematic in each of the conditions were 111, 121, and 116.

Second, we computed the percentage of the glosses that were correctly and falsely triggered under each of the conditions. The original aim was to record and compute these values only for the gaze-only condition, to verify the earlier results for the gaze-only condition. However, to keep the test setup identical in all conditions, we decided to record them for all conditions. Comparing the measurements for the conditions ended up revealing interesting aspects of the used evaluation method. The percentages computed were the following.

In condition C (gaze-only):

- 86% of the problematic words were triggered, and
- 0.7% of the familiar words were triggered.

In condition B (combined):

- 74% of the problematic words were triggered, and
- 0.1% of the familiar words were triggered.

In condition A (mouse-only):

- 87% of the problematic words were triggered, and
- 0.2% of the familiar words were triggered.

Thus, in the gaze-only condition (C), the percentage of correctly triggered glosses was lower than in the previous experiment (86% instead of 91%). But, on the other hand, the percentage of false alarms was lower (0.7% instead of 2.4%) as well. This verifies that when the readers are aware of the principles of how the application makes use of gaze behavior, the magnitude of correctly triggered glosses is rather closer to 90% than the 30% achieved in the early experiments when we were honing the triggering threshold function. In this experiment, the number of false alarms was even lower than expected.

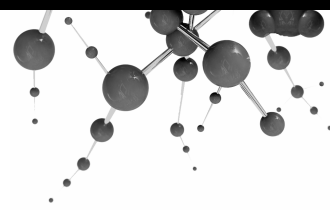
Examination of the efficiency measures for the combined condition B shows that the percentage of correctly triggered glosses is lower than for condition C - surprisingly so. One would expect that when the readers could initiate the gloss request by a key press, the percentage of correctly triggered glosses would be higher than when glosses were triggered automatically on the basis of the triggering threshold we developed. When interviewed after the experiment, five of the test readers said that they experienced the change in the visual appearance of the focused word in the combined gaze/mouse condition as disturbing. As explained before, in the combined input condition, the focused word turned gray whenever there was a prolonged gaze. This might have confused readers enough that they didn't ask for the gloss even though they were in need of it. We must keep in mind that all of the readers were new to gaze-aware systems

and that this experience could change with longer use of the application.

At first, the percentage of correctly triggered glosses in the mouse-only condition (A) was a big surprise. In fact, we had planned not to measure the triggering accuracy for the mouse-only setup at all. When the readers had a familiar way to request a gloss whenever in need of one, one would expect the triggering accuracy to be near 100%. The efficiency measurements for the mouse-only condition were included in the experiment only to keep all of the conditions as similar as possible. If a participant reports that a word was problematic, why not ask for a gloss for it via the familiar mouse interface? A plausible explanation is that even though not knowing the word the test reader did understand the sentence and did not bother to fetch help in translating the word.

The observation concerning non-triggered problematic words with the mouse-only condition allows us to interpret our efficiency measure, the percentage of correctly triggered words, in a milder way for gaze-aware conditions as well. When reading longer passages, the readers seem not to want translation for some of the problematic words that are not essential for comprehension of the sentence.

.....



12 Conclusions

Techniques facilitating the recording of eye movements have traditionally been used mainly in psychological and clinical research for studying perceptual and cognitive processes. Recently, studying the use of eye movements to enrich human-computer interaction has become an active field of research.

Even though gaze tracking provides the user of a gaze-aware application with a direct and effortless interface, the drawback is that such applications may be unpredictable. As noted in the introduction to this work, gaze tracking has proved its importance for special user groups. However, it could provide obvious benefits for standard users also. We believe that it would be possible to develop applications in which the benefits gained outweigh the required effort and the expense, which it is hoped will decrease owing to the inclusion of an eye tracking device as part of a standard computer installation. If we were able to achieve that breakthrough, it would eventually benefit all, both standard users and those who are restricted to gaze-based interaction due to physical limitations.

This work was undertaken to screen the prospects of including gaze tracking in standard user interfaces. We first analyzed gaze tracking from the biological, technological, and psychological perspectives. We found that three key issues are: **the accuracy of gaze tracking, the interpretation of gaze behavior, and the solutions used in designing the interface and the gaze-aware application as a whole.**

We studied how these three research problems could be handled in the context of one gaze-aware application, iDict, when tracking the progress of reading. However, many of the observations made are more general,

which we hope will make the contributions of this dissertation valuable for designers of gaze-aware applications generally.

12.1 TEMPERING THE GAZE TRACKING INACCURACY

We made the remark that the problems in automatic tracking of reading of text originate from three sources. Two of them, the measuring inaccuracy and the drift from calibration, may diminish as improvements are made in eye tracking technology. Since the third problem originates from the human visual system, gaze tracking will always be accompanied by inaccuracy. Thus, if we want to be able to work out the focus of visual attention with a greater than one- or two-degree visual angle's precision, we need some algorithmic compensation for the measured gaze paths. We developed such algorithms for tracking the progress of reading. The originality of the algorithms developed lies in their ability to track the progress of reading in real time and simultaneously correct the local inaccuracies in the measured focus of visual attention. We called these algorithms drift compensation algorithms, which gives a clear image of their purpose, even though a more accurate term might be "visual attention estimation algorithms."

Typical inaccuracies with two different eye trackers were studied by analyzing a large number of reading paths from different readers. The mapping of fixations in a reading path was designed to tolerate inaccuracy in those typical situations. This was done by using text object masks: we first assign each object (words, lines, and paragraphs) in the text being read a mask that encompasses this portion of text. Then the mapping of fixations is performed according to two principles. First, we developed algorithms that map the fixations to the target text objects on the basis of their masks. A search for the target text object is performed to efficiently find the most probable target when the history of reading thus far is known. Second, to allow for the inaccurately measured focus of visual attention during reading, the masks used in the mapping are not static; their sizes and locations are dynamically modified with algorithms we developed, which we called the "sticky lines" and "magnetic lines" algorithms.

The sticky lines algorithm compensates for the ascending and descending reading paths by dynamically resizing the currently read line's mask. The line mask is enlarged to cover the line spaces above and below, and the mask is further enlarged as long as reading of the current line appears to continue.

The magnetic lines algorithm dynamically affects the location of a line mask. The relocation is performed on the basis of a return sweep. In order to be able to identify them as reliably as possible, we analyzed the reading

paths in situations where the reader transfers the gaze from one line to the next, and we developed a function that is able to signal in real time the occurrence of such events.

Both of the algorithms correct the observed inaccuracies locally. The sticky lines approach affects only the size of the currently read line, and the repositioning of line mask done by magnetic lines spreads the correction cautiously, with the correction gradually spread only to the neighboring lines.

Testing out these algorithms proved that they improved the tracking of the reading process to the limit of frequently used text sizes (11–12-point font size with 1.5 line spacing). The magnitude of the improvement measured was that instead of 55%, as much as 85% of the fixations were correctly mapped to the line being read. We must address the issue that, even though the automatic drift compensation algorithms were able to improve the accuracy of interpretation of the reading process, they are not able to flawlessly interpret it with commonly used text sizes. In the context of iDict, there is the option of providing the user with the possibility of manually correcting the errors in the automatic interpretation of gaze behavior. In applications where the user gets some kind of feedback indicating the gaze tracking being performed, an ability to explicitly and effortlessly make manual corrections to erroneously performed mappings enhances the effect of the drift compensation algorithms.

12.2 INTERPRETATION OF GAZE PATHS

In developing a gaze-aware application, it is important that the functions triggered by the user's eye behavior be performed in a timely fashion. In particular, the user gets frustrated easily if the normal eye behavior initiates actions too eagerly. On the other hand, an interface that is too passive may lose the benefits that could be achieved by tracking the user's eyes, and the user may have to expend too much extra effort in getting the application to react.

In our case, we used eye behavior to determine the points of the reader's comprehension difficulties as accurately as possible. We analyzed a set of eye movement measures that previous research results indicate may reflect difficulties in comprehension. We found out that when the reading path is tracked during the use of a gaze-aware application, cumulative fixation duration (total time) for a word was the most effective of the measures.

On the basis of the experiments, we defined a threshold function that, for each word, sets a threshold for the total time after which a gloss for the word is to be given to the reader. Using the constant factors derived by

optimizing the accuracy of the glosses triggered, the function setting the threshold for the word w_k was reduced to the format

$$th(w_k) = 2000 \text{ ms} - 0.17(\text{freq}(w_k)) \text{ ms},$$

when $100 < \text{freq}(w_k) < 6000$.

where $\text{freq}(w_k)$ is the number of the word in the list when the words are ordered according to their frequency (starting from order number 1 for the most frequent word, “the”).

In our experiments, we found slight indications that personalizing the threshold could make the glosses given more accurately. However, in homogenous conditions like those in our tests, where the text being read was common prose and the readers had relatively good skills in English, personalizing the threshold made no substantial difference.

The complete form of the function (presented in Section 9.4) gives us the option of changing the thresholds to account for a reader’s personal reading behavior just by replacing the mean total time measurements μ_i^F and σ_i^F (mean total time and standard deviation of total time) with the corresponding personal measurements $\overline{t(w)}$ and s_d . This could be done by analyzing the reading path for an example text or maybe even in real time, during the reading. If it could be done on the basis of the current reading behavior, that would take into account the complexity of the text being read as well.

Having said that, we must add that we are not very convinced by the idea of automatically adjusting the application’s sensitivity to eye movements. It can be assumed that it might confuse the user if the application’s behavior is automatically adjusted. For example, a study concerning eye-typing dwell times yielded similar observations (Majaranta, Aula & R  ih  , 2004). That is why we did not delve more deeply into the matter in this work. Giving the user an explicit option of adjusting the sensitivity seems to be a better choice.

12.3 DESIGNING GAZE-AWARE APPLICATIONS

One essential characteristic of eye movements that sets them apart from other input modalities is that the eyes are “always on.” When using natural eye movements as input for an application, we should consider also the situations in which the user is not concentrating on performing the task. The application should not end up performing counterproductive actions as a result of unexpected eye behavior. In applications like iDict, where eye input is used for triggering auxiliary features and not performing any irreversible operations, this is not a problem. One of our findings on using natural eye movements as input for applications was that we should not assume that the eye movements are totally natural when the user knows that the application utilizes eye input. We think the fact that the user learns to manipulate the application with eye movements is a positive rather than a negative thing, but it is important for a developer of a gaze-aware application to be aware of this behavior in order to work with, rather than against, its effects.

In order to be usable, a good design should find the appropriate level of proactivity and transparency for the application. We categorized our observations concerning the design of gaze-aware applications as relating to three main principles.

First, even in transparent interfaces, the system state should be visible. The user should be provided with appropriate feedback. The user should understand why the application performs some actions automatically. This is especially important in gaze-aware applications, where occasional misinterpretations are unavoidable. Moreover, if the user understands the principles according to which the proactive actions occur, occasional mistakes are much easier to accept. Further, the form of feedback provided was found to be essential. The rough feedback of the measured gaze behavior is too erratic and disturbing. The feedback should be filtered in such a way that it consolidates the user’s conception of the background interpretation of the gaze behavior.

Second, even non-command interfaces should be controllable. The user should always have the feeling of being in charge. The ideal balance between automatically taken and user-directed actions is a matter of personal preference, so the user should always be able to tune the sensitivity of the system.

Third, by means of visual design decisions, we can affect how the user perceives the automatically triggered events. The proactive actions should be designed so that potentially needless, unwanted actions do not distract from the primary tasks of the user.

12.4 CONCLUDING REMARKS

In this dissertation, we studied the potential for using natural gaze behavior to make human-computer interaction more efficient and also more pleasing for the user. The main problems in using gaze input are the inaccuracy of the measured point of visual attention and the problems in interpreting the semantics of the measured gaze behavior in real time. The first of the two problems makes the second even harder to handle: rendering the semantics of the gaze behavior must be performed using the imperfectly measured gaze paths.

In our test-bed application (iDict), the core task was to be able to track the line of reading on the basis of eye movements. A general result of this work is that reliable tracking of reading of a text displayed in font sizes typically used in electronic documents is outside the limits of the accuracy we will ever be able to achieve with gaze tracking. Due to limitations of the human visual system, this holds regardless of the constant improvements in technology.

However, this work proves also that with algorithmic solutions we can sharpen the tracking of gaze beyond the limits of the human visual system. Moreover, we found that with proper interface design we are able to make the interaction more pleasing, even if the interpretation of the gaze behavior is not accurate.

We developed dynamic drift compensation algorithms to extrapolate a reader's gaze paths, and we applied the design principles deduced from the experiments we performed. The evaluation of the test-bed application had encouraging results. We experimented to determine how the users experienced the gaze-aware application compared to using the same application with the mouse only. Even though the mouse had the clear advantage of being familiar to all, users' opinions did diverge, with more than half of them (10 out of 18) finding the performance of the gaze-aware application so pleasing that they would prefer using gaze tracking over using the mouse. The evaluation experiment also included comparison to determine whether users preferred to use gaze to only implicitly focus a target object, with the option of triggering the actions left to be done explicitly by mouse. The user experiences of using merely gaze and the combined gaze-mouse interaction were judged to be about equal. The users who preferred using the gaze-only approach found the manual request to involve needless effort, since the action could be triggered with a prolonged gaze anyway.

Work studying the benefits of using natural eye movements in human/computer interaction is still in its beginning stages. In our test bed application, we could assume that the user was actually reading during the interaction with the application. In a wider context, we could presume

that a user of any application would profit from the application's ability to track the user's line of reading. For example, when we detect that a user of a Web browser is reading some parts of a Web page, tracking of reading could take place and we could assist the user with the task accordingly. In that case, the knowledge generated on identifying occasions when reading is taking place (Campbell & Maglio, 2001; Kollmorgen & Holmqvist, 2006) would nicely supplement the results of this work.

One of the basic presumptions for this work was that we study **natural** gaze behavior. The experiments were performed with the technology currently available, but we pointed out that the problems in tracking natural gaze behavior will not be resolved with improved technology. On the other hand, it was also noted that there is a difference in the accuracy of measuring the natural and intentional point of visual attention.

This consideration leaves room for further research. In addition to putting additional effort into interpreting gaze behavior in non-ideal conditions, subsequent research should also clarify whether the technology is able to utilize the fact that we are probably able to focus the gaze more accurately than when naturally perceiving the presented stimulus. Previous research (e.g., Yarbus, 1967) justifies presuming that the accuracy of intentional focus of attention is substantially greater than that of the natural focus of attention. In particular, command-based interfaces would benefit from the technology being able to distinguish the difference. Developments in the precise measurement of intentionally positioned gaze are important for interfaces using natural point of gaze, too. In order to ensure accurate real-time tracking of reading, we had to provide the ability for the reader to correct the measured focus of visual attention manually. We also observed that, when aware of the use of the gaze input, the user quickly learns to take advantage of it. If the eye tracking system used were able to track the intentionally positioned gaze accurately enough, intentionally prolonged gaze could take the place of the manual correction.



.....

References

- Abrams, R. A., & Christ, S. E. (2003). Motion onsets captures attention. *Psychological Science*, **14**, 427-432.
- Abrams, R. A., & Jonides, J. (1988). Programming saccadic eye movements. *Journal of Experimental Psychology: Human Perception and Performance*, **14**, 428-443.
- Adamczyk, P. D., & Bailey, B. P. (2004) If not now, when? The effects of interruption at different moments within task execution. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM Press, 271-278.
- Amir, A., Flickner, M., & Koons, D. (2002). Theory for calibration-free eye gaze tracking. IBM Research Division, Almaden Research Center. (Technical Report RJ10275 (A0212-006)).
- Ashmore, M., Duchowski, A. T., & Shoemaker, G. (2005). Efficient eye pointing with a fisheye lens. In *Proceedings of Graphical Interfaces 2005 (GI '05)*. Canadian Human-Computer Communications Society, 203-210.
- Bailey, B. P., Konstan, J. A., & Carlis, J. V. (2001). The effects of interruptions on task performance, annoyance, and anxiety in the user interface. In *Proceedings of the International Conference on Human-Computer Interaction (INTERACT'01)*. IOS Press, 593-601.
- Balota, D. A., Pollatsek, A., & Rayner, K. (1985). The interaction of contextual constraints and parafoveal visual information in reading. *Cognitive Psychology*, **7**, 346-390.
- Bartram, L., Ware, C., & Calvert, T. (2003). Moticons: detection, distraction and task. *International Journal of Human-Computer Studies*, **58**, 515-545.
- Bates, R., & Istance, H. (2002). Zooming interfaces! Enhancing the performance of eye controlled pointing devices. In *Proceedings of the Fifth International ACM SICACCESS Conference on Assistive Technologies (ASSETS '02)*. ACM Press, 119-126.
- Bates, R., & Istance, H. O. (2003). Why are eye mice unpopular? A detailed comparison of head and eye controlled assistive technology pointing devices. *Universal Access in the Information Society*, **2**, 280-290.

- Baudisch, P., DeCarlo, D., Duchowski, A. T., & Geisler, W. S. (2003). Focusing on the essential: considering attention in display design. *Communications of the ACM*, **46** (3), 60-66.
- Bavelas, J. B., Coates, L., & Johnson, T. (2002). Listener responses as a collaborative process: The role of gaze. *Journal of Communication*, **52**, 566-580.
- Bellotti, V., Back, M., Edwards, W. K., Grinter, R. E., Henderson, A., & Lopes, C. (2002). Making sense of sensing systems: Five questions for designers and researchers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM Press, 415- 422.
- Beymer, D., & Russell, D. (2005). WebGazeAnalyzer: A system for analyzing web reading behavior using eye gaze. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*. ACM Press, 1913-1916.
- BNC. (2005). British National Corpus. Oxford University Computing Services, at <http://www.natcorp.ox.ac.uk/> (25.4.2006)
- Bolt, R. A. (1980). "Put-that-there": Voice and gesture at the graphics interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '80)*. ACM Press, 262-270.
- Bolt, R. A. (1981). Gaze-orchestrated dynamic windows. In *Proceedings of the 8th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '81)*. ACM Press, 109-119.
- Bolt, R. A. (1985). Conversing with computers. *Technology Review*, **88** (2), 35-43.
- Brooke, J. (1996). SUS: A 'Quick and Dirty' usability scale. In Jordan, P. W., Thomas, B., Weerdmeester, B. A., & McClelland, I. L. (Eds.) *Usability Evaluation in Industry*. London: Taylor & Francis, 189-194.
- Brysbart, M., & Vitu, F. (1998). Word skipping: Implications for theories of eye movement control in reading. In Underwood, G. (Ed.) *Eye Guidance in Reading and Scene Perception*. Amsterdam: Elsevier, 125-147.
- Campbell, C. S., & Maglio, P. P. (2001). A robust algorithm for reading detection. In *Proceedings of the 2001 Workshop on Perceptive User Interfaces*. ACM Press, 1-7.
- Cheng, D., & Vertegaal, R. (2004). Using mental load for managing interruptions in physiologically attentive user interfaces. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*. ACM Press, 1513-1516.

- Cheng, D., & Vertegaal, R. (2004). An eye for an eye: a performance evaluation comparison of the LC technologies and Tobii eye trackers. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '04)*. ACM Press, 61.
- Clifton, C. Jr., Bock, J., & Radó, J. (2000). Effects of the focus particle *only* and intrinsic contrast on comprehension of reduced relative clauses. In Kennedy, A., Radach, R., Heller, D., & Pynte, J. (Eds.) *Reading as a Perceptual Process*. Oxford: Elsevier, 591-619.
- COGAIN. (2004). Communication by Gaze Interaction. EU supported IST Network of Excellence. Network web pages available at http://www.cogain.org/about_cogain (5.3.2006).
- Collewijn, H. (1998). Eye movement recording. In Carpenter, R. H. S., & Robson, J. G. (Eds.) *Vision Research. A Practical Guide to Laboratory Methods*. New York: Oxford University Press, 245-285.
- Coren, S., Ward, L. M., & Enns, J. T. (1999). *Sensation and Perception*. New York: Harcourt Brace.
- Corno, F., Farinetti, L., & Signorile, I. (2002). A cost-effective solution for eye-gaze assistive technology. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME 2002, vol 2)*. IEEE Computer Society Press, 433-436.
- Corno, F., & Garbo, A. (2005). Multiple low-cost cameras for effective head and gaze tracking. In *Proceedings of HCI International 2005*. Erlbaum.
- Cutrell, E., Czerwinski, M., & Horvitz, E. (2001). Notification, disruption, and memory: Effects of messaging interruptions on memory and performance. In *Proceedings of the International Conference on Human-Computer Interaction (INTERACT'01)*. IOS Press, 263-269.
- van Dam, A. (1997). Post-WIMP user interfaces. *Communications of the ACM*, **40** (2), 63-67.
- De Valois, R. L., & De Valois, K. K. (1990). *Spatial Vision*. New York: Oxford University Press.
- Deutch, J. A., & Deutch, D. (1966). *Physiological Psychology*. Homewood: Dorsey Press.
- Dickie, C., Vertegaal, R., Shell, J. S., Sohn, C., Cheng, D., & Aoudeh, O. (2004). Eye contact sensing glasses for attention-sensitive wearable video blogging. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*. ACM Press, 769-770.
- Dickie, C., Vertegaal, R., Fono, D., Sohn, C., Chen, D., Cheng, D., Shell, J. S., & Aoudeh, O. (2004). Augmenting and sharing memory with

- eyeBlog. In *Proceedings of the 1st ACM Workshop on Continuous Archival and Retrieval of Personal Experiences*. ACM Press, 105-109.
- Donegan, M., Oosthuizen, L., Bates, R., Daunys, G., Hansen, J. P., Joos, M., Majaranta, P., & Signorile, I. (2005). User requirements report with observations of difficulties users are experiencing. Communication by Gaze Interaction (COGAIN), IST-2003-511598: Deliverable 3.1. Available through COGAIN pages at <http://www.cogain.org/> (5.3.2006)
- Duchowski, A. T. (2002). A breadth-first survey of eye tracking applications. *Behavior Research Methods, Instruments, and Computers*, **34**, 455-470.
- Duchowski, A. T. (2003). *Eye Tracking Methodology: Theory and Practice*. London: Springer.
- Duchowski, A. T., Cournia, N., & Murphy, H. (2004). Gaze-contingent displays: A review. *CyberPsychology & Behavior*, **7**, 621-634.
- Duncan, J. (1984). Selective attention and the organization of visual information. *Journal of Experimental Psychology*, **113**, 501-17.
- Eaddy, M., Blaskó, G., Babcock, J., & Feiner, S. (2004). My own private kiosk: Privacy-preserving public displays. In *Proceedings of the Eighth IEEE International Symposium on Wearable Computers (ISWC 2004)*. IEEE Computer Society Press, 132-135.
- Ebisawa, Y. (1995). Unconstrained pupil detection technique using two light sources and the image difference method. In Brebbia, C. A., & Hernandez, S. (Eds.) *Visualization and Intelligent Design in Engineering and Architecture II*. WIT Press, 79-89.
- Eurostat (2004). *Work and health in the European Union. A statistical portrait*. 2003 edition. Luxembourg Office for Official Publications of the European Communities, Luxembourg, available at http://epp.eurostat.cec.eu.int/cache/ITY_OFFPUB/KS-57-04-807/EN/KS-57-04-807-EN.PDF (5.3.2006).
- Fono, D., & Vertegaal, R. (2005). EyeWindows: evaluation of eye-controlled zooming windows for focus selection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. ACM Press, 151-160.
- Franconeri, S. L., & Simons, D. J. (2003). Moving and looming stimuli capture attention. *Perception & Psychophysics*, **65**, 999-101.
- Franconeri, S. L., & Simons, D. J. (2005). The dynamic events that capture visual attention: A reply to Abrams & Christ. *Perception & Psychophysics*, **67**, 962-966.

- Frazier, L., & Rayner, K. (1982). Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive Psychology*, **14**, 178-210.
- Frizer, F., Droege, D., & Paulus, D. (2005). Gaze tracking with inexpensive cameras. In *Proceedings of the 1st Conference on Communication by Gaze Interaction (COGAIN 2005)*, 10-11.
- Gemmell, J., Toyama, K., Zitnick, L. C., Kang, T., & Seitz, S. (2000). Gaze awareness for video-conferencing: A software approach. *IEEE Multimedia*, **7** (4), 26-35.
- Gregory, R. L. (1997). *Eye and Brain. The Psychology of Seeing*. 5th edition. New York: Oxford University Press.
- Groner, R., & Groner, M. T. (1989). Attention and eye movement control: An overview. *European Archives of Psychiatry and Neurological Sciences*, **239** (1), 9-16.
- Haber, R. N., & Hershenson, M. (1973). *The Psychology of Visual Perception*. New York: Holt, Rinehart and Winston.
- Hansen, D. W., & Pece, E. C. (2005). Eye tracking in the wild. *Computer Vision and Image Understanding*, **98** (1), 155-181.
- Hansen, J. P., Hansen, D. W., Johansen, A. S., & Elvesjö, J. (2005) Mainstreaming gaze interaction towards a mass market for the benefit of all. In *Proceedings of HCI International 2005*. Erlbaum.
- Haritaoglu, I., Cozzi, A., Koons, D., Flickner, M., Zotkin, D., & Yacoop, Y. (2001). Attentive toys. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME2001)*. IEEE Computer Society Press, 1124-1127.
- Henderson, J., & Hollingworth, A. (1999). The role of fixation position in detecting scene changes across saccades. *Psychological Science*, **10**, 438-443.
- Henderson, J. M., & Ferreira, F. (1990). The effects of foveal difficulty on the perceptual span in reading: Implications for attention and eye movement control. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, **16**, 417-429.
- Hoanca, B., & Mock, K. (2006). Secure graphical password system for high traffic public areas. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '06)*. ACM Press, 35.
- Hornof, A., & Halverson, T. (2002). Cleaning up systematic error in eye tracking data by using required fixation locations. *Behavior Research Methods, Instruments, & Computers*, **34**, 592-604.

- Horvitz, E., & Apacible, J. (2003). Learning and reasoning about interruption. In *Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI '03)*, 20-27.
- Howell, D. C. (1987). *Statistical Methods for Psychology*. Boston: Duxbury Press.
- Hutchinson, T. E., White, K. P. Jr., Martin, W. N., Reichert, K. C., & Frey, L. A. (1989). Human-computer interaction using eye-gaze input. *IEEE Transactions on Systems, Man and Cybernetics*, **19**, 1527-1534.
- Hyönä, J. (1995). Silmänliikkeet, kognitio ja lukeminen. *Psykologia*, **30**, 89-95.
- Hyönä, J., Tommola, J., & Alaja, A.-M. (1995). Pupil dilation as a measure of processing load in simultaneous interpretation and other language tasks. *The Quarterly Journal of Experimental Psychology*, **48**, 598-612.
- Hyrskykari, A. (2003). Detection of comprehension difficulties in reading on the basis of word frequencies and eye movement data. Abstract in the *12th European Conference on Eye Movements (ECEM12)*, Dundee, Scotland.
- Hyrskykari, A. (2006). Utilizing eye movements: Overcoming inaccuracy while tracking the focus of attention during reading. *Computers in Human Behavior*, **22**, 657-671.
- Hyrskykari, A., Majaranta, P., Aaltonen, A., & Räihä, K.-J. (2000). Design issues of iDict: A gaze-assisted translation aid. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '00)*. ACM Press, 9-14.
- Hyrskykari, A., Majaranta, P., & Räihä, K.-J. (2003). Proactive response to eye movements. In *Proceedings of the International Conference on Human-Computer Interaction (INTERACT'03)*. IOS Press, 129-136.
- Hyrskykari, A., Majaranta, P., & Räihä, K.-J. (2005). From gaze control to attentive interfaces. In *Proceedings of HCI International 2005*. Erlbaum.
- Inhoff, A. W. (1984). Two stages of word processing during eye fixations in the reading of prose. *Journal of Verbal Learning and Verbal Behaviour*, **23**, 612-624.
- Inhoff, A.W., & Rayner, K. (1986). Parafoveal word processing during eye fixations in reading: Effects of word frequency. *Perception and Psychophysics*, **40**, 431-439.
- Ishii, H. (2004). Bottles: A transparent interface as a tribute to Mark Weiser. *IEICE Transactions on Information and Systems*, **E87-D**, 1299-1311.

- ISO 9241-11. (1998). *Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability.*
- Jacob, R. J. K. (1991). The use of eye movements in human-computer interaction techniques: What you look at is what you get. *ACM Transactions on Information Systems*, **9**, 152-169.
- Jacob, R. J. K. (1993). Eye-movement-based human-computer interaction techniques: Toward non-command interfaces. In Hartson, H. R., & Hix, D. (Eds.) *Advances in Human-Computer Interaction*. Ablex Publishing, 151-190.
- Jacob, R. J. K. (1995). Eye tracking in advanced interface design. In Barfield, W., and Furness, T. A. (Eds.) *Virtual Environments and Advanced Interface Design*. New York: Oxford University Press, 258-288.
- Jacob, R. J. K., & Karn, K. S. (2003). Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In Radach, R., Hyönä, J., & Deubel, H. (Eds.) *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*. Oxford: Elsevier Science, 573-605.
- Jerald, J., & Daily, M. (2002). Eye gaze correction for videoconferencing. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '02)*. ACM Press, 77-81.
- Just, M., & Carpenter, P. (1980). A theory of reading: From eye fixations to comprehension. *Psychological Review*, **87**, 329-354.
- Kalat J. W. (1984). *Biological Psychology*. Belmont: Wadsworth.
- Kaur, M., Tremaine, M., Huang, N., Wilder, J., Gacovski, Z., Flippo, F., & Mantravadi, C. S. (2003). Where is "it"? Event synchronization in gaze-speech input systems. In *Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI '03)*. ACM Press, 151-158.
- Kembel, J. A. (2003). Reciprocal eye contact as an interaction technique. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*. ACM Press, 952-953.
- Khiat, A., Matsumoto, Y., & Ogasawara, T. (2004a). In *Proceedings of the 12th IEEE Workshop on Robot and Human Communication (RO-MAN 2003)*. IEEE Computer Society Press, 121-127.
- Khiat, A., Matsumoto, Y., & Ogasawara, T. (2004b). Task specific eye movements understanding for a gaze-sensitive dictionary. In *Proceedings of the 9th International Conference on Intelligent User Interface (IUI 04)*. ACM Press, 265-267.

- Kollmorgen, S., & Holmqvist, K. (2006). Automatic detection of reading. A presentation in the Scandinavian Workshop on Applied Eye-tracking, Lund.
- Koons, D., & Flickner, M. (2003). PONG: The attentive robot. *Communications of the ACM*, **46** (3), 50.
- Koskinen, D. (2006). Comparison of three different input modes in a gaze-aware reading aid environment (in Finnish). Unpublished manuscript for Master's Thesis. Department of Computer Sciences, University of Tampere.
- Kowler, E. (1990). The role of visual and cognitive processes in the control of eye movement. In Kowler, E. (Ed.) *Eye Movements and Their Role in Visual and Cognitive Processes*. Amsterdam: Elsevier, 1-70.
- Lankford, C. (2000). Effective eye-gaze input into Windows. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '00)*. ACM Press, 23-27.
- Latorella, K. A. (1996). Investigating interruptions: An example from the flightdeck. In *Proceedings of the Human Factors and Ergonomics Society 40th Annual Meeting*. HFES, 249-253.
- LC Technologies. (2005). The Eyegaze Communication System. Fairfax, VA, <http://www.eyegaze.com/> (5.3.2006)
- Lieberman, H. A., & Selker, T. (2000). Out of context: Computer systems that adapt to, and learn from, context. *IBM Systems Journal*, **39**, 617-631.
- Liversedge, S. P., Paterson, K. B., & Pickering, M. J. (1998). Eye movements and measures of reading time. In Underwood, G. (Ed.) *Eye Guidance in Reading and Scene Perception*. Amsterdam: Elsevier, 55-75.
- Lukander, K. (2004). Measuring gaze point on handheld mobile devices. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*. ACM Press, 1556.
- Maglio, P. P., Barrett, R., Campbell, C. S., & Selker, T. (2000). SUITOR: an attentive information system. In *Proceedings of the 5th International Conference on Intelligent User Interfaces (IUI '00)*. ACM Press, 169-176.
- Maglio, P. P., & Campbell, C. S. (2003). Attentive agents. *Communications of the ACM*, **46** (3), 47-51.
- Maglio, P. P., Matlock, T., Campbell, C. S., Zhai, S., & Smith, B. A. (2000). Gaze and speech in attentive user interfaces. In *Proceedings of the Third International Conference on Advances in Multimodal Interfaces (ICMI 2000)*. Springer, 1-7.

- Majaranta, P., Aula, A., & R ih a, K.-J. (2004). Effects of feedback on eye typing with a short dwell time. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '04)*. ACM Press, 139-146.
- Majaranta, P., & R ih a, K.-J. (2002). Twenty years of eye typing: systems and design issues. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '02)*. ACM Press, 15-22.
- McCrickard, D. S., Czerwinski, M., & Bartram, L. (2003). Introduction: design and evaluation of notification user interfaces. *International Journal of Human-Computer Studies*, **58**, 509-514.
- Morrison, R. E. (1984). Manipulation of stimulus onset delay in reading: Evidence for parallel programming of saccades. *Journal of Experimental Psychology: Human Perception and Performance*, **10**, 667-682.
- Morimoto, C., Amir, A., & Flickner, M. (2002). Detecting eye position and gaze from a single camera and 2 light sources. In *Proceedings of the 16th International Conference on Pattern Recognition (ICPR '02)*. International Association for Pattern Recognition, 314-317.
- Morimoto, C., Koons, D., Amir, A., & Flickner, M. (2000). Pupil detection and tracking using multiple light sources. *Image and Vision Computing*, **18** (4), 331-335.
- Morimoto, C., Koons, D., Amir, A., Flickner, M., & Zhai, S. (1999). Keeping an eye for HCI. In *Proceedings of the XII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI '99)*. IEEE Press, 171-176.
- Nguyen, K., Wagner, C., Koons, D., & Flickner, M. (2002). Differences in the infrared bright pupil response of human eyes. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '02)*. ACM Press, 133-138.
- Nielsen, J. (1993). Noncommand user interfaces. *Communications of the ACM*, **36** (4), 82-99.
- Oh, A., Fox, H., van Kleek, M., Adler, A., Gajos, K., Morency, L.-P., & Darrell, T. (2002). Evaluating look-to-talk: A gaze-aware interface in a collaborative environment. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems*. ACM Press, 650-651.
- Ohno, T. (1998). Features of eye gaze interface for selection tasks. In *Proceedings of the Third Asia Pacific Computer Human Interaction (APCHI '98)*. IEEE Computer Society, 176-181.

- Ohno, T. (2004). EyePrint: support of document browsing with eye gaze trace. In *Proceedings of the 6th International Conference on Multimodal Interfaces (ICMI '04)*. ACM Press, 16-23.
- Ohno, T., & Mukawa, N. (2003). Gaze-based interaction for anyone, anytime. In *Proceedings of the HCI International 2003*. Erlbaum, 1452-1456.
- Ohno, T., Mukawa, N., & Yoshikawa, A. (2002). FreeGaze: a gaze tracking system for everyday gaze interaction. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '02)*. ACM Press, 125-132.
- O'Regan, J. K. (1990). Eye movements and reading. In Kowler, E. (Ed.) *Eye Movements and Their Role in Visual and Cognitive Processes*. Amsterdam: Elsevier, 395-453.
- O'Regan, K. (1981). The "convenient viewing position" hypothesis. In Fisher, D. F., Monty, R. A., & Senders, J. W. (Eds.) *Eye Movements: Cognition and Visual Perception*. Erlbaum, 289-298.
- Osaka, N. (1993). Asymmetry of the effective visual field in vertical reading as measured with a moving window. In d'Ydewalle, G., & Van Rensbergen, J. (Eds.) *Perception and Cognition: Advances in Eye Movement Research*. Amsterdam: North-Holland, 275-283.
- Oulasvirta, A., & Saariluoma, P. (2004). Long-term working memory and interrupting messages in human-computer interaction. *Behavior Research Methods & Instrumentation*, **23** (1), 53-64.
- Pashler, H. (Ed.) (1998). *Attention*. Hove: Psychology Press.
- Pashler, H., Johnston, J. C., & Ruthruff, E. (2001). Attention and performance. *Annual Review of Psychology*, **52**, 629-651.
- Paulson, E. J., & Goodman, K. S. (1999). Influential studies in eye-movement research. *Reading Online*. Available at <http://www.readingonline.org/research/eyemove.html> (5.3.2006)
- Pollatsek, A., Bolozky, S., Well, A. D., & Rayner, K. (1981). Asymmetries in the perceptual span for Israeli readers. *Brain and Language*, **14**, 174-180.
- Pomplun, M., Ivanovic, N., Reingold, E. M., & Shen, J. (2001). Empirical evaluation of a novel gaze-controlled zooming interface. In *Proceedings of HCI International 2001*. Erlbaum.
- Posner, M. I. (1980). Orienting of attention. *Quarterly Journal of Experimental Psychology*, **32**, 3-25.

- Qvarfordt, P. (2004). *Eyes on Multimodal Interaction*. Ph.D. dissertation No 893. University of Linköping.
- Qvarfordt, P. & Zhai, S. (2005). Conversing with the user based on eye-gaze patterns. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. ACM Press, 221-230.
- Qvarfordt, P., Beymer, D., & Zhai, S. (2005). RealTourist – A study of augmenting human-human and human-computer dialogue with eye-gaze overlay. In *Proceedings of the International Conference on Human-Computer Interaction (INTERACT 2005)*. Springer, LNCS 3585, 767-780.
- Ramloll, R., Trepagnier, C., Sebrechts, M., & Finkelmeyer, A. (2004). A gaze contingent environment for fostering social attention in autistic children. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '04)*. ACM Press, 19-26.
- Raskin, J. (2000). *The Humane Interface: New Directions for Designing Interactive Systems*. Boston: Addison-Wesley.
- Rayner, K. (1977). Visual attention in reading: Eye movements reflect cognitive processes. *Memory & Cognition*, **4**, 443-448.
- Rayner, K. (1978). Eye movements in reading and information processing. *Psychological Bulletin*, **85**, 618-660.
- Rayner, K. (1995). Eye movements and cognitive processes in reading, visual search, and scene perception. In Findlay, J. M., Walker, R., & Kentridge, R. W. (Eds.) *Eye Movement Research: Mechanisms, Processes and Application*. New York: Elsevier, 3-21.
- Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, **124**, 372-422.
- Rayner, K., & Duffy, S. A. (1986). Lexical complexity and fixation times in reading: Effects of word frequency, verb complexity, and lexical ambiguity. *Memory and Cognition*, **14**, 191-210.
- Rayner, K. & Pollatsek, A. (1989). *The Psychology of Reading*. Englewood Cliffs: Prentice-Hall.
- Rayner, K., Sereno, S. C., Morris, R. K., Schmauder, A. R., & Clifton, C. (1989). Eye movements and on-line language comprehension processes. *Language and Cognitive Processes*, **4**, 21-50.
- Reichle, E., Pollatsek, A., Fisher, D., & Rayner, K. (1998). Toward a model of eye movement control in reading. *Psychological Review*, **105**, 125-157.

- Reingold, E. M., Loschky, L. C, McConkie, G. W., & Stampe, D. M. (2003). Gaze-contingent multiresolutional displays: An integrative review. *Human Factors*, **45**, 307-328.
- Richardson, D. C., & Spivey, M. J. (2004). Eye-tracking: Characteristics and methods. In Wnek, G., & Bowlin, G. L. (Eds.) *Encyclopedia of Biomaterials and Biomedical Engineering*. Marcel Dekker, 568-572.
- Roda, C., & Thomas, J. (2006). Attention aware systems: Theories, applications, and research agenda. *Computers in Human Behavior*, **22** (4), 557-587.
- Ruddaraju, R., Haro, A., Nagel, K., Tran, Q. T., Essa, I. A., Abowd, G., & Mynatt, E. D. (2003). Perceptual user interfaces using vision-based eye tracking. In *Proceedings of the 5th International Conference on Multimodal interfaces (ICMI '03)*. ACM Press, 227-233.
- Salvucci, D. D., & Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '00)*. ACM Press, 71-78.
- Sandstone. (2001). An electronic English - Finnish - German - Italian dictionary, the version from the year 2001. Sandstone information available at <http://www.sandstone.fi/> (5.3.2006)
- Selker, T. (2004). Visual attentive interfaces. *BT Technology Journal*, **22** (4), 146-150.
- Selker, T., Burleson, W., Scott, J., & Li, M. (2002). Eye-Bed. In *Proceedings of the Workshop on Multimodal Resource and Evaluation, in the Third International Conference on Language Resources and Evaluation (LREC 2002)*, 71-76.
- Selker, T., Lockerd, A., & Martinez, J. (2001). Eye-R, a glasses-mounted eye motion detection interface. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems*. ACM Press, 179-180.
- Shell, J. S., Vertegaal, R., Cheng, D., Skaburskis, A. W., Sohn, C., Stewart, A. J., Aoudeh, O., & Dickie, C. (2004). ECSGlasses and EyePliances: using attention to open sociable windows of interaction. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '04)*. ACM Press, 93-100.
- Shell, J. S., Vertegaal, R., & Skaburskis, A. W. (2003). EyePliances: attention-seeking devices that respond to visual attention. In *CHI '03 Extended Abstracts in Computing Systems*. ACM Press, 770-771.
- Shih, S.-W., Wu, Y.-T., & Liu, J. (2000). A calibration-free gaze tracking technique. In *Proceedings of the International Conference on Pattern*

- Recognition (ICPR 2000)*. International Association for Pattern Recognition, 201-204.
- Shneiderman, B. (1983). Direct manipulation: A step beyond programming languages. *IEEE Computer*, **16** (8), 57-69.
- Shneiderman, B., & Maes, P. (1997). Direct manipulation vs. interface agents. *Interactions*, **4** (6), 42-61.
- Sibert, J. L., Gokturk, M., & Lavine, R. A. (2000). The reading assistant: eye gaze triggered auditory prompting for reading remediation. In *Proceedings of the Symposium on User Interface Software and Technology (UIST '00)*. ACM Press, 101-107.
- Sibert, L. E. & Jacob, R. J. K. (2000). Evaluation of eye gaze interaction. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '00)*. ACM Press, 281-288.
- Simons, D. J. & Rensink, R. A. (2005). Change blindness: Past, present, and future. *Trends in Cognitive Sciences*, **9** (1), 16-20.
- Skaburskis, A. W., Vertegaal, R., & Shell, J. S. (2004). Auramirror: reflections on attention. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '04)*. ACM Press, 101-108.
- SMI. (2005). SensoMotoric Instruments GmbH, iView X eye tracker, information available at <http://www.smi.de/iv/index.html> (5.3.2006).
- Sodhi, M., Reimer, B., Cohen, J. L., Vastenburger, E., Kaars, R., & Kirschenbaum, S. (2002). On-road driver eye movement tracking using head-mounted devices. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '02)*. ACM Press, 61-68.
- Špakov, O., & Miniotas, D. (2004). On-line adjustment of dwell time for target selection by gaze. In *Proceedings of the Third Nordic Conference on Computer-Human Interaction 2004 (NordiCHI 2004)*. ACM Press, 203-206.
- SR Research. (2005). EyeLink eye tracker, information available at <http://www.eyelinkinfo.com/> (5.3.2006)
- Stampe, D. M. (1993). Heuristic filtering and reliable calibration methods for video-based pupil-tracking systems. *Behavior Research Methods, Instruments & Computers*, **25**, 137-142.
- Stampe, D. M., & Reingold, E. M. (1995). Selection by looking: A novel computer interface and its application to psychological research. In Findlay, J. M., Walker, R., & Kentridge, R. W. (Eds.) *Eye Movement Research: Mechanisms, Processes and Application*. Elsevier Science, 467-478.

- Starker, I., & Bolt R. A. (1990). A gaze-responsive self-disclosing display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90)*. ACM Press, 3-9.
- Surakka, V., Illi, M., & Isokoski, P. (2004). Gazing and frowning as a new technique for human-computer interaction. *ACM Transactions on Applied Perception*, **1** (1), 40-56.
- Takagi, H. (1997). How to detect higher-order thinking processes and knowledge states from eye-movements. In *Proceedings of the Workshop on Perceptual User Interfaces (PUI '97)*. Available at <http://nicosia.is.s.u-tokyo.ac.jp/members/hironobu/pui/pui.html> (27.4.2006)
- Takagi, H. (1998). Development of an eye-movement enhanced translation support system. In *Proceedings of the Third Asian Pacific Computer and Human Interaction*. IEEE Computer Society, 114-119.
- Tapanainen, P., & Järvinen, T. (1997). A non-projective dependency parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing*. Morgan Kaufman, 64-71.
- Tennenhouse, D. (2000). Proactive computing. *Communications of the ACM*, **43** (5), 43-50.
- Tobii Technology. (2005). Tobii eye tracker, information available at <http://www.tobii.se/> (5.3.2006)
- Triesch, J., Ballard, D. H., Hayhoe, M. M., & Sullivan, B. T. (2003). What you see is what you need. *Journal of Vision*, **3** (1), 86-94.
- Tummolini, L., Lorenzon, A., Bo, G., & Vaccaro, R. (2002). iTutor: A wireless and multimodal support to industrial maintenance activities. In *Proceedings of the 4th International Symposium on Mobile Human-Computer Interaction (Mobile HCI '02)*. London: Springer, 302-305.
- Underwood, G., & Radach, R. (1998). Eye guidance and visual information processing: Reading, visual search, picture perception and driving. In Underwood, G. (Ed.) *Eye Guidance in Reading and Scene Perception*. Amsterdam: Elsevier, 1-27.
- Vertegaal, R. (1999). The GAZE groupware system: mediating joint attention in multiparty communication and collaboration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM Press, 294-301.
- Vertegaal, R. (2002). Designing attentive interfaces. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '02)*. ACM Press, 22-30.

- Vertegaal, R. (2003). Attentive user interfaces. *Communications of the ACM*, **46** (3), 30-33.
- Vertegaal, R., Dickie, C., Sohn, C., & Flickner, M. (2002). Designing attentive cell phone using wearable eyecontact sensors. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems*. ACM Press, 646-647.
- Vertegaal, R., Slagter, R., van der Veer, G., & Nijholt, A. (2001). Eye gaze patterns in conversations: there is more to conversational agents than meets the eyes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*. ACM Press, 301-308.
- Vertegaal, R., Weevers, I., & Sohn, C. (2002). GAZE-2: an attentive video conferencing system. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems*. ACM Press, 736-737.
- Vertegaal, R., Weevers, I., Sohn, C., & Cheung, C. (2003). GAZE-2: conveying eye contact in group video conferencing using eye-controlled camera direction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM Press, 521-528.
- Villanueva, A., Cabeza, R., & Porta, S. (2004). Eye tracking system model with easy calibration. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '04)*. ACM Press, 55.
- Vitu, F., McConkie, G. W., & Zola, D. (1998). About regressive saccades in reading and their relation to word identification. In Underwood, G. (Ed.) *Eye Guidance in Reading and Scene Perception*. Amsterdam: Elsevier, 101-124.
- Vitu, F., & O'Regan, J. (1995). A challenge to current theories of eye movements in reading. In Findlay, J. M., Walker, R., & Kentridge, R. M. (Eds.) *Eye Movement Research: Mechanisms, Processes, and Applications*. New York: Elsevier, 381-391.
- Wade, N. J. (1998). *A Natural History of Vision*. Cambridge: The MIT Press.
- Wade, N. J., Tatler, B. W., & Heller, D. (2003). Dodge-ing the issue: Dodge, Javal, Hering and the measurement of saccades in eye movement research. *Perception*, **32**, 793-804.
- Wandell, B. A. (1995). *Foundations of Vision*. Sunderland: Sinauer Associates.
- Wang, H., Chignell, M., & Ishizuka, M. (2006). Empathic tutoring software agents using real-time eye tracking. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '06)*. ACM Press, 73-78.

- Ware, C. (2000). *Information Visualization: Perception for Design*. San Francisco: Morgan Kauffman.
- Ware, C., & Mikaelian, H. H. (1987). An evaluation of an eye tracker as a device for computer input. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface (CHI '87)*. ACM Press, 183-188.
- Westheimer, G. (1986). The eye as an optical instrument. In Boff, K. R., Kaufman, L., & Thomas, J. P. (Eds.) *Handbook of Perception and Human Performance. Volume I: Sensory Processes and Perception*. Wiley.
- Wolfe, J. M. (1998). Visual search. In Pashler, H. (Ed.) *Attention*. 5th edition, Hove: Psychology Press, 13-73.
- WSOY (2000). An electronic version tailored from the printed English-Finnish General Dictionary (Hurme R., Pesonen M., & Syväoja, O.) WSOY, Porvoo, 1994.
- Zhai, S. (2003). What's in the eyes for attentive input. *Communications of the ACM*, **46** (3), 34-39.
- Zhai, S., Morimoto, C., & Ihde, S. (1999). Manual and gaze input cascaded (Magic) pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM Press, 246-253.
- Zhu, Z., Fujimura, K., & Qiang, J. (2002). Real-time eye detection and tracking under various light conditions. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA '02)*. ACM Press, 139-144.
- Yantis, S. & Jonides, J. (1990). Abrupt visual onsets and selective attention: voluntary versus automatic allocation. *Journal of Experimental Psychology*, **16** (1), 121-134.
- Yarbus, A. R. (1967). *Eye Movements and Vision*. New York: Plenum Press.
- Young, L. R., & Sheena, D. (1975). Methods & designs - survey of eye movement recording methods. *Behavior Research Methods and Instrumentation*, **7**, 397-429.

1. **Timo Partala:** Affective Information in Human-Computer Interaction
2. **Mika Käki:** Enhancing Web Search Result Access with Automatic Categorization
3. **Anne Aula:** Studying User Strategies and Characteristics for Developing Web Search Interfaces
4. **Aulikki Hyrskykari:** Eyes in Attentive Interfaces: Experiences from Creating iDict, a Gaze-Aware Reading Aid