Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers in Human Behavior 22 (2006) 657–671

Computers in Human Behavior

www.elsevier.com/locate/comphumbeh

Utilizing eye movements: Overcoming inaccuracy while tracking the focus of attention during reading

Aulikki Hyrskykari

Tampere Unit for Computer–Human Interaction, Department of Computer Sciences, University of Tampere, FIN-33014, Finland

Available online 8 February 2006

Abstract

Even though eye movements during reading have been studied intensively for decades, applications that track the reading of longer passages of text in real time are rare. The problems encountered in developing such an application (a reading aid, iDict), and the solutions to the problems are described. Some of the issues are general and concern the broad family of Attention Aware Systems. Others are specific to the modality of interest: eye gaze. One of the most difficult problems when using eye tracking to identify the focus of visual attention is the inaccuracy of the eye trackers used to measure the point of gaze. The inaccuracy inevitably affects the design decisions of any application exploiting the point of gaze for localizing the point of visual attention. The problem is demonstrated with examples from our experiments. The principles of the drift correction algorithms that automatically correct the vertical inaccuracy are presented and the performance of the algorithms is evaluated.

Keywords: Eye movements; Eye-tracking; Attention aware systems (AAS); Attentive user interfaces (AUI); Reading; Gaze-aware applications; Inaccuracy of eye tracking

1. Introduction

Graphical user interfaces have been the dominating interaction paradigm in human–computer interaction already for two decades. However, the restricted input devices (mainly only the mouse and the keyboard are used) seem to waste the richness of the forms

E-mail address: ah@cs.uta.fi.

0747-5632/$ - see front matter
doi:10.1016/j.chb.2005.12.013

human beings naturally use in expressing themselves. The explosive development of techniques supporting the presentation of multimedia contents deepens the existing imbalance of deploying the human input and output abilities (Zhai, 2003). Eye tracking techniques have been used for a long time to explore human perceptual and cognitive processes (for a review, see Rayner, 1998). However, only now eye trackers have reached the level where they can be considered as potentially useful input devices; the prices of high quality trackers are falling steadily and their usability begins to be viable as a part of a standard computer set-up. The potential benefits of using an eye tracker as an input device have been discussed widely in several recent papers (Hyrskykari, Majaranta, Aaltonen, & Räihä, 2000; Sibert & Jacob, 2000; Jacob & Karn, 2003; Zhai, 2003; Surakka, Illi, & Isokoski, 2004). The main advantage of eye movements is that they are fast and natural. For example, Sibert and Jacob (2000, p. 281) report that "*when eye gaze interaction is working well, the system can feel as though it is anticipating the user's commands, almost as if it were reading the user's mind*".

1.1. Striving towards natural interfaces

In fact, the call for interaction methods taking better account of human capabilities in transmitting information has existed for at least as long as the era of graphical user interfaces. Already the research on natural language interfaces in the late sixties (see Androutsopoulos, Ritchie, & Thanisch, 1995) shared the common goal of making the interaction more natural and pleasant leading to more efficient interaction. More recent interaction paradigms include speech user interfaces, tangible interfaces, perceptual (or pervasive) user interfaces, context–aware interfaces, and the connective paradigm of multimodal user interfaces. Also a set of related concepts like intelligent environments, affective computing, adaptive systems, transparent interfaces, non-command interfaces, and proactive interaction share the same objective of natural and effective human–computer interaction.

In principle, the means for achieving the goal is different within different concepts, but in many cases the concepts are confusingly overlapping. Attentive User Interfaces (AUIs) is yet another novel user interface paradigm; in May 2003 the Communications of the ACM dedicated a special issue to AUIs. What distinguishes AUIs from the other related HCI paradigms is that they emphasize *designing for attention* (Vertegaal, 2003).

1.2. Attention aware systems (AAS)

Roda and Thomas (2006) use the term Attention Aware Systems (AAS, instead of AUI) to emphasize that questions related to attention involve designing the whole system, rather than being restricted to the design and implementation of the interface only. Sense of attention is of cardinal importance in human–human interaction (Bellotti et al., 2002). Surely the sense of attention is also potentially valuable information in situations when the interlocutor is a computer instead of another human being. Maglio, Matlock, Campbell, and Zhai (2000) itemize that such systems

(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 user interacting with a system. Perceptual sensors (Horvitz, Kadie, Paek, & Hovel, 2003), including microphones listening to acoustic information, cameras enabling analysis of the user's gaze and body gestures, and electronic sensors recording muscle and brain activity can be used to monitor users' actions during the performance of a task (Stiefelhagen, Chen, & Yang, 2003; Surakka et al., 2004). In addition to using perceptual sensors, valuable information may also be gathered by observing the user's interaction with software and devices, as well as analysing the history of the user's prior interests and patterns of activities and attention (Maglio et al., 2000; Horvitz et al., 2003; Maglio & Campbell, 2003). This information can then be used to infer the focus of the user's attention either presently or more generally over a longer period of time, or even to forecast the probable actions. Thus, the information may be used to adapt the behaviour of the system to anticipate the user's needs or preferences.

1.3. Eye tracking and attention aware systems

Tracking the user's gaze behaviour is obviously a valuable technique for monitoring the user's behaviour from which we can then infer the goals and interests. Our application, iDict (to be presented in Section 3), uses eye input to catch the situations when the reader—while reading text written in a foreign language—seems to have comprehension difficulties. To obtain the goal, it is essential that we are able to map the fixations[1] of a reader onto the words being read. However, the fact that measuring the user's position of gaze is not accurate, but always exhibits a positional tolerance, sets a challenge for the application.

The problem is often neglected in experiments deploying eye tracking, even though it is very substantive when considering the reliability of the results (Hornof & Halverson, 2002). Hornof and Halverson (2002) introduced the idea of using "implicitly required fixation locations" (RFLs) to correct the systematic error of the measured position of gaze. The principle is that when—during an eye tracking session—we can reliably assume that a user's visual attention is focused at a certain object, we use the information of the real position of the object and the measured point of the object to reason the probable error given by the eye tracker. This information can then be used to correct the subsequent eye tracking data. Hornof and Halverson managed to correct the inaccuracy in their laboratory experiment, when the required fixation location was artificially betoken within the task setup. In this paper, the idea is tried out in the context of a real application. Moreover, our drift correction algorithms are not only able to correct the systematic error commonly introduced by eye trackers, but also the variable error caused by drift of calibration.

In the following I first elaborate issues related to gaze when used as a source of information in attentive user interfaces (Section 2). In Section 3 I present the main ideas behind iDict, and also briefly discuss the design principles that emerged during the development

[1] For the readers not familiar with gaze: definition of a fixation is given in Section 4.

and the implementation of the application. The rest of the paper concentrates on the inaccuracy problems encountered during the development of the system. I demonstrate the problems with examples of reading paths from our experiments (Section 4), and introduce the principles of the algorithms developed to deal with the problem along with the evaluation of their performance (Section 5). The article concludes with an assessment of the success achieved in handling inaccuracy problems (Section 6).

2. Gaze in attention aware systems

Unlike any other input technique, the point of gaze reveals the focus of the user's visual attention (some researchers use the term locus instead of focus in the context of attention, e.g. Raskin, 2000). In this paper we are studying the eye movements while the user concentrates on performing a task requiring visual attention, in our case reading. Hence, I use the term focus of *visual* attention; focus of attention is a more general concept. The focus of visual attention does not necessarily coincide with the focus of attention. For example, when you are reading this article and you hear a dog barking outside of your window, your focus of attention may be drawn to the sound for a moment. However, when concentrating on reading we assume that there is a high correlation between the focus of visual attention and the focus of attention.

2.1. Role of gaze in AAS

Some studies on the use of diverse input modalities (like speech or gestures) rely on giving commands to the computer, whereas others concentrate on using the natural behaviour of the user to guide the system's actions. Similarly, we can distinguish two different ways of using eye movements in the interface: the user can either intentionally use her gaze to initiate actions, or the natural eye movements of the user can be followed and interpreted for the application to better adapt the system's actions to the user's behaviour.

Applications using gaze for command-and-control have been created for physically challenged users, like for people suffering from motor neuron diseases. For a restricted user group the eyes may be even the only means of communication (Majaranta & Räihä, 2002). However, since eyes are naturally used as a perceptual organ, using them for giving commands involves problems which make gaze commands less appealing to standard user groups. Jacob (1991) called one of the problems the Midas Touch problem. The problem arises from the difficulty of differentiating the aim to use eyes for activating functions from the occasions when the user's aim is simply to obtain information. We do not want the application to react every time the target of the gaze changes, only in appropriate situations, and at the right moment.

Additionally, an untrained user often finds using eyes in an unnatural way stressful. The AAS approach is to let the user disregard the monitoring of eye movements by using the natural gaze paths as the source of information for the application. The distinction between intentional and natural eye movements is not necessarily strict, as the study on "Magic Pointing" (Zhai, Steven, & Ihde, 1999) demonstrates. Generally, in pointing tasks the eyes always move first to the target, and the cursor then follows. Thus, using the point of visual attention directly for positioning the mouse cursor seems to be an attractive idea. However, the inaccuracy of eye tracking makes it burdensome for the user; the focus point usually is a bit off from the targeted object. Magic Pointing combines the strengths of the two input modalities; the speed of the eye and the accuracy of the hand. The gaze location only defines the cursor a dynamical "home" position. Thus, when the user is about to point and select a target the cursor is already "automatically" at the vicinity of the target. Even though the gaze-command interfaces seem to be valuable for restricted user groups only, Magic Pointing encourages further study of innovative ways to use eye tracking as an additional input modality combined with other input methods.

For the moment eye-aware applications using the user's natural eye movements are still rare, but different applications are clearly emerging from the work of several research groups around the world. The chicken-and-egg dilemma of eye tracking technology has been a widely recognized problem of eye tracking addressed already by Bolt (1985). The costs of eye tracking systems have been extremely high; so far they have mainly been used in research laboratories as measurement tools. The gradual development of applications making versatile use of eye tracking (targeted not only for people with motor disabilities but also for standard users) is just beginning to create a volume of demand that helps the tracking technology providers to lower the costs to an affordable level.

2.2. Inaccuracy of the measured point of gaze

For a new-comer to eye movement research the idea of using eye movements in the interface sounds appealing. However, eye tracking entails problems which temper the initial enthusiasm. The most critical problems arise from the inaccuracy of eye tracking. In present eye trackers the inaccuracy problems of the measured focus of visual attention originate from at least the following three different sources:

1. Inaccuracy of measurement—the accuracy of measured point of gaze varies depending on the used eye tracking device and success of the performed calibration.
2. Drift of calibration—originates from the imprecise compensation of head movements or from the change of the size or shape of the measured characteristics of the eye (e.g. changes of the size and the shape of the tracked pupil). Thus, the accuracy often decreases during the session, even though the quality of a calibration was originally good.
3. Biological characteristics of an eye—a positional tolerance for the tracking accuracy is due to the fact that the reader can focus the visual attention without moving the eyes (e.g. Rayner, 1995; Coren, Ward, & Enns, 1999, p 437). Bates and Istance (2003) phrase the issue as follows

   > *The fovea of the eye, which gives clear vision, covers a visual angle of ~1° arc of the retina, hence when fixating a target the eye only needs to be within ~1° of the target position to clearly see the target. This gives an inaccuracy in measured gaze position.*

The present inaccuracy of the measured point of gaze is a combination of the three reasons presented above. Even though the inaccuracy problems caused by the first two reasons may diminish along the development of eye tracking technology, biological characteristics of an eye will make the positional tolerance a permanent feature in eye tracking.

An application, Reading Assistant (Sibert, Goturk, & Lavine, 2000), has partly similar objectives with iDict. It uses eye gaze to trigger auditory prompting for remedial reading instruction. In the tests performed with Reading Assistant the font size used was 40 pt with 70 pt line separation, which of course was acceptable in an experimental setting. However, if we want to track reading in normal user interface conditions, we need to be able to track text written with a notably smaller font size.

A common text size read from the screen is 11–14 pt with 1.5 five line spacing. For example, height of a character displayed with 11 pt Verdanda on a 17″ screen, when using 1024×768 resolution is about 3.5 mm. Viewed from a distance of 60 cm it covers a visual angle of 0.33°. The row height of a single spaced of that text would be 6.3 mm, which ends up covering a visual angle of 0.6°. This means that tracking the reading of text written in 11 pt font size approaches the limits of eye tracking; some algorithmic compensation for the positional tolerance should be developed.

3. iDict, assisting in reading texts written in a foreign language

Our gaze-aware application, iDict, helps the user in reading electronic documents written in a foreign language. The goal is to proactively provide the user with the right kind of help at the right time. The main ideas of iDict were published in (Hyrskykari et al., 2000).

3.1. User's perspective of iDict

The use of iDict begins with a calibration of the eye tracker. Some of the new eye trackers support persistent calibration: they do not require calibration for every session. In that case the user has to be identified for the application, so that the calibration information can be loaded into the application. Then, the user just starts reading the text. As soon as the user hesitates while reading a word or a phrase, the embedded dictionaries are automatically consulted and a gloss (an instant translation) is provided.

Normally, when reading text documents written in a foreign language, the unfamiliar words or phrases cause the reader to interrupt the reading and to get help from either printed dictionaries, or from electronic dictionaries installed in the computer. In both cases the process of reading and line of thought gets interrupted. After the interruption getting back to the context of text takes time, and may even affect the comprehension of the text being read.

Through using eye tracking, iDict aims to minimize the interference in the process of reading while still providing the reader with the appropriate help. The reader's eyes are tracked and the reading path is analyzed in order to detect deviations from the normal path of reading, indicating that the reader may be in need of help with the words or phrases being read at the time. The main window of the application is divided into two frames (see Fig. 1): Document Frame contains the text to be read and Dictionary Frame is used for displaying the complete information retrieved from the dictionaries embedded in the application. Thus, the assistance is provided to the reader on two levels.

Firstly, when a probable occurrence of difficulties of comprehension is detected, the reader gets a gloss (an instant translation) for the word. The gloss is positioned right above the problematic word to allow a convenient quick glance at the available help. The gloss is the best guess for the translation on the spot. It is deduced from the syntactical and lexical features of the text combined with information derived from the embedded dictionaries.

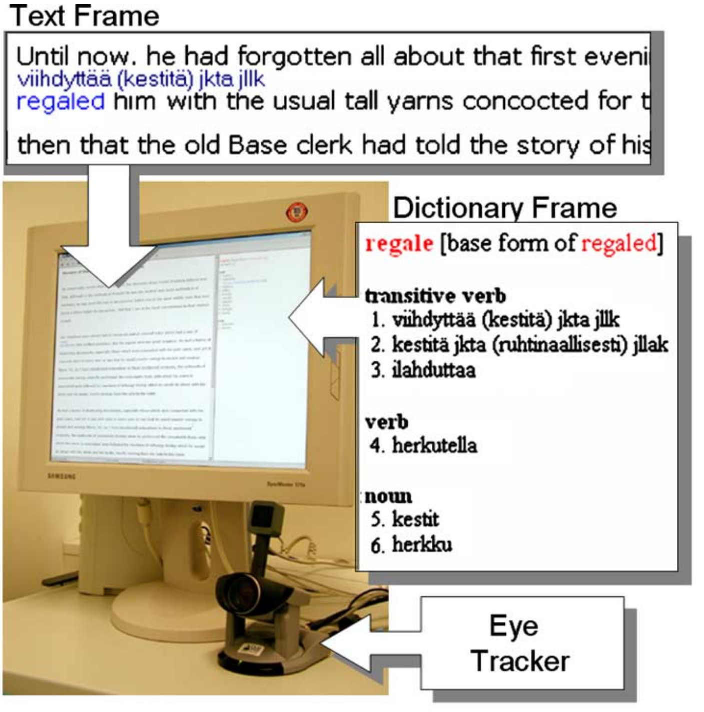

Fig. 1. Overview of the iDict application. A gloss is given proactively in-between lines of the document in the Text Frame. On a basis of a gaze gesture, a whole dictionary entry is fetched in the Dictionary Frame. iView X was one of the trackers used in implementation.

For example, for deciding the Finnish gloss for the word ‘‘regale’’ in Fig. 1, iDict has been able to parse the base form for the word. Additionally, linguistic analysis for the text gives iDict information that the word is a verb, and furthermore, a transitive verb. Even though linguistic and lexical analysis is used to choose the most probable gloss from among several options, the choice cannot always be the right and sometimes not even the only one. If the user is not satisfied with the gloss, a gaze gesture denoting attention shift to the Dictionary Frame on the right makes the whole dictionary entry with optional translations to appear.

Hence, the eye movements of the user during reading are used in iDict to infer the user's point of attention. The gloss is provided proactively on the basis of several indicators, with the total time spent on the words being the dominating factor for the triggering (Hyrskykari, 2003). The entry from the dictionary is fetched into the Dictionary Frame when the reader turns her/his eyes at the right part of the application window. It can be seen either as a proactive function of the system or alternatively as a conscious command to go and consult the dictionary. The Dictionary Frame always displays the entry for the most recently triggered gloss, overwriting the old information. In user tests this proved to be the most self-explanatory interpretation for the gaze gesture of gaze suddenly leaving the Document Frame and entering the Dictionary Frame.

Thus, the principle is that when using iDict all the reader has to do is to read the text and the help is provided automatically. We used three different eye trackers when developing iDict: EyeLink, a tracker with head mounted optics, iView X, the tracker shown in Fig. 1, and Tobii (for more information on each of the trackers, see the links provided in the references). iView X and Tobii use remote optics, i.e. the camera following the user's eyes is located in the proximity of the screen, so the user needs not "to dress" any additional equipments, whereas with EyeLink the user has to wear a head band with the cameras attached to the band. Tobii supports persistent calibration.

3.2. Design principles of iDict

The goal of our system is to provide a transparent interface for the user, with no need for explicit commands. However, if the application is poorly designed, automatically triggered actions easily become irritating. In our previous paper (Hyrskykari, Majaranta, & Räihä, 2003) the issues of designing proactive applications are discussed in detail. Most of the problems we encountered concern either inaccuracy of the measured point of gaze (resulting erroneous glosses), or too eagerly triggered proactive actions. Inaccuracy is reviewed more closely in the last two sections of this paper.

The three most essential things worth recommending to the designers of applications similar to iDict, are: (1) proper feedback, (2) controllability, and (3) noninterfering design. These principles are by no means new, but the fact that they are valid in applications of a proactive nature is not self evident. I will next treat each of them briefly.

3.2.1. Proper feedback

Proactive behaviour was originally one of the goals of our system. In evaluations of iDict we observed that without proper feedback the proactivity may be disturbing. As a concept, a proactive application is very close to an application having a transparent interface. A proactive application tries to predict the intentions and needs of the user, so that the application may perform the action on behalf of the user (Tennenhouse, 2000). In that sense our application tries to make the interface disappear and lets the user concentrate on the reading task. However, we concluded that the transparency should not mean that the operation of the application is hidden from the user.

On the contrary, the user should understand why something happens automatically, thus the role of feedback is essential. Especially if some unexpected action occurs, understanding why it happened helps the user to accept it. In iDict, for example, getting a gloss for a word on a line above or below the line the user is reading does not make sense if the reader does not know how the application arrived at the faulty gloss. We therefore implemented different modes of feedback in iDict.

One of them is a line marker, a smooth grey line under the line which is assumed to be currently read. The line marker makes the operation of line tracing algorithms visible to the user. The line marker was found valuable not only for making the line tracing visible, but also for minimizing the disruption when the user transferred the visual attention to the Dictionary Frame to check the whole dictionary entry for a word. The line marker, which is left on the line where the reading was proceeding before the interruption, helps the user to return to the correct point after consulting the dictionary. Roda and Thomas discuss related issues of minimizing disruption (in this issue, see "attentive dispatching" in the section "Defining attention aware systems"). Another feedback mode the user may turn on is a small, almost invisible, red spot displayed under the word assumed to be the focused. The spot is small enough not to disturb the reader, but still available.

3.2.2. Controllability

By controllability we mean that proactive actions should not make the user feel helpless. The user should not only understand what is happening, but also be able to influence the operation. In iDict the user can always use the mouse to trigger help for the desired word, and additionally, use arrow keys to correct the measured focus of visual attention. The manual correction using the arrow keys allows the user (without grabbing the mouse) to help the system in triggering the help for the right word.

The user is also able to explicitly adjust the level of proactivity (the sensitivity of triggering actions) of the system.

3.2.3. Noninterfering design

Even if the level of proactivity was tuned to comply with the user's individual preferences, false actions are unavoidable. The third issue, noninterfering design, emphasises that the costs of wrong decisions (Horvitz & Apacible, 2003) can be minimised by careful visual design. For example, the studies on the change blindness effect can be taken into account when designing the given feedback. It has been 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 and Rensink, 2005). On the other hand, onset motion is shown to attract observer's attention (e.g. Abrams & Christ, 2003).

When testing iDict many of the test users reported that they did not notice the glosses which they did not expect. That may be due to the fact that the glosses were designed to appear smoothly without visual noise. Additionally, the location of unnecessary gloss is often outside the focus of the user's visual attention, since the reading has already continued beyond the point where the system erroneously decided to take action.

Besides the change blindness, the observation can be explained with the Grossberg's Adaptive Resonance theory (ART) discussed by Roda & Thomas (in this issue). According to the interpretation of the theory given by Roda & Thomas

> "… *intentions reflect expectations of events that may (or may not) occur*." [*and*] "… *the user's attention will be focussed on information that matches their momentary expectations*"

In the context of iDict and unnecessarily given glosses, the fact that the user is not expecting a gloss to occur accounts for the reports of the users not perceiving them.

4. Inaccuracy in tracking the focus of attention during reading

In order to interpret the progress of reading iDict has to be able to keep track of the words focused during a reading session. Eye movements consist of fixations and saccades. During fixations the eyes stay relatively still for a short while (typically 200–600 ms). Between fixations the gaze jumps rapidly from one fixation to another; the jumps are called saccades. Thus, we should be able to map the fixations onto the focused words. The following examples illustrate the difficulties of inaccuracy of the measured fixation location in performing the required mapping.

The reading path of a clip of three lines of text is presented in Fig. 2. The fixations are visualised as circles (radius being proportional to the duration of the fixation) and saccades as lines connecting the fixations. The figure illustrates that mapping the fixations onto the words must take into account the wider context of reading the whole text. For

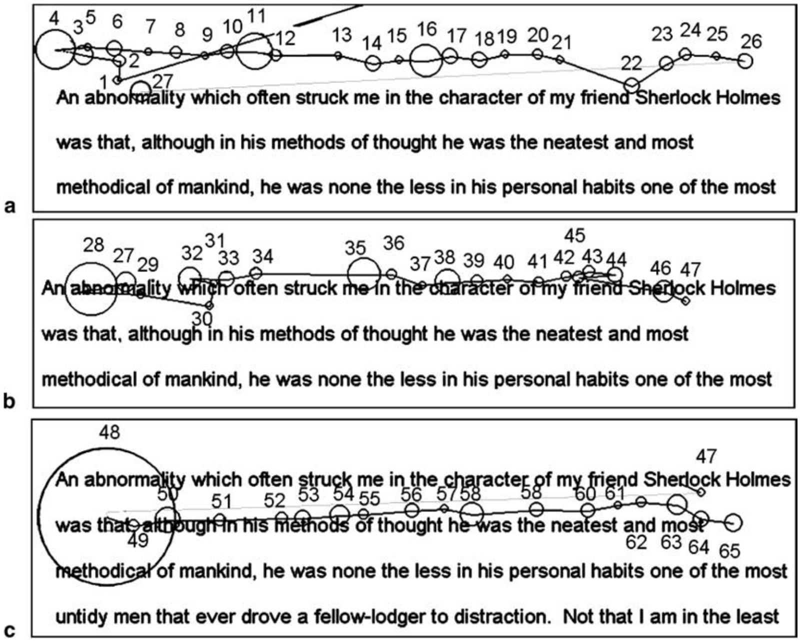

Fig. 2. Reading paths of three lines of text.

example, if the reading path from fixations 27 to 47 were viewed without the context, and the fixations were straightforwardly mapped onto the closest words, they would be mapped onto the words in the first line. However, the context reveals that those fixations are actually used for reading the second line of the clip. The same goes for the fixations from 48 to 65, which—when interpreted out of context—would be mapped onto the second line instead of the third line.

Thus, in Fig. 2 there is an obvious vertical inaccuracy in the measured fixation locations. It would be more convenient to determine the error if it were consistent in all parts of the screen, but this is seldom the case.

Fig. 3 is an example of an ascending reading path very common in our experiments. At the beginning of the line, measured fixation locations hit the line quite accurately (in this case the first fixations were really focused on the second line), but then the locations rise as the reading of the line proceeds. If these fixations were directly mapped onto the closest words, all the fixations from 9 to 18 would be mapped onto the words in the upper line. However, reviewing the reading paths of the previous and successive lines (not present in Fig. 3) makes it obvious that the lower line was read with fixations 1–18. Although these ascending reading paths were more common in our experiments, similar examples with descending fixation locations could also be found.

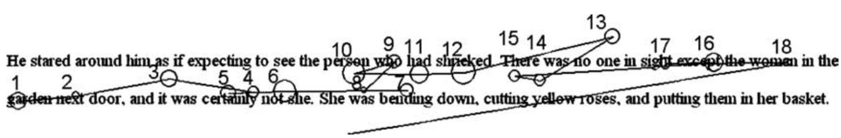

Fig. 3. An ascending reading path.

Horizontal inaccuracy is much more difficult to spot from the reading path, but in our experience (and also reported by other researchers, e.g. by Stampe & Reingold (1995)), horizontally eye trackers give more accurate data than vertically. It is interesting, though, that the controlled experiments performed by Hornof and Halverson (2002) did not affirm this common observation. Nevertheless, in iDict evaluations the participants seldom had troubles with horizontal accuracy. Hence, it is possible that the observation is due to the features of the tracked task. When reading the visual angle for a word being read is horizontally much bigger than vertically. Additionally, it is notable that the same trends in measured gaze paths were found with all three trackers used. Thus the troubles with vertical inaccuracy is not totally due to a slipping head band when the head mounted tracker was used.

5. Dealing with the inaccuracy

The example paths above illustrate clearly that when the fixations are to be mapped onto words in real time, some kind of algorithmic compensation for the inaccuracy is needed. Furthermore, studying the reading paths reveals that the compensation of the inaccuracy cannot be applied uniformly to all fixations (as was done in the experiment performed by Hornof & Halverson, 2002), because it varies in different parts of the screen.

5.1. Drift correction algorithms

The drift correction algorithms developed for iDict operate on three levels. The first two of them are performed automatically, and the third one is left to be done explicitly by the user.

1. *Sticky lines*. The algorithm was designed to follow the line of reading in spite of the ascending or descending gaze paths due to inaccuracy of the measured point of gaze. The compensation is performed on the basis of observing the reading proceeding fluently across the line being read.
2. *Magnetic lines*. The algorithm was designed to compensate vertical inaccuracy at the beginning of a line. The compensation is performed on the basis of identified transfers from the end of a line to a new line.
3. *Manual correction*. Recalculates the relation between a line (vertical correction), or a word (horizontal correction) and the measured point of gaze on the basis of the feedback given by the user.

The algorithm for sticky lines tracks the path of reading and allows greater vertical errors for fixations as long as the reading of the current line appears to be continuing.

The algorithm for magnetic lines adjusts the mapping for the line being read and for a few lines in its vicinity. We analysed reading paths and developed algorithms to identify when the reader transfers from a line to a new one. We call these events "new line events". The new line events were then used to define the apparent need for dynamic correction. Here, the automatically detected new line events take the role of RFLs introduced by Hornof and Halverson (2002).

Manual correction is performed by the user, in a situation when (s)he notices that iDict has mapped the fixations onto the wrong words. The arrow keys can be used to correct the measured point of gaze either horizontally (one word left/right) or vertically (one line up/down). The correction is local in the same way as the automatically performed correction.

Also the action of manual correction gives us a point in time when we can use the line, for which the correction request was performed, as a RFL.

Thus, the dynamic correction of vertical inaccuracy is performed automatically. In case the automatic corrections fail, the correction can also be called explicitly via arrow keys. Also the manual corrections affect the subsequent interpretation of fixation locations.

5.2. Performance of the drift correction algorithms

The performance of the developed algorithms was tested using texts with varying line spacing. Six participants read three text documents, each of which contained about 250 words and 20 lines. The font used in the texts was 11 pt Verdana. One of the texts was displayed with single line spacing, one with 1.5 line spacing, and one with double line spacing. The total numbers of fixations recorded from the sessions for single, 1.5, and double spaced texts were 1380, 1470, and 1484 fixations, respectively.

iDict was used for recording the fixations during the reading sessions. The recordings were "pure" in the sense that iDict was not active during the experiments. Neither glosses nor the third level algorithms of correction (manual corrections) were used during the experiment. Afterwards, the fixations were first algorithmically mapped to the closest words, and then the drift correction algorithms were used for mapping the words onto the lines. Then the correct lines for each fixation were manually determined. The examples in Section 2 demonstrate that the right line of a fixation can be decided 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 the reason why the number of participants in the experiment had to be restricted to six. As a result from the fact that we now knew the correct mappings we got a "hit percentage", the percentage of the fixations that was mapped onto the right line with and without the drift correction algorithms. The hit percentages for each test participant and line spacing condition are presented in Table 1.

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 line spacing 1.5 and 2 were 56% rising to 86%, and 76% rising to

Table 1
Hit percentages of correct line mappings of fixations by readers for three differently line spaced texts, when the drift correction algorithms were not applied and when they were applied

Participant	Algorithms					
	Line spacing 1.0		Line spacing 1.5		Line spacing 2.0	
	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
Average	39	53	56	86	76	78

78%, respectively. The improvement was statistically significant in the condition of reading text with 1.5 line spacing was ($F(1,5) = 9.2$, $p < 0.5$). However, making inferences on the basis of statistical analysis performed for a data with only six participants is highly dubious. Thus, the results are discussed and interpreted in words below.

Overall, the results were very satisfactory. The algorithms performed best for the most commonly used 1.5 line spacing. For three of the participants (3, 4, and 6) the algorithms performed almost perfectly in a condition of 1.5 line spacing: for each of them only isolated fixations were mapped onto a wrong line. It should be remembered that iDict was not active during the experiments. Manual correction made 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 easily stays on the wrong course for a longer time. An extreme example of that can be seen in the reading session of the fifth reader with double line spacing. Correspondingly, the algorithms would in many cases give substantially higher hitting rates with even a single corrective action performed by the user. It is interesting to see that on average, even without manual correction, the algorithms performed reasonably, and mostly the hitting percent rose considerably.

The surprising fact that the correction algorithms resulted in a better average outcome with the 1.5 line spacing than with the double spaced text can be mainly explained with the results of the tracking session of the fifth reader (double spaced text). Still, the overall improvement achieved with the algorithms was smaller in the single and double spaced than in the 1.5 spaced line condition. Likely explanations for this can be given. Tracking the reading of a double spaced text (with 11 pt 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 (i.e. in our case the double line spaced text seemed to collide the accuracy provided by used eye tracker, iView X), the effect of drift correction algorithms vanishes. 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 line spaced. A closer look at the reading paths suggests that the reading paths of the single spaced texts were less fluent. That is, there were occasions where the reader had trouble keeping in the right line. Although the algorithms are designed to take into account the normal regressions that the reader often performs in order to check words already read, the algorithms failed more easily to trace the line of reading, because the reader her/himself lost track of the line being read. 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.

6. Conclusions

The indisputable advantage of using eye tracking over any other input device is that the gaze point carries the information of the focus of the user's visual attention. In attention aware systems we can exploit the information of the user's natural eye behaviour. We provided and exemplified guidelines for designing of applications, which act on the basis of their knowledge of user's visual attention. Even if the guidelines are developed while designing and implementing iDict they are potentially applicable to many other AAS systems, especially to other gaze aware systems.

However, the fact that measuring the user's position of gaze is not accurate – and despite of the constantly developing eye tracking technology will probably never be – sets a challenge to study solutions to overcome the difficulties rising from the inaccuracy.

The drift correction algorithms described in this paper are used in iDict to improve locating the focus of visual attention during reading. Their performance was tested with varying line spacing and the accuracy of succeeding to keep track of the line being read improved from 56% to 86% when the line spacing used was 1.5. The algorithms have also shown their value in many demonstrations. An early version of the algorithms was used in the Exhibition of the IST Conference in Copenhagen in November 2002. We demonstrated iDict on a stand for three days and provided all visitors with a chance to try out the application them selves. During the exhibition about 60 people tried out the application, and for most iDict performed well (Hyrskykari et al., 2003).

The experiments with the drift correction algorithms demonstrated that the inaccuracy of the measured point of gaze inherent in eye trackers can be reduced by designing intelligent algorithms using knowledge of the normal eye behaviour in the context of the task performed. The algorithms implemented for the iDict application showed that is the case when reading is tracked. The idea can also be generalised to other gaze-aware applications. The premise to the real-time correction is that the task introduces some recognizable gaze patterns, or that during the task performed, we can reliably assume that the user's visual attention is targeted to a certain point (RFL) on the screen.

Acknowledgements

I thank Kari-Jouko Räihä and three anonymous reviewers for their valuable comments on this paper. The first prototype of iDict was mostly developed in the iEye project funded in the fifth research framework programme of the European Union, IST. The current work is supported by the Academy of Finland (Project 178099).

References

Abrams, R. A., & Christ, S. E. (2003). Motion onset captures attention. *Psychological Science, 14*, 427–432.

Androutsopoulos, I., Ritchie, G. D., & Thanisch, P. (1995). Natural language interfaces to databases—an introduction. *Journal of Language Engineering, 1*(1), 29–81.

Bates, R., & Istance, H. O. (2003). Why are eye mice unpopular? A detailed comparison of head eye controlled assistive technology pointing devices. *Universal Access in the Information Society, 2*, 280–290.

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 Conference on Human Factors in Computing Systems (CHI 2002)* (pp. 415–422). NewYork: ACM Press.

Bolt, R. A. (1985). Conversing with computers. *Technology Review, 88*, 35–43.

Coren, S., Ward, L. M., & Enns, J. T. (1999). *Sensation and perception*. Harcourt Brace College Publishers.

EyeLink, SR Research, information available at http://www.eyelinkinfo.com/.

Hornof, A., & Halverson, T. (2002). Cleaning up systematic error in eye tracking data by using required fixation locations. *Behavior Research Methods, Instruments, and 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 2003)*, pp. 20–27.

Horvitz, E., Kadie, C., Paek, T., & Hovel, D. (2003). Models of attention in computing and communication: from principles to applications. *Communications of the ACM, 46*(3), 52–59.

Hyrskykari, A., Majaranta, P., Aaltonen, A., & Räihä, K.-J. (2000). Design issues of iDict: a gaze-assisted translation aid. In *Proceedings of Eye Tracking Research and Applications Symposium (ETRA 2000)* (pp. 9–14). New York: ACM Press.

Hyrskykari, A. (2003) Detection of comprehension difficulties in reading on the basis of word frequencies and eye movement data. Presented as a poster in 12th European Conference on Eye Movements (ECEM12), Dundee, Scotland, 2003.

Hyrskykari, A., Majaranta, P., & Räihä, K.-J. (2003). Proactive response to eye movements. In: *Proceedings of INTERACT 2003*, pp. 129–136.

iView X, SMI, information available from http://www.smi.de/iv/.

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*(3), 152–169.

Jacob, R. J. K., & Karn, K. S. (2003). Eye tracking in human–computer interaction and usability research: Ready to deliver promises. In J. Hyönä, R. Radach, & H. Deubel (Eds.), *The mind's eye: Cognitive and applied aspects of eye movement research* (pp. 573–605). Amsterdam, The Netherlands: North-Holland.

Maglio, P. P., & Campbell, C. S. (2003). Attentive user interfaces: attentive agents. *Communications of the ACM, 46*(3), 47–51.

Maglio, P.P., Matlock, T., Campbell, C.S., Zhai, S., & 0Smith, B.A. (2000). Gaze and speech in attentive user interfaces. In: *Proceedings of the Third International Conference on Multimodal Interfaces (ICMI 2000)*, pp. 1–7.

Majaranta, P., & Räihä, K.-J. (2002). Twenty years of eye typing: systems and design issues. In *Proceedings of Eye Tracking Research and Applications Symposium (ETRA 2002)* (pp. 15–22). New York: ACM Press.

Raskin, J. (2000). *The humane interface: new directions for designing interactive systems*. Addison-Wesley.

Rayner, K. (1995). Eye movements and cognitive processes in reading, visual search, and scene perception. In J. M. Findlay, R. Walker, & R. W. Kentridge (Eds.), *Eye movement research: Mechanisms, processes and application* (pp. 3–22). New York: Elsevier Science Publishers.

Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin, 124*, 372–422.

Roda, C., & Thomas, J. (2006). Attention aware systems. In C. Ghaoui (Ed.), *Encyclopaedia of HCI*. PA: IDEA Group Hershey.

Sibert, L. E., & Jacob, R. J. K. (2000). Evaluation of eye gaze interaction. In *Proceedings of the Human Factors in Computing Systems Conference (CHI'00)* (pp. 281–288). New York: ACM Press.

Sibert, L. E., Goturk, M., & Lavine, R. A. (2000). The reading assistant: eye gaze triggered auditory prompting for reading remedation. In *Proceedings of the 1st Annual ACM SIGGRAPH symposium on User Interface Software (UIST'00)* (pp. 101–107). New York: ACM Press.

Stampe, D. M., & Reingold, E. M. (1995). Selection by looking: a novel computer interface and its application to psychological research. In J. M. Findlay, R. Walker, & R. W. Kentridge (Eds.), *Eye movement research: Mechanisms, processes and application* (pp. 467–478). New York: Elsevier Science Publishers.

Stiefelhagen, R., Chen, X., Yang, J. (2003). Capturing Interactions in Meetings with Omnidirectional Cameras, International Workshop on Multimedia Technologies in E-Learning and Collaboration (Womtec), Nice, France.

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.

Tennenhouse, D. (2000). Proactive computing. *Communications of the ACM, 43*(5), 43–50.

Tobii, Tobii Technology, information available at http://www.tobii.se/.

Vertegaal, R. (2003). Attentive user interfaces. *Communications of the ACM, 43*(3), 31–33.

Zhai, S. (2003). What's in the eyes for attentive input. *Communications of the ACM, 43*(3), 34–39.

Zhai, S., Steven, C., & Ihde, S. (1999). Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the Human Factors in Computing Systems Conference (CHI'99)* (pp. 246–253). ACM Press.