Design Issues of iDict: A Gaze-Assisted Translation Aid

Aulikki Hyrskykari*
Päivi Majaranta*
Antti Aaltonen*
Kari-Jouko Räihä*

University of Tampere

Abstract

Eye-aware applications have existed for long, but mostly for very special and restricted target populations. We have designed and are currently implementing an eye-aware application, called iDict, which is a general-purpose translation aid aimed at mass markets. iDict monitors the user's gaze path while s/he is reading text written in a foreign language. When the reader encounters difficulties, iDict steps in and provides assistance with the translation. To accomplish this, the system makes use of information obtained from reading research, a language model, and the user profile. This paper describes the idea of the iDict application, the design problems and the key solutions for resolving these problems.

CR Categories and Subject Descriptors: H.5.2 [Information Systems]: Information Interfaces and Presentation/User Interfaces - Input device, Interaction styles

Additional Keywords: gaze, eye tracking, non-command interfaces, post-WIMP interfaces, input techniques

1 INTRODUCTION

Recent interest in widening the channel between the computer and the user has intensified research efforts on new input modalities, such as gestures, voice, and eye gaze. Compared to the use of any other input channel, eye gaze has a unique property: it is the only input modality that implicitly carries information on the focus of the user's attention at a specific point in time. This may be essential information for many applications. Consider, for example, how useful it would be for a computer-aided instruction program to know which parts of a page or scene the learner has reviewed when s/he turns to the next phase of the program, and especially which parts s/he has ignored. Eye gaze data can also reveal where the gaze path differs from the expected. For certain applications – like iDict – this may indicate that the user has problems with that specific spot.

Monitoring the eye movements increases the computer's "consciousness" of the user's state and consequently helps in designing an interface that reacts in a more natural way. The more natural an interface gets, the more transparent it appears to the user: the user starts to interact with the task instead of interacting with the computer [12].

The use of gaze as an input channel is limited by the inaccuracy of the measured point of gaze. This is derived partly from the incapability of the hardware to acquire the exact point of gaze and maintain the accuracy during the session. Fortunately, the development of eye tracking technology has considerably decreased the magnitude of this problem. However, a more permanent restriction stems from the biological characteristics of the eye. The size of the fovea limits the accuracy of the measured point of gaze to 0.5 degrees. Therefore, compared to *e.g.* mouse coordinates, eye gaze data will always be less accurate and in this sense "noisy".

iDict is a translation aid for promoting fluent reading of foreign language documents with the help of gaze path information. In our prototype the language of the documents is English, so the target population is non-native readers of English. The key means for filtering out the useful information amongst the noisy eye movement data is the use of additional information. The supporting information includes:

- general knowledge of the reading process based on the results of reading research,
- lexical and syntactical language analysis of the text for obtaining information on the potentially unfamiliar words, phrasal verbs or idioms and syntactically complex structures, and
- user profile data containing knowledge of the user's personal characteristics as a reader and history data of the user's reading process in the past.

The rest of the paper is structured as follows. We first briefly review eye tracking research, focusing on studies where eye gaze data is interpreted in real time. We then discuss how iDict operates, and continue with a more detailed description of how iDict makes use of the various sources of information.

*Human-Computer Interaction Group, Department of Computer and Information Sciences, FIN-33014 University of Tampere, Finland
tel. +358-3-215{6776, 8550, 8556, 6952} email: {ah, curly, aa, kjr}@cs.uta.fi www: http://www.cs.uta.fi/research/hci/ieye

Eye Tracking Research & Applications Symposium 2000
Palm Beach Gardens, FL, USA
© 2000 ACM ISBN 1-58113-280-8/00/0011...$5.00

2 PREVIOUS RESEARCH

The history of eye tracking goes back at least to the sixties. The traditional way of exploiting eye tracking technology has been psychological research for testing empirically human perceptual or cognitive processes [28]. Likewise, eye tracking is well fit for testing software usability or the effectiveness of marketing materials and newspaper layouts [4], because those tests are typically set up in laboratory environments. Another related line of research is the study of how users perceive and work with the graphical user interface [1].

Common to all these applications is the a posteriori analysis of the eye tracking data saved during a session. Using the eye movement data in real-time to control or adapt the user interface is a much younger trend.

Gaze controlled applications have mainly been designed for disabled people [6]. Many times in those applications simple on/off information has been adequate and cheap enough, because it is easily accomplished for example with eye blinks. Some systems utilizing line of gaze have been used as well, most notably for typing with the eyes. This is particularly useful and more feasible for subjects who are not able to move their heads, since gaze tracking is then easier and cheaper [5].

The use of eye gaze as a pointing device is the most straightforward approach when we consider using gaze in real time to control the interface in normal human-computer interaction. However, a review of the experiences in using this approach exposes its limitations.

Jacob has conducted an extensive set of experiments in which gaze is used as the main control device for controlling conventional interface components [8, 9, 18]. He experimented with different gaze-assisted methods for selecting an object, eye-controlled pull-down menus, dragging of objects, and gaze scrollable text.

When eye gaze is used as a device that activates commands we meet the so-called "Midas touch" problem: everywhere the user looks, the gaze position is interpreted to have some kind of intention. This may lead to annoying situations. For example, Glenstrup and Engell-Nielsen reported that when testing the EyeCatcher application [5], the users were consciously trying not to look at the parts of the screen where eye-controlled buttons resided.

Eyes are very rapid in searching for an object, but hand gestures perform better with tasks requiring accuracy. This observation resulted in the implementation of Magic pointing [30] that combines the benefits of both modalities. Similarly, Salvucci takes advantage of both modalities in his gaze-assisted IGO operating system [16].

The experiments described above, as well as some others (*e.g.*, [26]) lead us to conclude that even though gaze is potentially a powerful way to control an application, it is not suitable to be used as the main input device. After all, we do not want to restrict the natural use of eyes as a sense for observing and gathering information. More promising is the possibility to use the acquired eye movement data as an additional information source for enhancing the function of the interface. The information about the user's actions (like browsing and reading) as well as the actual point of attention is very useful data for many applications.

Accordingly, the connection between a person's eye movements and intentions or thoughts has gained a lot of attention recently. For example, Edwards is designing the Eye Interpretation Engine [3] that will interpret the user's natural eye-movements, and based on that tries to categorize his/her behavior. His goal is to recognize:

- when the user is searching for a target,
- when the user knows where a desired target is located, and
- when the user is looking around the screen with no intention to select anything.

In his dissertation, Salvucci [14, 15] also concentrates on mapping the eye movements to cognitive processes.

Other applications that have used gaze as an additional input jointly with other input sources are, for example, the Little Prince Storyteller application [19], the GAZE Groupware System [24, 25], and a gaze-assisted language instruction system [20, 21]. The latter application takes advantage of eye tracking in an environment where a student is taught to translate English sentences into Japanese. That application shares many problems and solutions with ours, and has acted as an inspiration for iDict.

3 I-DICT

iDict is a translation aid that tracks the user's eye movements for inferring when and what kind of help the reader needs while s/he is reading a document written in a foreign language. The application is designed for a wide target population. Therefore it is crucial that the use of eye movement information is implemented in a smooth and acceptable way.

The starting point for developing iDict was to find new forms of interaction that make use of natural eye movements. In other words, iDict uses the information from normal eye movements rather than requires the user to perform any special actions with her/his eyes.

The planned operation of iDict is that when a user reads an on-screen document, the system tracks the reading process and detects the situations where the user seems to have troubles. iDict then concludes what is the help needed, and delivers it to the user. In the prototype, the language of the text in the document window is in English, and help is provided as translations into Finnish, German and Italy (depending on the user's choice). Initially, the text is assumed to be everyday English. The possibility to work with special vocabularies (such as legal or technical text) will be studied later.

The operation of iDict is illustrated in Figure 1, where the gaze path is, as customary, drawn as circles connected by straight lines. Of course, the gaze path is not shown to the user, it is only displayed here for illustration. The topmost frame in Figure 1 shows normal reading behavior. In the middle frame, the reader has realized that s/he does not understand the word "gaze", and regressed back to that word, with many long fixations in the proximity of the word.

The method for delivering assistance to the reader is beyond the scope of this paper. There are many possibilities [29], ranging from various visual forms to synthesized speech. Several methods will be implemented and compared in usability tests.

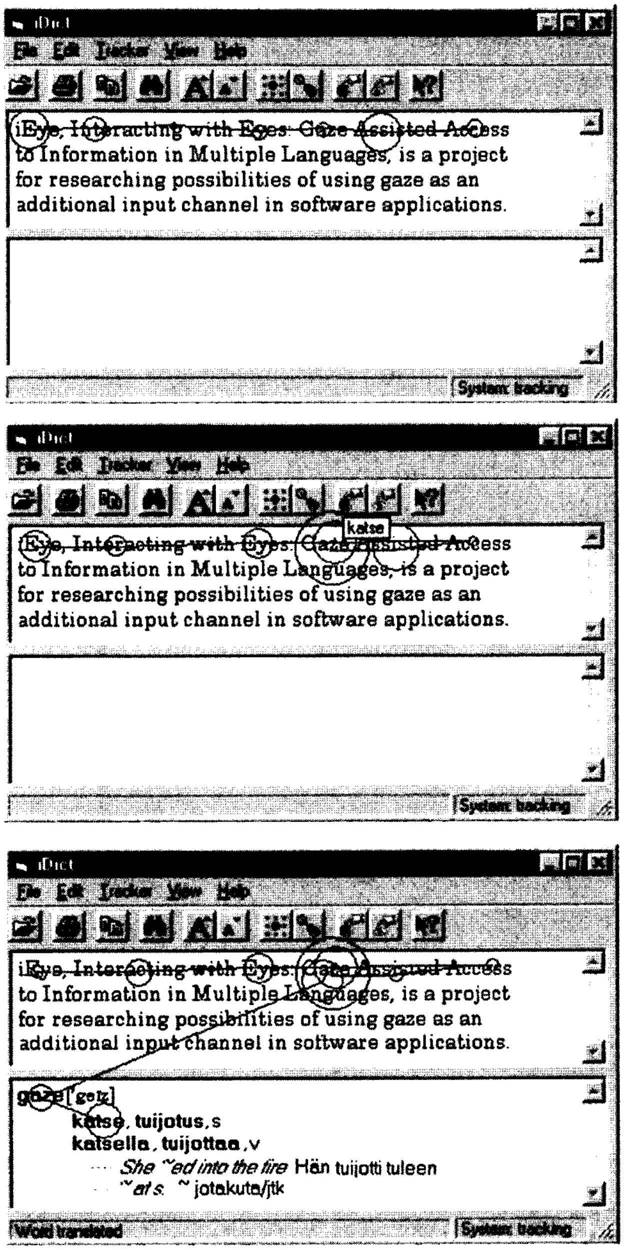

Figure 1. The iDict prototype.

Figure 1 illustrates our first shot at this issue. Immediate, minimal help is provided as a tooltip very soon after detecting that the user is having problems. It contains the "best guess" translation for the word (or phrase or idiom) that we can produce by combining the gaze path data with the lexical and syntactic analysis of the text. The tooltip disappears when the user moves on – either proceeding with the reading process, or by moving his/her gaze to another window (see the bottom frame in Figure 1). When gaze hits that window, more detailed information on the word or phrase appears.

The prime problems in designing and implementing eye-aware applications and thus also iDict arise from the difficulties of interpreting the noisy eye movement data.

First, in order to find out the real gaze path we must filter out the inaccuracies originating both from the characteristics of the tracker and the biological characteristics of the eye.

Second, after we have filtered the gaze path information reliably enough, we still need additional knowledge for the interpretation of the user's state in the reading task. We will use additional information from (1) the reading process, (2) the language model of the text, and (3) the user's profile data gathered during reading sessions. The information acquired from these three sources is described in more detail in the next section.

The approach outlined above requires a variety of knowledge and skills. In the consortium that is developing iDict, we co-operate with an eye tracker manufacturer, psychology researchers and language engineers.

The iDict system is designed to run on a normal desktop computer (*e.g.*, Intel Pentium III processor, 128Mb RAM) with a USB (Universal Serial Bus) connector. The eye tracker is developed by SensoMotoric Instruments, GmbH (SMI) [17]. In the development phase we use the very accurate EyeLink system, with a temporal resolution of 250 Hz. However, the target hardware is a new inexpensive version fit for mass production. The temporal resolution is then 50 Hz and the accuracy at least 1 degree.

4 READING AND EYE MOVEMENTS

4.1 Normal Reading

Previous studies have shown that eye movements reflect thinking processes [10, 11, 20]. Generally, a fixation reveals the reader's attention and interest; one's eyes reveal what one is thinking about. The actual retrieval of information (reading) only happens during fixations. No information is acquired between fixations, during the saccadic eye movements [27].

We can follow the reading process by tracking the eye movements. There is a strong correlation between the number of words and the number of fixations in a sentence: a good reader fixates on almost every word. Short, common words may be skipped, but long words usually require at least two fixations [7, 13].

Fixations tend to land approximately in the middle of the word. The parafoveal field of vision guides the saccades. The reader can focus his/her attention on the next word without actually fixating on it. The length and shape of the next word is estimated before the actual saccade takes place. Also, micro saccades within a word can be used to fine-tune the landing of a saccade [13, 23].

The duration of a fixation reflects the relative difficulty of the word. A reader fixates longer on words that have low frequency. Familiar, high frequency words require only a short fixation [7, 23].

Regressions indicate comprehension failure. The reader has to return to the beginning of the sentence or a phrase and re-read it. Typically, difficult or poorly written text acquires a lot of regressions, and therefore, takes a lot of time to process. However, regressions are not always related to difficulties. A good reader returns to the previously read parts of the text to make cross-references and to refresh the important information.

4.2 Detecting Problems

Eye movements reflect the cognitive burden during the processing of the text. Long fixations and regressions reveal difficult lexical access. In iDict, we want to provide relevant assistance when the user needs it. By analyzing the eye movements during the reading process, we should be able to detect when and where the user has problems. For this, we need to be able to distinguish situations that resemble each other in many respects.

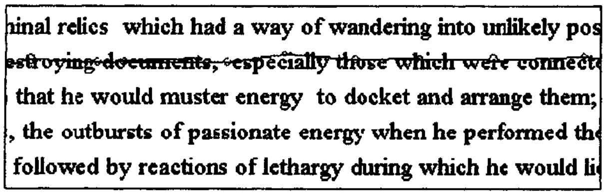

Figure 2. Fluent reading.

First, we must be able to separate fluent, uninterrupted reading (Figure 2) from the case in which the user has problems (Figure 3). In addition to reading, the user may also stop to think about the things s/he has read. Thus, we cannot say that the user has problems whenever the fluent reading pauses. The eyes may wander around the screen (and text). It is also possible that the user searches for something in the text by scanning through it without actually reading it. Furthermore, re-reading a phrase does not always mean that the reader did not understand it. How can we filter out such behavior? How can we tell what the user is thinking, and more importantly, when the user is having trouble in understanding the text and needs help with the translation?

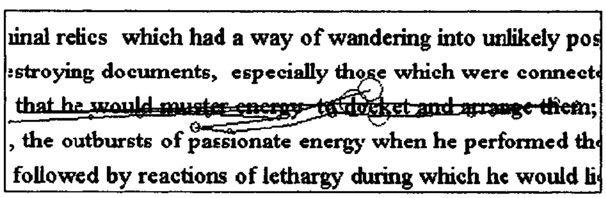

Figure 3. The reader has problems with the phrase "muster energy" and the word "docket".

Our solution to the problem is to combine information about the lexical content of the text with the information we get from analyzing the eye movements. We detect patterns from the reader's behavior. Patterns can be used to separate reading from other activities.

4.3 Detecting Patterns

There are at least four kind of patterns in the user's behavior when s/he processes text.

1. Reading

The user's attention is focused on the text. The direction of the eye movements proceeds from left to right and follows the layout of the text on the screen. The duration of fixations stays within predefined threshold values.

2. Scanning

The user gazes through the text. S/he may be searching or speed-reading the text. There are not enough fixations for the reader to be able to actually read the text. The direction of the gaze path does not necessarily follow the text.

3. Dormant Gazing

The user is neither reading nor scanning. The user may have paused to think about the contents of the text. In this case the user's attention is introverted. The eyes may or may not move; the user stares at the screen with "blank eyes".

4. Encountering Difficulties

The user has encountered an unfamiliar word that needs to be translated. The duration of fixations increases and/or the number of fixations within and around the difficult word increases as the user tries to figure out the meaning of the word.

We assume that there is a detectable change in the gaze path, regression structure and fixations that match one of the patterns, even though it may not be easy to classify the observations. For example, scanning (for searching the text), re-reading (for cognitive processing of the contents of the text), and pausing at a difficult word may all produce quite similar behavior (*e.g.*, regressions). If we want to distinguish the meanings of various regressions, we have to continuously observe the user's actions to learn his/her reading style. That information needs to be combined with the linguistic analysis of the underlying text.

4.4 Lexical and Syntactic Analysis

An important part of the iDict application is a module for analyzing the text. The lexical module (provided by Conexor Oy) parses the text [22]. It detects word classes, differentiates words and compounds, and decomposes reformed words into their basic forms. It also detects idioms and phrases.

This module helps us in two ways: it can be used to give more accurate and appropriate translations, and it can be used to detect when that assistance is needed in the first place.

First, the lexical analysis helps in making queries to the translation dictionaries. The queries are more likely to produce the desired result when we know the basic form of the word and the word class. When the fixations and regressions gather around a phrase, we can offer a translation of the phrase instead of separate translations of the words in the phrase.

Second, the lexical module can detect difficult text structures and can therefore suggest which part(s) of the text may be difficult to comprehend. For example, there may be sentences in the text that cause parsing problems to the reader. The reader has to re-read the sentence because the syntax does not make sense at first.

A classical example is the so-called garden path structure, such as the one in the sentence "Although Kari often jogs a mile seems like a long way."[1] The reader probably reads this sentence fluently till the word "seems", assuming that s/he has understood everything, but then "seems" does not fit in the pattern that s/he has formed. The reader has to backtrack to re-parse the beginning of the sentence.

---

[1] This garden path provided courtesy of Geoff Underwood.

The analysis module can locate the garden paths and other syntactically problematic items in the text, and mark the word(s) where the reader has to start recomposing the sentence. By combining the information about the marked word and the user's gaze path, iDict can figure out what happened. When the user stops at a word and makes a regression towards the beginning of the sentence, iDict knows that this time the regression is related to the garden path problem.

4.5 User Profile and Session History

Initially, iDict bases its analysis on general rules and facts about how the gaze behaves in certain situations. The general rules are modified to form a baseline for each individual reader by providing each new user an example text to read. As the user continues to use iDict, the program collects data from the user's behavior. It builds a user profile and tries to adapt to the user. The user profile should help in providing more accurate help.

iDict uses the user history both for adjusting the user profile and for the analysis of the user's actions. The user profile can contain simple information that the user can modify him/herself, such as the form and speed of getting help: some users might prefer a longer time for figuring the problems out themselves. The user profile can also be adjusted by the system, *e.g.*, to modify the threshold values for normal fixation duration. On the average, fixations during reading last for 200-250 ms, but the variation can be high [13]. If the system finds that for a given reader the average fixation takes, say, longer than usual, storing this information in the user profile will help in avoiding "false alarms" in cases where the user just wants to go on with reading.

For analyzing the user's behavior, iDict stores the last points of hesitation (long fixations and pauses). These can be used later if the user makes regressions to those items. Repetitive regressions to the same item(s) are an indication of difficulties. In our preliminary tests it was obvious that there were several types of readers. Some wanted to get immediate help and fixated for a long time on difficult words, whereas others continued reading in hopes of understanding the words later. If that turned out not to be the case, they regressed back to the problematic word several times, probably to refresh that problem point in their memory. Combined with information on word frequencies, detecting such behavior helps in deducing that the reader does not know the meaning of the particular word.

4.6 Putting It Together

We have described the use of various information sources for interpreting the intentions of the user during a reading process. During the development phase we will, of course, analyze all data carefully after reading sessions to come up with interpretations that are as accurate as possible. However, for the delivered product a big challenge is that it will have to work in real time. All analysis has to be carried out within fractions of a second.

Therefore the text will be preprocessed and indexed by the lexical module. It is extremely fast, handling several hundred lines in a second, so this will go unnoticed by the user during the opening of a document. The indexing phase will create a structure that allows easy and fast access to all parts of the text.

This structure can then be supplemented with, *e.g.*, word frequencies. This is further (dynamically) modified by the user profile: if the user is repeatedly shown the translation of an unknown difficult word, the difficulty of that word for the remainder of the text is gradually decreased. Expected durations for reading chunks of text can then be computed on the basis of word difficulty, knowledge from reading research, and the user profile. If actual times differ from this prediction by more than a given (user dependent) threshold, actions for delivering assistance are triggered.

The above is a simplified description of the actual operation. The processing of regressions has been omitted here, and the granularity of operation (word vs. phrase vs. line) will be decided during development. The key principle, however, is to condense the various sources of information into simple parameters, and to test that those parameters stay within given limits. This enables the fast, real-time operation needed for the application.

We will start with simple algorithms and develop them further as we get more knowledge through the tests. At first we will only try to detect the state when the user is reading. Then, within that state we will try to capture the situations when the reader is experiencing difficulties during reading.

5 CONCLUDING REMARKS

The use of eye gaze for controlling the computer is an attractive possibility. The use of eyes is natural and fast. They indicate the locus of attention: usually the eyes are watching the very same spot that is being manipulated by the mouse pointer. Yet there have been very few applications that make use of eye gaze as an active input channel.

Most current applications that use gaze are designed for people with disabilities – *e.g.*, people who can move only their eyes in order to communicate with other people. The iDict application is a gaze-assisted translation tool targeted for a much wider audience.

iDict is uni-modal in the sense that the use of the translation aid is based only on the eyes. For controlling the document window we will start with a normal mouse controlled interface, but we will also experiment with eye controlled scrolling.

Nevertheless, iDict has similarities with multi-modal interfaces, because of the many different information sources used in interpreting the eye movement data [2]. These sources (knowledge from reading research, linguistic model of the text, the user's personal reading history) differ from typical multi-modal interfaces in that they are not actively produced by the user. However, they share the property that many information streams are combined to disambiguate information that alone would be ambiguous and difficult to interpret.

The status of iDict is that the design phase has been completed. Wizard-of-Oz studies were made to guide the design (and to give scan path data, like those in Figures 2 and 3). Implementation is taking place at the moment, and the first prototype is due by the end of the year.

Acknowledgements

The iDict system is developed within the project iEye: Gaze-Assisted Access to Information in Multiple Languages, funded by the European Commission via the Information Society

Technologies Programme (IST) within the Fifth RTD Framework Programme.

We would like to thank the other partners of the iEye consortium: Conexor Oy (Finland), GIUNTI Interactive Labs S.r.l. (Italy), SensoMotoric Instruments GmbH (Germany), and University of Nottingham (England).

References

1. Aaltonen, A., Hyrskykari, A., and Räihä, K.-J., 101 spots, or how do users read menus? *Proc. CHI'98*, Addison-Wesley, 132-139.
2. Bolt, R.A., Put-That-There. *Proc. SIGGRAPH'80*, ACM Press, 262-270.
3. Edwards, G., A tool for creating eye-aware applications that adapt to changes in user behaviors. *Proc. ASSETS'98*, ACM Press, 67-74.
4. Garcia, M., 10 Universal newspaper design myths, debunked. Available at http://www.poynter.org/Visual/designmyths.html.
5. Glenstrup, A.J. and Engell-Nielsen, T., *Eye Controlled Media: Present and Future of State*. Thesis in the Institute of Computer Science, University of Copenhagen, 1995. Available at http://www.diku.dk/~panic/eyegaze/article.html.
6. Hutchinson, T.E., White, K.P. Jr., Martin, W.N., Reichert, K.C., and Frey, L.A., Human-computer interaction using eye-gaze input. *IEEE Transactions on Systems, Man, and Cybernetics 19*, 6 (1989), 1527-1533.
7. Hyönä, J., Niemi, P., and Underwood, G., Reading long words embedded in sentences: Informativeness of word halves affects eye movements. *Journal of Experimental Psychology: Human Perception and Performance 15* (1989), 142-152.
8. Jacob, R.J.K., 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 (April 1991), 152-169.
9. Jacob, R.J.K., Eye tracking in advanced interface design. In W. Barfield and T. Furness (eds.), *Advanced Interface Design and Virtual Environments*. Oxford University Press, Oxford, 1995. Available at http://www.eecs.tufts.edu/~jacob/papers/barfield.html.
10. Just, M.A. and Carpenter, P.A., A theory of reading: From eye fixations to comprehension. *Psychological Review 87*, 4 (1980), 329-354.
11. Monty, R.A. and Senders, J.W., *Eye Movements and Psychological Processes*. Lawrence Erlbaum, Hillsdale, N.J., 1976.
12. Nielsen, J., Noncommand user interfaces. *Comm. ACM 36*, 4 (April 1993), 82-99. Available at http://www.useit.com/papers/noncommand.html.
13. Rayner, K., Eye movements and cognitive processes in reading, visual search, and scene perception. In J. Findlay *et al.* (eds.), *Eye Movement Research: Mechanisms, Processes and Applications*, 1995, 3-22.
14. Salvucci, D.D., Inferring intent in eye-based interfaces: Tracing eye movements with process models. *Proc. CHI'99*, Addison-Wesley, 254-261.
15. Salvucci, D.D., *Mapping Eye Movements to Cognitive Processes*. Doctoral Dissertation. Department of Computer Science, Carnegie-Mellon University, 1999.
16. Salvucci, D.D. and Anderson, J.R., Intelligent gaze-added interfaces. *CHI Letters 2*, 1, *Proc. CHI'2000*, Addison-Wesley, 273-280.
17. SensoMotoric Instruments GmbH. Information available at http://www.smi.de/el/index.html.
18. Sibert, L.E. and Jacob, R.K.J., Evaluation of eye gaze interaction. *CHI Letters 2*, 1, *Proc. CHI'2000*, Addison-Wesley, 281-288.
19. Starker, I. and Bolt, R.A., A gaze responsive self disclosing display. *Proc. CHI'90*, Addison-Wesley, 3-9.
20. Takagi, H., How to detect higher-order thinking processes and knowledge states from eye-movements. *Proc. PUI'97* (Workshop on Perceptual User Interfaces), 1997. Available at http://nicosia.is.s.u-tokyo.ac.jp/members/hironobu/pui/pui.html
21. Takagi, H., Development of an eye-movements enhanced translation support system. *Proc. APCHI'98* (Asian-Pacific Computer-Human Interaction Conference), 114-119.
22. Tapanainen, P. and Järvinen, T., A non-projective dependency parser. *Proc. ANLP'97* (Conference on Applied Natural Language Processing), 64-71.
23. Underwood, G., Clews, S., and Everatt, J., How do readers know where to look next? Local information distributions influence eye fixations. *Quarterly Journal of Experimental Psychology 42A* (1990), 39-65.
24. Vertegaal, R., The GAZE groupware system: Mediating joint attention in multiparty communication and collaboration. *Proc. CHI'99*, Addison-Wesley, 294-301.
25. Vertegaal, R., *Look Who's Talking to Whom*. Doctoral Dissertation. Cognitive Ergonomics Department, University of Twente. 1999.
26. Ware, C. and Mikaelian, H.T., An evaluation of an eye tracker as a device for computer input. *Proc. ACM CHI+GI'87*, ACM Press, 183-188.
27. Wolverton, G.S. and Zola, D.A., The temporal characteristics of visual information extraction during reading. In K. Rayner (ed.), *Eye Movements in Reading: Perceptual and Language Processes*, Academic Press, New York, 1983, 41-51.
28. Yarbus, A.L., Eye movements during perception of complex objects. In L.A. Riggs (ed.), *Eye Movements and Vision*, Plenum Press, New York, 1967, Chapter VII, 171-196.
29. Zellweger, P., Mackinlay, J., Chang, B.-W., and Regli, S.H., The impact of fluid documents on reading and browsing: An observational study. *CHI Letters 2*, 1, *Proc. CHI'2000*, Addison-Wesley, 249-256.
30. Zhai, S., Morimoto, C., and Ihde, S., Manual and gaze input cascaded (MAGIC) pointing. Proc. CHI'99, Addison-Wesley, 246-253.