

Detailed Monitoring of User's Gaze and Interaction to Improve Future E-Learning

Heiko Drewes[#], Richard Atterer[#], and Albrecht Schmidt[‡]

[#] Media Informatics Group, Ludwig-Maximilians-University München, Germany

[‡] Fraunhofer IAIS, Sankt Augustin and B-IT University of Bonn, Germany

{Heiko.Drewes, Richard.Atterer}@ifi.lmu.de, Albrecht.Schmidt@acm.org

Abstract. In this paper, we investigate how to use future interaction technologies to enhance learning technologies. We examine in detail how tracking the mouse pointer and observing the user's gaze can help to monitor the use of web applications and in particular E-learning applications. To improve learning and teaching, it is of interest to understand in what order and to what extent users read texts, how much time they spend on individual parts of the teaching materials, and where they get stuck. Based on a standard web browser as an application platform, extended with a gaze tracking facility, we conducted studies to explore the feasibility of this novel approach. The concept includes an extension of current technologies to allow JavaScript code running in the browser to access the current gaze position of the user. Our work shows how pieces of web technology and eye gaze tracking can be put together to create a new platform for E-learning that provides additional benefits for learners and teachers.

Keywords: User monitoring, eye-gaze user interfaces, interaction tracking, reading detection, E-learning

1 Introduction and Motivation

For learners and teachers, the use of electronic media has become common in education. E-learning systems are available for many different domains and are often used in addition rather than as replacement for traditional media (e.g. books and exhibits). The provision of multimedia content in an appropriate form has been a focus in research over the last decade. Systems often combine the presentation of static information (e.g. text, graphics and photos), animated information (e.g. dynamic graphics and movies) and interactive elements (e.g. simulations, navigation options). Additionally, means for assessing the progress in learning are included (e.g. tests at the end of a section or chapter). Monitoring the use of online learning materials is a basic source to understand what the learner has seen and what learning strategies she or he follows. Traditional and web-based systems achieve this at the page level by logging the URLs that have been requested and how long the user spent on certain pages. However, the information received is rather coarse as a single page may contain text, images and photos. Similarly, it is hard to tell how long a person looked

at the page, as the interval between accessing two URLs does not necessarily yield information on how long the person has been reading or interacting with the page.

Novel technologies such as gaze-tracking systems that allow the normal use of a desktop system while recording the user's gaze path are becoming widely available. Additionally, WWW technologies such as AJAX create new options for detailed tracking of interaction with digital media [1] using a standard web browser. We combined both approaches to research usability. In this paper, we show that these technologies also offer new means for the creation of novel user interfaces for E-learning and E-teaching.

Current WWW technology provides a platform-independent format to deliver and display content, which is accessible from everywhere. For this reason, we expect that a majority of E-learning application will make use of web technologies for content provision and delivery, even if used from a DVD or on the intranet. Research to utilize the web as a platform for E-learning is manifold, for an example see [12].

Camera devices will become a standard hardware for computers in the same way that speakers are now. Many new computers and mobile phones already have build-in cameras. As processor power and memory is also steadily increasing, we expect that gaze tracking technology will be available in the future at little extra cost.

Previous research exists on how to use the gaze information to model the user's meta-cognition and use this for intelligent pedagogical agents [7]. Our approach presented in this paper follows a more pragmatic approach and looks at concrete enhancements for E-learning systems.

In the remainder of the paper, we first analyze potential scenarios of future E-learning. The scenarios are the basis for the implementation of a framework. A technological proof of concept for general tracking technologies and gaze tracking in particular shows the feasibility of the approach. We include one specific example where we developed an algorithm, which uses gaze-tracking technology for reading detection.

2 Scenarios

Detailed monitoring of interaction and gaze for the presented learning material can give benefits to both learners and teachers. The following scenarios show how they benefit from the combination of eye gaze tracking and web activity tracking.

2.1 Author and Teacher Benefits

Using the envisioned system, the author or content creator has additional options to monitor the system's use by the learner at her or his disposal. For every object (e.g. text, graphics, animations), the author can decide to record the time the user gazed on an object as well as the order in which elements have been read or watched. For text objects, there are options to make reading compulsory (e.g., continuing is only

possible once the text has been read) and to record the intensity of reading (e.g. was the text skimmed, read word by word, or not read at all).

Similar to the `onmouseover` event in JavaScript, there is also an `ongazeover` event to trigger other activities. For example, if there is a diagram with explaining text, a part of the diagram can be highlighted depending on the keyword that was just read in the text. On a similar note, when looking for a while at a detail of the diagram, the corresponding explanatory text is highlighted. Jacob [11] initially investigated such approaches in 1990. He displayed a sea map with positions of ships and information on an individual ship, which corresponded to the gaze position of the user.

As soon as a lesson is online, the teacher has access to statistics how much time the learner's gaze spent on each page and on elements in the page. This information is available for individual learners and accumulated for a whole course. The value is much more reliable than the time the lesson was displayed on the screen because any time the learner spent e.g. at the coffee machine or chatting with colleagues does not contribute to the time spent for the lesson. The report also states how well the user read a text.

All this knowledge on the use of materials and the learner's behavior can help the teacher to improve the learning material, creation of tests, and the revision of teaching materials. If information is available on how long the user needed to read a certain paragraph, whether she or he re-read it multiple times or just skimmed through it, this can provide the basis for deciding on the inclusion of additional material, rewriting of paragraphs or restructuring of the course. The order in which users look at material, the general behavior (e.g. first skimming, then reading) as well as the viewing times of pictures, graphics and photos can give additional information that helps to restructure the online teaching materials or to create questions on comprehension.

2.2 Learner benefits

To make online learning more efficient and to personalize the presentation of information tracking, the user's learning behavior in detail can provide essential clues.

When learners log onto the E-learning system, they also get statistics on how much time they spent on different lessons up to now. Texts provide feedback (e.g. through changes of the background color) on how intensively they have been studied so far. This makes it easy to go on when interrupted during learning, especially with lengthy texts. It also gives the freedom to work through the material in any self-chosen order without forgetting something.

When looking at details of the material, explanations appear automatically – there is no need to grab the mouse and position the cursor over the object of interest. Animations and videos start automatically the moment the learner looks at them, and stop the moment the gaze moves away [10]. This minimizes distraction during reading as animations and videos can pause when not looked at.

3 Future Technologies for E-Learning

3.1 WWW and AJAX

The worldwide web has been used as a way of providing services for many years. Whereas at first, the emphasis was mainly put on providing information to users of the WWW for their consumption, and to give access to this information via search facilities, in recent years web-based applications have gained more and more importance. Together with the types of tasks the web is used for, the technologies behind the WWW have been improved. Today, browsers are capable of executing complex JavaScript programs efficiently, and technologies like AJAX make web-based GUIs more similar to the GUIs of desktop programs. Thus, the web is now well suited for more types of applications, including remote E-learning.

E-learning systems of the future will be able to use an established application platform (the web browser) as their development target, and the users will benefit from the advantages of this platform, in particular its independence of operating system, screen resolution and standardized technologies.

We assume that in future things like key presses and mouse movements, which are available to web applications today (in the form of JavaScript `onkeypress`, `onmousemove` etc. events), will have been extended to include support for the capture of eye gaze data in future browsers. Previous work in this area concentrated on tracking mouse movements [13] and deducing from it the probable gaze direction of users in the absence of hardware eye tracking [1] [14].

3.2 Eye Gaze

Nowadays, eye-trackers work with a video camera in combination with an infrared LED. The light from the LED creates a reflection on the user's eyeball. As the eyeball is a nearly perfect sphere, the reflection stays stable on the same position, independent of the direction of the gaze. Image recognition detects the center of the pupil and after calibration (i.e. looking at the corners of the screen), the direction of the gaze can be calculated using the vector from the reflection point to the center of the pupil. As the detection of a black pupil inside a brown iris is difficult because of low contrast, many eye-trackers use the white-pupil-method: If infrared light illuminates the eye, the eye background (retina) reflects the light and causes a white pupil to appear for the infrared camera device.

To allow free head movement in front of the display, without use of a chin rest, the eye-tracker can be combined with a head-tracker. Head-trackers are also video-based. Commercial systems combining both technologies, such as the Tobii 1750 Eye Tracker¹ and the Eyefollower² are still expensive today, but work quite well.

¹ <http://www.tobii.com>

² <http://www.eyegaze.com>

One intrinsic problem of eye tracking is accuracy. The eye constantly performs micro-saccades which results in noisy position data. Another effect arises from the fact that the retina only has a small area of high resolution (fovea), which is about 1 degree of visual angle [8]. To see something clearly, the eye needs to move to bring the object of interest into this high-resolution area, but not necessarily into the center. On average desktop displays, the accuracy is around 30 pixels, which is about the size of standard icons and buttons, but twice the height of a line of text.

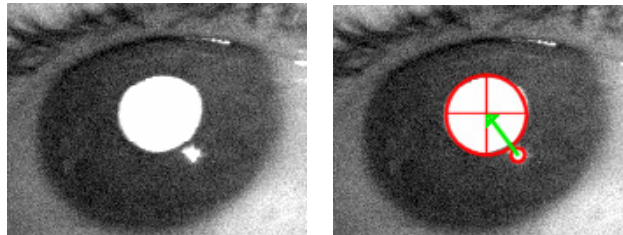


Figure 1: How eye gaze detection works. The small white spot is the reflection of the LED.

4 Reading

Psychologists intensively researched the reading process, which is understood in detail [2]. Nowadays, many researchers in the field of human computer interaction are interested in reading [6]. Practical applications are activity recognition (e.g. is someone reading, as presented in [3] and [4]) or a reading assistant introduced by [5].

Gaze-assisted E-learning systems need basic modules, so we focused on compulsory reading and on extracting values, which tells how intensively a text was read. In contrast to related work, the suggested algorithms can not only provide information on whether or not someone is reading, but also on which part of the text was read and how long the reading of a certain part took.

4.1 Compulsory Reading

One possibility to detect whether a text was read is to use the reading detection from [3] or [4], collect the x and y coordinates and find out whether these coordinates completely cover the text area. This will result in a rather complex algorithm and as the error rate for reading detection is still above 10 percent, a reliable result cannot be guaranteed – skimming the page would likely be detected as reading.

In an initial experiment, we recorded the eye gaze of several test readers. Figure 2 shows a typical result of our commercial gaze-tracking system called ERICA³.

³ <http://www.eyerresponse.com/>

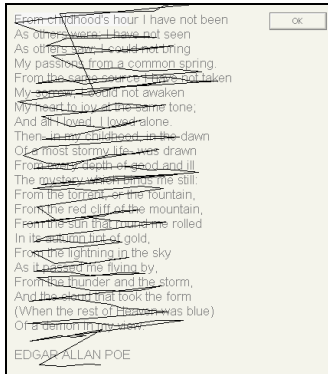


Figure 2: Gaze trail reading a text

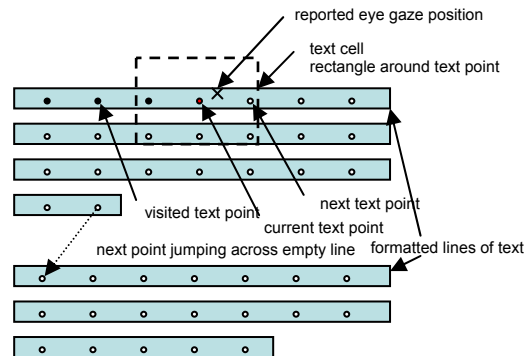


Figure 3: Illustration of the reading detection algorithm

We can see that the gaze moves in saccades with brief fixations and revisits. A preliminary evaluation of the data gives saccade lengths of around 30 pixels within a line and backward saccades, which are slightly smaller than the length of the line. The geometry of the system results in the 1° visual angle being 36 pixels wide. Thus, the angle of the forward saccades is about the same as the angle of the fovea, the area with high resolution on the eye's background [8]. It is interesting to note that consequently, the first fixation points in a line are about half a degree ahead of the beginning of the line, and the last fixation is the same distance away from the end of the line.

The algorithm presented here uses the fact that the gaze visits the text in sequential order. As traditional reading always works in sequential order independent of the scripture used, the algorithm can be applied to right-to-left scripts (Arab) and to vertical scripts (Chinese).

The basic parameters of the algorithm are the width w and the height h of the rectangle around the gaze position and the distance d of the text points. The algorithm starts with a point, which is $d/2$ from the beginning of the first line. As soon as this point is within a rectangular area around the reported gaze position, it is marked as read. Then the algorithm chooses the next point on the same line, d ahead of the previous one. In case of a line wrap, the next point is at the beginning of the next line. Once the gaze visited all points, the text was read completely.

By design, this algorithm allows the reader to take a break, perhaps to have a look at the system clock, before continuing to read. It is also robust against calibration shift because the dimensions of the rectangle can be relatively large without decreasing the reliability. A problem of this algorithm is the case that some words or lines are left out by the reader. For this reason, the system should provide feedback on the portion of text being read, for example by a change of the background color.

Another problem arises with very short texts, which can be read with one glance. To detect reading in a line of text there should be at least two, better three fixations. Together with the fact that the first and last fixation points are 0.5 degrees from the beginning and end of the line and that the fixations steps are around one degree, a line should have a minimum length of 3° visual angle. This is about 100 pixel on the system used.

4.2 Reading Intensity

For a teacher or pedagogical agent it can be of value to know how intensively a text was read and how much effort was required to read a part. There are two common ways to visualize the user's gaze. One method is drawing the gaze path (as done in Figure 2) and the other method is to use heat maps. Both methods provide too much information and it is difficult to find rules based on this data.

We introduce a method to reduce the data to a few values. The algorithm starts with formatting the text to find text points and surrounding rectangles. Every fixation inside the rectangle increases a counter for this area. It is also possible to accumulate the fixation time. This mechanism is like a low-resolution heat map. The data allows calculating two values, a mean, which tells the number of fixations per text cell, and the variance or standard deviation, which tells how uniformly the fixations are distributed over the text cells.

If there are n text cells and f_1, f_2, \dots, f_n are fixations for each cell, the intensity of reading I can be defined by the mean and a normalizing factor c :

$$I = \frac{c}{n} \sum_{i=1}^n f_i$$

Using the mean makes I independent from the length of a text. The factor c adjusts the value of I , so that it will be 1.0 if a text was read carefully once, and depends on the size of the text cell used in the algorithm. If the horizontal extent of the text cell is the length of an average forward saccade and the vertical extent is the height of a text line, c is 1. If the text cells are bigger and they overlap, one fixation is counted multiple times, because it is inside several text cells and the factor c will correct this.

To decide whether a text was read carefully, I is not enough, because reading half of the text twice results in the same value as reading the whole text once. Consequently, a second parameter D is necessary to know whether the gaze positions are distributed equally over the text cells. The variance provides such a parameter.

$$D = \frac{1}{n} \sum_{i=1}^n (cf_i - I)^2$$

If the text cells were all visited equally, D will be close to zero, otherwise D will be higher. A perfect read will result in $I = 1$ and $D = 0$.

5 Prototype and User Study

We conducted a user study on a stand-alone system to show that compulsory reading and detecting of reading intensity work reliably. To prove the technical feasibility of the concept, we then extended the functionality of our monitoring tool UsaProxy [1] to receive the eye gaze data via a separate TCP connection. Additional code evaluated the gaze data to generate the `ongazeover` events.

5.1 The Architecture of the Monitoring System

The special extensions to the JavaScript event mechanism outlined in section 3.2 do not exist yet. In order to be able to experiment with possible application areas and to evaluate whether our approach to E-learning applications works in practice, we designed an architecture that makes eye-tracking data available to the code running within a browser. Due to the technical restrictions of today's computers, a system needs to be specially prepared and the different components need to be set up independently, but this will not pose a problem with future, integrated systems, and from the point of view of the web application programmer, the environment is very similar to our envisioned future computer.

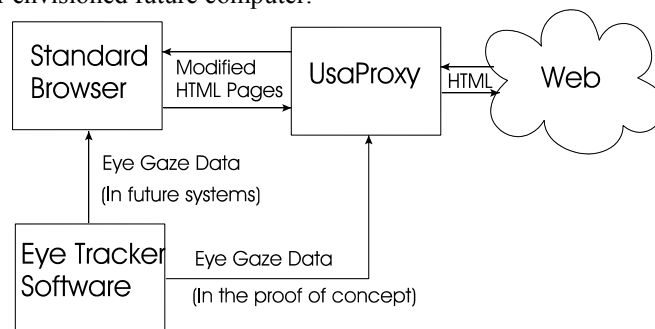


Figure 4: Overall design of the system

Figure 4 shows the overall design of the system: a standard web browser accesses the website, which provides the E-learning application; it can be located on any server on the Internet. In addition to the web browser, special eye tracking software is installed locally. Our UsaProxy HTTP proxy [1] is the key to make eye tracking data available to the browser: the web browser is configured to access the WWW via UsaProxy. UsaProxy simply passes on the requests of the browser and waits for the responses of the server. However, it modifies the HTML pages before passing them to the browser. The modification causes the browser to execute the JavaScript parts inserted by UsaProxy in the context of every page, regardless of which server it came from. Once it executes, this JavaScript code contacts UsaProxy, polling it for new events. This mechanism can be implemented in a way that cross-browser compatibility is possible.

The gaze tracking software that is also running on the local machine can now be interfaced to UsaProxy as well: via a special HTTP-based API, it can repeatedly contact UsaProxy and supply the current gaze position. UsaProxy can immediately pass this information to the browser.

5.2 The User Study

For the experiment on reading, we used the commercial low-cost gaze tracker ERICA that is commonly used in usability research and as an input device for the

disabled. The ERICA system does not allow free head movement, hence we used a chin rest for the experiment to stabilize the head position.

A group of nine people, six male, three female in the age from 23 to 47, took part in the user study. Every subject got three tasks. The first task presented a dialog with a poem of Edgar Allen Poe (23 lines, 150 words). The dialog presented feedback for the text already read and disappeared when totally read. The task was to read the text to make the dialog disappear. The second task was another dialog with a poem in the native language of the user (13 lines, 69 words), this time without providing feedback. The third task was a repetition of the second task, but this time with the demand to read the poem loudly.

5.3 Results

For the first task, all test users were able to let the dialog disappear (100%). Some of them complained that the feedback is faster than their reading and they felt being chased. In the second task, the dialog detected correctly complete readings in six cases (67%) and failed in the other three cases (33%). In the third task, the dialog correctly detected complete readings in five cases and failed in the other four cases.

The results of the second task were much better than expected, while the third task was worse than expected. Without visual feedback, the expectation was that users do not read carefully and the algorithm will not complete. Actually, the task given was to read the text, so the users did this carefully. Perhaps this would not be the case if another task had been given for which it is necessary to read something. The third task, reading loudly, had the intention to force the users to read the text slowly and carefully. However, the chin rest for head fixation obstructed speaking and the users moved their heads, which had a negative effect because the eye tracker calibration assumes a fixed head position.

A detection rate of 67% is sub-optimal and consequently compulsory reading should provide feedback, where the success rate is 100%. To detect whether a text was read carefully without feedback for the user, the values as introduced in 4.2 are more suitable.

When looking at the results, one should also consider the small font size. The font size of 20 pt used in our study was only slightly larger than the normal font of the system (16 pt), but much smaller than the font size of 48 pt used by Sibert, Gortuk and Lavine [5].

6 Discussion and Conclusions

We have outlined that novel technologies can provide new means for creating more effective user interfaces for E-learning systems. The scenarios show that having detailed knowledge of gaze and mouse movement will improve the experience for the learner and teacher.

Our user study shows that all components to build gaze-aware E-learning environments with fine-grained monitoring are already available. With falling prices

for eye-trackers and standards for eye-tracker integration into conventional systems, E-learning could change significantly in the future.

Our pragmatic approach of using the gaze is easy to understand by teachers and learners and consequently well accepted as an improvement of the interface. Further research to extract more knowledge from the user's gaze as presented in [9] is necessary, but it should be applied with care to keep the E-learning systems understandable for teachers and learners.

Acknowledgments. The work has been conducted in the context of the research project Embedded Interaction ('Eingebettete Interaktion') and was partly funded by the DFG ('Deutsche Forschungsgemeinschaft').

References

1. Atterer, R., Wnuk, M., Schmidt, A.: Knowing the User's Every Move – User Activity Tracking for Website Usability Evaluation and Implicit Interaction. In Proc. of the 15th International World Wide Web Conference (WWW2006), Edinburgh, Scotland, 2006
2. Reichle, E.D., Pollatsek, A., Fisher, D.L., Rayner, K.: Toward a Model of Eye Movement Control in Reading. *Psychological Review*. Vol. 105 No. 1, 125 - 127. 1998
3. Campbell, C.S., Maglio, P.P.: A Robust Algorithm for Reading Detection. In Proc. of the 2001 Workshop on Perceptive User interfaces. PUI '01. Vol. 15 ACM Press, 1-7, 2001
4. Keat, F.T., Ranganath, S., Venkatesh, Y.V.: Eye Gaze Based Reading Detection. TENCON '03. Conference on Convergent Technologies for Asia-Pacific Region. 825-828 Vol.2. 2003
5. Sibert, J.L., Gokturk, M., Lavine, R.A.: The Reading Assistant: Eye Gaze Triggered Auditory Prompting for Reading Remediation. Proc. of the 13th Annual ACM Symposium on User interface Software and Technology. UIST'00, 101-107, ACM Press. 2000
6. Nielsen, J.: Jakob Nielsen's Alertbox: F-Shaped Pattern For Reading Web Content, http://www.useit.com/alertbox/reading_pattern.html. April 17, 2006
7. Merten, C., Conati, C.: Eye-tracking to Model and Adapt to User Meta-cognition in Intelligent Learning Environments. In Proc. of the 11th international Conference on Intelligent User Interfaces. IUI '06, 39-46, ACM Press. 2006
8. Duchowski, T.D.: *Eye Tracking Methodology: Theory and Practice*. London: Springer-Verlag, 31-42. ISBN 1-85233-666-8. 2003
9. Iqbal, S.T., Bailey, B.P.: Using Eye Gaze Patterns to Identify User Tasks. The Grace Hopper Celebration of Women in Computing, 2004.
10. Dickie, C., Vertegaal, R., Sohn, C., Cheng, D.: eyeLook: Using Attention to Facilitate Mobile Media Consumption. In Proc. of the 18th Annual ACM Symposium on User interface Software and Technology UIST'05. 103-106. ACM Press. 2005
11. Jacob, R. J.: The Use of Eye Movements in Human-Computer Interaction Techniques: What You Look At is What You Get. 152-169. *ACM Trans. Inf. Syst.* 9(2). 1991
12. Shaw, E., Johnson, L. Ganeshan, R.: Pedagogical Agents on the Web. In Proc. of the Third Annual Conference on Autonomous Agents AGENTS '99. 283-290, ACM Press, 1999
13. Mueller, F., Lockerd, A.: Cheese: Tracking Mouse Movement Activity on Websites, a Tool for User Modeling. In Proc. of the Conference on Human Factors in Computing Systems CHI 2001 Extended Abstracts, Seattle, Washington, USA, 2001
14. Ullrich, C., Melis, E.: The Poor Man's Eyetracker Tool of ActiveMath. In Proc. of the World Conference on E-Learning in Corporate Government Healthcare and Higher Education eLearn-2002, 2313-2316, Montreal, Canada, 2002