

***e5Learning*, an E-Learning Environment Based on Eye Tracking**

Clara Calvi, Marco Porta^{*}, Dario Sacchi

Dip. di Informatica e Sistemistica, Università di Pavia

Via Ferrata, 1 – 27100 – Pavia (Italy)

clara.calvi01@ateneopv.it, marco.porta@unipv.it, dario.sacchi01@ateneopv.it

Abstract

Users' eyes can be a meaningful source of information for e-learning systems. What we look at, and the way we do that, can in fact be exploited to improve the learning process, disclosing information which would otherwise remain concealed. In this paper we describe an e-learning environment where eye tracking is used to observe user behavior, in order to adapt content presentation in real-time. To achieve such purpose, we consider both the way learning activities are carried out and those eye signals that can be related to the user's "emotional states".

1. Introduction

Computer-based teaching systems are a common reality nowadays. However, even if users are becoming more and more accustomed to interacting with machines, the perceived learning experience is often quite different from that characterizing interpersonal communication. In particular, the "emotional" part of the interaction between teacher and learner is missing [1, 2]. Like several other authors, we strongly think that "naturalness" is a key factor in e-learning [3,4]: a computer-mediated tutoring system provided with sensing capabilities could (at least partially) make up for the missing teacher in the flesh. Eye tracking, especially, can disclose important information concerning *what* the user is doing, as well as interesting data about *how* and *when* certain actions are being (or have been) performed.

Eye movements occur as sudden (almost instantaneous) *saccades*, followed by *fixation* periods of about 200-600 milliseconds, during which eyes are nearly still. Eye tracking technology has evolved very rapidly in the last years. Current commercially-available devices look almost like ordinary LCD screens, and al-

low a relatively high freedom of movements, thus not constraining users in their activities. Apart from costs, which remain very high, eye trackers are now a technology at hand, that can significantly improve the way we interact with the computer.

In this paper we present *e5Learning* (from enhanced exploitation of eyes for effective eLearning), an e-learning environment where eye tracking is used to allow the computer to get valuable data about users and their activities.

To date, only very few projects have explicitly considered eye tracking for e-learning. Among these, AdeLE (Adaptive e-Learning with Eye tracking) is probably the first, started with the main goal to dynamically capture user behavior based on real-time eye tracking [5]. Another interesting system is an empathic software agent interface developed to facilitate empathy-relevant reasoning and behavior, in which eye movements are used to get indications of learner interest and to provide feedback to character agents [6]. Other projects are more specifically focused on building models of user features, to properly shape the interaction. For instance, the system presented in [7] uses real time eye tracking information to try to assess student meta-cognitive behavior during the interaction with an "intelligent learning environment". Also noteworthy is *iDict* [8], a translation aid designed for language courses which exploits the user's eye movements to understand whether and when the reader needs help while reading a document written in a foreign language.

2. System Description

e5Learning is characterized by three main components: (1) a *Monitor of Accessed Screen Areas/History Recorder*, (2) a *Contextual Content Generator*, and (3) an *Emotion Recognizer*.

* Contact author

2.1. Monitor of Accessed Screen Areas/History Recorder

This component is actually formed of two sub-modules, used jointly in our implementation.

Thanks to the Monitor of Accessed Screen Areas, the author of the course can decide “how much attention” the user should pay to certain portions of content. A subject matter is usually composed of different kinds of media, such as text, images, animations, etc. Each part of the content is useful for the comprehension of the topic being described, but some elements, more than others, may be essential for the theme to be correctly understood. In general, we want the course creator to be able to specify, for rectangular areas, “how much attention” the user should pay to them.

In our prototype, a course is simply made up of web pages. We use an ad-hoc-built web browser which, along with page content, reads additional information defined by the author. Among other things, such information specifies the coordinates and sizes of screen rectangles corresponding to relevant portions of content, and associated data. We call these rectangles Regions Of Interest (ROIs). Using a special version of the browser, the course creator easily draws (with the mouse) ROIs around the areas that are to be monitored, as shown in Figure 1. There is no limit to the number of rectangles that can be defined.



Figure 1. Rectangles identifying ROIs and dialog box for parameter input

We distinguish two kinds of content, namely text and non-text. For textual areas, the system tries to understand whether they have been actually read by the user. Unfortunately, due to the limited precision of current eye trackers, an accurate check of the reading process is only possible for big text.

When line spacing is low, in fact, eye fixations on the different rows are hardly detectable, and we simply check that the progression of the eye scanpath within the content rectangle is roughly from left to right and from top to bottom, with right-to-left abrupt jumps. When line spacing is sufficiently high (about 1 cm), fixations on single rows can be exactly identified, and the check can be more precise; in the following, we will call this case “big text”, as opposed to “small text”. For non-textual areas, the course author simply specifies how much time the user should look at them (sum of the durations of the different fixations detected within content rectangles). Once a ROI has been drawn, an input window appears (Figure 1) which allows the author to enter such data.

The History Recorder submodule relies on the Monitor of Accessed Screen Areas and keeps track of which portions of content (ROIs defined by the author) have already been accessed by the user, as well as “how much”. For big text, the number of scanned lines is used to choose the best state, while for other elements more general criteria have to be employed. In particular, the complete reading of small text is checked through the detection of “reading patterns”, whereas the observation level of non-textual content is determined by fixation times.

Information about the state of a portion of content could be exploited both implicitly and explicitly. For example, if in a previous session the user did not devote sufficient time to a certain area, such portion might be subsequently proposed before others, independently of its position in the logical structure of the course. Another strategy, which is the one we have actually implemented, explicitly highlights the regions which need attention. When the user presses the ‘next’ button to load the next page in the course, if in the current page there are ROIs that have not been fully read/observed, the system emphasizes them through colored rectangles (Figure 2). Each ROI can be only in one of four possible states, namely *fully*, *partially*, *very little* and *not at all* accessed — corresponding, respectively, to green, yellow, orange and red rectangles. The only exception is given by ROIs with small text, which admit only the *read* (green) and *not read* (red) states.

With reference to a ROI, if V is the number of lines within it (“big text”) or a fixation time associated with it (non-textual content), then, indicating with v the value detected in real-time by the system, the state rectangle will be red if $0 \leq v \leq V/3$, orange if $V/3 \leq v < 2V/3$, yellow if $2V/3 \leq v < V$ and green if $v = V$. Even if not all the content has been fully accessed, however, the user can decide to proceed anyway with the next page, as we want the learner to always have the final word.

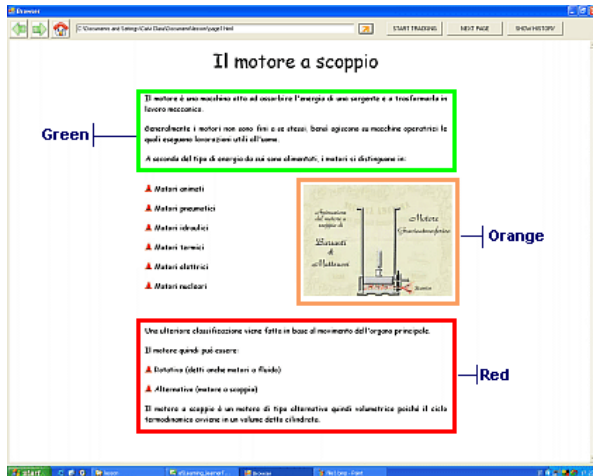


Figure 2. Colored rectangles highlighting ROIs which need attention

While on a page, at any time the user can also press a ‘Show history’ button to directly display the state rectangles. Moreover, since the history is saved in a file, even in subsequent sessions users can be explicitly reminded about their earlier activities.

2.2. Contextual Content Generator

The presentation of contextualized content depending on the main content being accessed at a certain moment by the user is another important feature characterizing *e5Learning*. The creator of the course can associate new content to ROIs, and indicate the requirements for the additional information to be displayed (in the form of HTML pages appearing within a popup window, as shown in Figure 3).

A condition for the new window to be shown is that the fixation time within a ROI is higher than a threshold. In this case, we can say that the accessory content is displayed when the user’s focus of attention is found within a corresponding region. For big text, the author can also decide to show the popup window when a repeated reading of the same lines is detected. In this second case, the additional content is typically used to provide further explanations to the user, who is probably not properly understanding something

As always, the course author creates ROIs for gaze-driven content presentation by simply drawing them over the pages displayed within the special browser. Once a rectangle has been sketched, the dialog box shown in Figure 1 appears. Besides the time threshold, such input pane allows the author to specify the URL of the new page to be displayed and the size and coordinates of the popup window that will contain it.

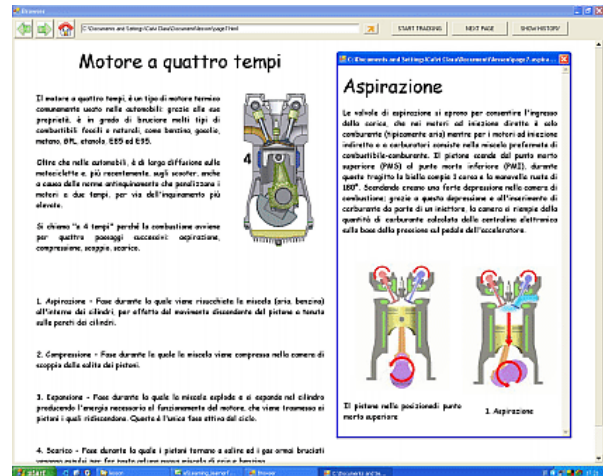


Figure 3. Additional content displayed when the user looks at a ROI

2.3. Emotion Recognizer

The importance of emotions, or “affect”, in e-learning has been stressed by several authors (e.g. [1] and [2]). Affective user interfaces applied to e-learning aim at improving learning environments by recognizing learners’ emotions and adapting their processes accordingly. From eye position tracking, and indirect measures such as gaze position, fixation numbers and fixation duration, it is possible to draw information about attention, stress, relaxation, problem solving, successful learning, etc. Especially in the Psychology and Physiology fields, several studies have been carried out which have tried to find correlations between eye behaviors and emotional states. Experiments have demonstrated that, for instance, pupil size is significantly larger after highly arousing stimuli than after neutral stimuli [9]. Also, pupil diameter seems to be task-dependent (being, for example, notably larger when “searching” than when simply “viewing” [10]). Other investigations (e.g. [11]) suggest that the mental workload can be assessed by analyzing the fluctuation rhythm of the pupil area. Indications about intellectual efforts can be also derived from blink rate and saccadic data. For instance, while pupil size and blink rate usually increase in response to task difficulty, both the saccade occurrence rate and saccade length typically decrease with the increased complexity of the task [12]

In our project we have practically considered all of these factors, trying to use them to detect signals that can be related to two main user conditions: (a) high workload or non understanding, and (b) tiredness.

For example, if the average pupil size has progressively increased within a certain time interval, also

user workload may have augmented. A decreased blink rate in the same period would further confirm such a supposition. When detected, such evidences could for example be used to dynamically modify the learning path, proposing a topic related to the main one but less complex (a sort of “break”). Or, if the user is potentially having problems in understanding something, extra information may be displayed. Since several external factors may come into play, however, it is practically impossible to be absolutely sure that these signals derive from changes in the user emotional state. Therefore, rather than undertaking direct actions, such as displaying help windows, we prefer to assist the user indirectly, with gradual aids. In our current implementation, when signs of non-understanding or high mental workload are detected, the system simply proposes links to additional material, which progressively enlarge as the signals of stress persist. When eye data suggest that the user may be tired, and the session has been going on for more than a configurable time interval (e.g. one hour), a message advising to take a break is shown.

We briefly describe now the strategies adopted in *e5Learning* to deal with the two potential user states that we take into account. To simplify the reference to the different cases in the next sections, we will label them with C_i ($i = 1..6$).

2.3.1. Cognitive load or understanding problems.

Within time intervals of t seconds, we check:

- the number of blinks n_b
- the number of fixations n_f
- the arithmetic mean of pupil diameters p_d

If, between two successive intervals t_{i-1} and t_i , a decrease of n_b (case C_1), an increase of n_f (C_2) or an increase of p_d (C_3) is noted, then in t_i the user may have gone through a high workload or non-understanding phase; the occurrence of more than one of these cases can be considered a further confirmation of that. In our experiments we have also noticed that a high increase of pupil size between two consecutive samplings of the eye tracker (C_4) is usually detected when the user is “disoriented” (e.g. because he or she is reading an unknown word).

2.3.2. Tiredness. Considering a number n_{int} of successive time intervals lasting t seconds, for each one of them we compute the arithmetic mean p_d of pupil diameters and the number n_b of blinks. If the last n_{int} values obtained for p_d are monotonically increasing (C_5), then this may be interpreted as a tiredness sign. To confirm such a possibility, we check if also the last n_{int} values obtained for n_b are monotonically increasing

(C_6). Once the first group of n_{int} intervals has been exploited, a new check of the above-described conditions occurs after $n_{int} - 2$ intervals; this way, temporal spans partially overlap, since the first two intervals of a new group are the last two of the previous one.

3. Implementation and Experiments

The system has been implemented in C#, within the .NET Microsoft framework. As an eye tracker, we use the Tobii 1750 (Figure 4), which integrates all the components (camera, infrared lighting, etc.) into a 17” monitor. With an accuracy of 0.5 degrees and a relatively high freedom of movements, the system is ideal for real-use settings, where it would be intolerable to constrain users. The sampling rate is 50 Hz (i.e., on average, eye data are acquired 50 times a second).



Figure 4. The Tobii 1750 eye tracker

We have informally tested *e5Learning* several times during its development. To get more rigorous data, however, we have also carried out a controlled experiment with 25 testers, aged between 19 and 28 (25 on average). Due to lack of space, here we will only present the main results obtained.

The course — which was about the “internal combustion engine”, providing both historical notes and descriptions of its functioning — was composed of 10 pages. Globally, it included 19 ROIs, and precisely: 5 regarding small text, 8 regarding big text and 6 regarding non-textual content (images and animated GIFs). 10 ROIs had contextual content associated with them, to be displayed when looked at for sufficiently long times (6) or because a repeated reading of the same lines of text was detected (4). With reference to subsections 2.3.1 and 2.3.2, the values for t and n_{int} were 120 sec and 6, respectively. As a “fixation” we considered a sequence of 10 consecutive samplings whose gaze coordinates were found within a circle with a radius of 30 pixels.

Each user was asked to read the pages, in sequence, trying to concentrate on the topic as much as possible. On average, a single test session lasted 27 minutes. ROIs containing small text were found in the “read”

state (green rectangle) in 92% of cases, while ROIs containing big text or non-textual elements provided the following percentages: 51% green (fully accessed), 13% yellow (partially), 17% orange (very little) and 19% (not at all). Since, after the test, users had also to answer multiple-choice questions about the topic of the course, we could verify that in more than 70% of cases wrong answers actually corresponded to red or orange ROIs, which were therefore not properly read.

Besides recording the occurrences of cases $C_1 \dots C_6$ (as well as all other events) in a log file, we also checked them in real-time, using an additional screen; this way, we were able to identify “interesting” situations as they happened. In particular, for the detection of high workload/non-understanding states, we considered the occurrence of C_1 , C_2 and C_3 (together, = C_{123}) and C_4 , while for the recognition of tiredness conditions we relied on the occurrence of C_5 and C_6 (= C_{56}). On average, C_{123} was detected 2.56 times per tester, C_4 2.2 times and C_{56} 0.44 times (such low value for C_{56} is probably due to the fact that it is difficult to find real tiredness signs after only half an hour, while test sessions could not last too much). Immediately after noticing such events, we explicitly asked users if they were actually reading something that they judged difficult to understand, and/or if they were getting tired. On average, we obtained the following confirmation percentages: 79% for C_{123} , 83% for C_4 and 81% for C_{56} .

4. Conclusions

We think that eye tracking technology can provide great benefits to e-learning, making it possible to create computer-based teaching systems able to “understand” the user and properly adapt content presentation. Currently, eye trackers are a (very expensive) niche product, used mostly in assistive and research fields, but things may change in a not too far future.

e5Learning, the system we have presented in this paper, is a real working e-learning environment where eye data are exploited to track user activities, behaviors and “affective” states. The experiments we have carried out have fully confirmed the viability of such an approach, in line with our expectations. However, much work can still be done, especially for the Emotion Recognizer component. To date, we have considered only two broad user states, namely “high workload or non understanding” and “tiredness”; more focused experiments will certainly allow to identify other learning conditions and better discriminate among them. Additionally, the integration with a (separate) facial expression recognition module could improve the whole process.

5. Acknowledgement

This work has been supported by funds from the Italian FIRB project “Software and Communication Platforms for High-Performance Collaborative Grid” (grant RBIN043TKY).

6. References

- [1] N. Hedge, and D. MacDonald, “Affective Considerations in Distance e-Learning”. *Proc. of the IASTED Conference on Education and Technology*, 2005.
- [2] R. W. Picard, and J. Klein, “Computers that Recognize and Respond to User Emotions: Theoretical and Practical Implications”. *MIT Media Lab Tech Report 538*, 2002.
- [3] M. Porta, “E-learning and machine perception: in pursuit of human-like interaction in computer-based teaching systems. *Int. Journal of Knowledge and Learning*, Vol. 3., Nos. 2/3, 2007, 281-298.
- [4] V. Cantoni, M. Cellario, and M. Porta, “Perspectives and Challenges in E-Learning: Towards Natural Interaction Paradigms”. *Journal of Visual Languages and Computing*, 15, 2004, 333-345.
- [5] M. Pivec, C. Trummer, and J. Pripfl, “Eye-Tracking Adaptable e-Learning and Content Authoring Support”. *Informatica*, 30 (2006), 83–86.
- [6] H. Wang, M. Chignell, and M. Ishizuka, “Empathic Tutoring Software Agents Using Real-time Eye Tracking”. *Proc. of ETRA 2006*, 73-78.
- [7] C. Merten, and C. Conati, “Eye-Tracking to Model and Adapt to User Meta-cognition in Intelligent Learning Environments”. *Proc. of IUI 2006*, 39-46.
- [8] A. Hyskykari, P. Majoranta, A. Aaltonen, and K. R ih a, “Design Issues of iDict: A Gaze-Assisted Translation Aid”. *Proc. of ETRA 2000*, 9-14.
- [9] T. Partala, M. Jokiniemi, and V. Surakka, “Pupillary Responses to Emotionally Provocative Stimuli”. *Proc. of ETRA 2000*.
- [10] M. Nakayama, and Y. Shimizu, “Evaluation of a Multimedia Learning Exercise using Oculo-motors”. *Proc. of ETRA 2006*, 46.
- [11] A. Murata, and H. Iwase, “Evaluation of Mental Workload by Fluctuation Analysis of Pupil Area”. *Proc. of the 20th Int. Conf. of the IEEE Eng. in Medicine and Biology*, 1998.
- [12] M. Nakayama, K. Takahashi, and Y. Shimizu, “The Act of Task Difficulty and Eye-movement Frequency for the ‘Oculo-motor’ indices”. In *Proc. of ETRA 2002*, 37-42.