Halbrügge, M. (2015). *Automatic Online Analysis of Eye-Tracking Data for Dynamic HTML-based User Interfaces*. In Gramann, K.; Zander, T. O. & Wienrich, C. (eds.). 11. Berliner Werkstatt Mensch-Maschine-Systeme, pp322-324, Universitätsverlag der Technischen Universität Berlin

Automatic Online Analysis of Eye-Tracking Data for Dynamic HTML-based User Interfaces

Marc Halbrügge[1]

Keywords: Eye-Tracking, HTML

Abstract

Knowing how humans distribute their visual attention while interacting with user interfaces is very important for interface designers and usability analysts. Eye-tracking can provide answers to this question. But while eye-tracking as a method is not new anymore, it is still hard to apply to real-word dynamic scenarios, the main hindrance being the complexity of the analysis. We present a tool that eases this situation for dynamic HTML applications by providing a mapping of gaze coordinates to HTML elements.

Introduction and Related Work

During the last decade, eye-tracking has become a commonplace method. Eye-tracking devices are now inexpensive enough to be introduced into the consumer market (e.g., Steelseries Sentry eye-tracker for computer gaming, Pupil Labs open source eye-tracker). While today's eye-tracking hardware is relatively easy to use, the analysis of the data created by it remains complicated. One major problem is the definition of areas of interest (AOI) in the visual scene that are a necessary prerequisite of most analyses. The AOIs have to be created by the researcher or analyst. Even for a static visual scene, this can be a time consuming task. Things get more complicated for scenes that change over time, as these need dynamic AOIs that appear, move, or disappear as the visual scene evolves.

When applying eye-tracking as part of usability evaluations, it is desirable that the participants can interact with the user interface (UI) under analysis as freely and naturally as possible. This leads to unique interaction paths for each participant and in consequence to a unique sequence of dynamic AOIs per participant. How can these be managed?

If the setting is under complete control of the researcher, dynamic AOIs can be created using special tools; most other cases require post-experimental annotation of the observed material on a frame-by-frame basis for every single participant (Jacob & Karn, 2003). Existing tools for dynamic AOIs use game engines (Nacke et al., 2011), movies with fixed timing (Papenmeier & Huff, 2010), or computer vision (De Beugher et al., 2014). Common to all of these tools is that they depend on the timing and interaction flow being controlled by the experimenter.

Because of this, these approaches are not applicable when we want to apply eye-tracking during user interaction with dynamic HTML applications. We are especially interested in this type of UIs as they become more and more popular with the raise of mobile devices. Eye-tracking has been applied while users were surfing the web before (e.g., WebEyeMapper, Reeder, Pirolli & Card, 2001; Buscher, Cutrell & Morris, 2009), but these examples are either bound to specific web browser applications or eye-tracking hardware and both do not supply the necessary software tools. Instead, we are proposing a new technical solution that maps the recorded user gaze to an element of the UI while the interaction is happening.

[1] Quality and Usability Lab, Telekom Innovation Laboratories, Technische Universität Berlin
Ernst-Reuter-Platz 7, 10587 Berlin, marc.halbruegge@telekom.de

Halbrügge, M. (2015). *Automatic Online Analysis of Eye-Tracking Data for Dynamic HTML-based User Interfaces*. In Gramann, K.; Zander, T. O. & Wienrich, C. (eds.). 11. Berliner Werkstatt Mensch-Maschine-Systeme, pp322-324, Universitätsverlag der Technischen Universität Berlin

Eye-Tracking of Dynamic HTML-based User Interfaces

The tool presented here is based on ACT-CV, a toolbox that provides the "glue" between graphical user interfaces and a cognitive architecture in the cognitive modeling domain (Halbrügge 2013, 2015). ACT-CV uses Mozilla's XUL framework for the display of HTML5 content. XUL allows to interact with unchanged web content by the means of an additional ECMAScript layer. ACT-CV then translates the graphical objects into a symbolic representation that is suitable for the Lisp-based cognitive architecture ACT-R.

How can we leverage these capabilities for eye-tracking analysis? As ACT-CV already holds a representation of the visual scene, this representation can be used to map gaze positions to the HTML elements they point to. Due to the hierarchical nature of HTML, this mapping is often not unique. A screen position inside a button on the screen could at the same time be inside a form element which is the button's parent. Ambiguities like this one are resolved by the tool by a) preferring elements that support interaction (e.g., a button instead of a table element), and b) preferring elements that are lower in the hierarchy (e.g., a table cell instead of the complete table).

Following this path, we implemented a direct connection between a web browser and an SR Research Ltd EyeLink II eye-tracker. During the users' interaction with a dynamic web application, the position of the pupils, the corresponding gaze coordinates on the computer screen, and the graphical element underneath are recorded. Using this approach, many important indicators, e.g., time on target, can be examined right away. Fixation detection algorithms can be applied as well, but should be chosen carefully (Salvucci & Goldberg, 2000).

Conclusions

Eye-Tracking is used in many scientific and applied domains. Due to the complexity and sheer amount of data created by this method, the analysis is often limited to gaze heat map overlays on static visual scenes like photographs or screenshots. The potential of eye-tracking for interactive content has not yet been fully exploited, mainly because of technical difficulties. This paper presents a tool that allows easy eye-tracking analyses of dynamic web content by mapping the calculated gaze position to its corresponding HTML element.

Acknowledgment. The author gratefully acknowledges financial support from the German Research Foundation (DFG) for the project "Automatische Usability-Evaluierung modellbasierter Interaktionssysteme für Ambient Assisted Living" (MO 1038/18-1).

Literature

Buscher, G., Cutrell, E., & Morris, M. R. (2009). What do you see when you're surfing? Using eye tracking to predict salient regions of web pages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 21-30

De Beugher, S.; Brône, G. & Goedemé, T. (2014). Automatic analysis of in-the-wild mobile eye-tracking experiments using object, face and person detection. In *Proceedings of the international conference on computer vision theory and applications (VISIGRAPP 2014)*, 625-633

Halbrügge, M. (2013). ACT-CV: Bridging the gap between cognitive models and the outer world. In E. Brandenburg, L. Doria, A. Gross, T. Günzlera, & H. Smieszek (Eds.), *Grundlagen und Anwendungen der Mensch-Maschine-Interaktion*, 205–210. Berlin: Universitätsverlag der TU Berlin.

Halbrügge, M. (2015). Fast-Time User Simulation for Dynamic HTML-based Interfaces. In Taatgen, N. A.; van Vugt, M. K.; Borst, J. P. & Mehlhorn, K. (Eds.) *Proceedings of the*

13th International Conference on Cognitive Modeling, 51-52. Groningen: University of Groningen

Jacob, R. J., & Karn, K. S. (2003). Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In Hyona, Radach & Deubel (eds.) *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*, 573-605. Oxford: Elsevier.

Nacke, L. E.; Stellmach, S.; Sasse, D.; Niesenhaus, J. & Dachselt, R. (2011). LAIF: A logging and interaction framework for gaze-based interfaces in virtual entertainment environments. *Entertainment Computing*, *2*, 265-273

Papenmeier, F. & Huff, M. (2010). DynAOI: A tool for matching eye-movement data with dynamic areas of interest in animations and movies. *Behavior research methods*, *42*, 179-187

Reeder, R. W., Pirolli, P., & Card, S. K. (2001). WebEyeMapper and WebLogger: Tools for analyzing eye tracking data collected in web-use studies. In *CHI'01 extended abstracts on Human factors in computing systems*, 19-20

Salvucci, D. D., & Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications*, 71-78. ACM.