

Visualization of Gaze Tracking Data for UX Testing on the Web

Róbert Móra
Slovak University of
Technology in Bratislava
Faculty of Informatics and
Information Technologies
Ilkovičova 2, 842 16
Bratislava, Slovakia
robert.moro@stuba.sk

Jakub Daráž
Slovak University of
Technology in Bratislava
Faculty of Informatics and
Information Technologies
Ilkovičova 2, 842 16
Bratislava, Slovakia
xdarazj@stuba.sk

Mária Bieliková
Slovak University of
Technology in Bratislava
Faculty of Informatics and
Information Technologies
Ilkovičova 2, 842 16
Bratislava, Slovakia
maria.bielikova@stuba.sk

ABSTRACT

Visualizations on the Web can help users to understand complex concepts, such as when too many objects of possible interest are present. For the purpose of evaluation of their usability, gaze tracking data represent a valuable source of information. These data are themselves complex, time-varying and in large quantities, thus posing challenges on their manipulation and visualization. We propose an infrastructure for collection and visualization of the gaze tracking data from dynamic Web applications. Its main purpose is to support researchers in UX (user experience) testing of their proposed interfaces (and visualizations). In the paper, we provide a user study on the usability of the infrastructure and compare it to existing solutions.

Categories and Subject Descriptors

H.1.2 [Models and Principles]: User/Machine Systems—*human factors*; H.5.2 [Information Interfaces and Presentation]: User Interfaces—*evaluation/methodology, user-centered design*

General Terms

Design, Experimentation, Human Factors

Keywords

gaze tracking, infrastructure, visualization, UX testing, areas of interest, web

1. INTRODUCTION

For a picture (a visualization) to be worth a thousand words, it has to have a clear message that is easily understandable by the users (receivers of the message). However, visualizations nowadays are usually not only static pictures, but require often complex interaction with the interface elements, such as filtering values, selecting ranges (e.g. time, price, etc.), zooming or navigation. In addition, this interaction is in many cases carried out in the Web environment with dynamically generated, or even streamed content. Evaluating proposed visualization, its usability and the overall user experience (UX) can be, therefore, an uneasy task.

There are many questions that can be of importance during evaluation, such as: How much time do the users spend looking at the visualization and how much time interacting

with the interface? In what order do they receive the information? Do they read the accompanying text? Does the pattern change when we change a particular element (its position, design, etc.)? In order to answer these questions, it is not enough to rely on the indirect or implicit forms of feedback, such as position of a mouse cursor, clicks or scrolling. We need to evaluate what the users are actually looking at.

For this purpose, we can utilize gaze tracking technology that is becoming more affordable for the researchers and the ordinary users alike. Existing solutions have, however, often only limited support for the Web-based dynamic applications. In this paper, we propose an infrastructure for gaze tracking data collection and visualization focusing on the Web environment. We developed a prototype that can transparently work with gaze tracking devices from various manufacturers and supports multiple browsers. We provide an empirical evaluation of the proposed infrastructure and its visualization capabilities for UX testing and compare it to some of the existing solutions.

2. RELATED WORK

Eye tracking has been applied in many user studies in the recent years. With lowering price and increasing availability of low-end models, it is becoming possible to have eye-trackers not only in UX laboratories, but also in end-users' notebooks. It opens up new possibilities for types of interactions and adaptation, i.e. personalization of the applications to the users as noted in [1]. The authors verified that adapting the displayed ads on a website based on gaze data resulted in significant increase of users' attention.

Adaptation of visualization based on gaze data was proposed in [5]. The authors compared two types of visualization, namely bar chart and radar graph on fourteen tasks of differing type and complexity. In addition, the participants' personal traits (cognitive abilities), such as perceptual speed or visual working memory have been tested. They were able to correctly classify the task's type, complexity and the users' cognitive ability based on the gaze data and selected areas of interest, thus showing, that there are distinct differences in patterns and interaction styles worth of adapting to the users.

Individual differences in gaze patterns and behaviours were

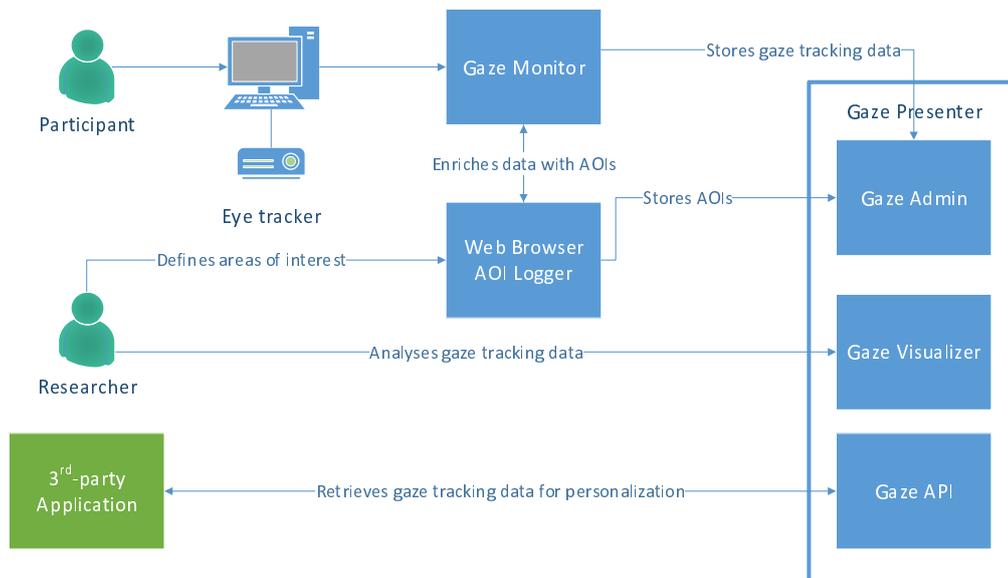


Figure 1: Conceptual design of the proposed architecture.

observed in [2] as well. Eye tracking has also been utilized in a user study on visualization of faceted interface [3]. The authors were interested in finding out, whether the users do not use facets just because they are shown to them. Therefore, they automatically hid (collapsed) them. Using the eye-tracker they verified that the faceted interface was used heavily in both cases (when visible as well as when hidden) with no significant difference in gaze patterns.

In order to be able to effectively evaluate areas of interest, we need to be able to track them throughout dynamically changing content. An algorithm for this purpose was proposed in [4] focusing on the tracking in video content.

However, tracking areas of interests on the Web usually requires different approach as the content can change completely, although it is still the same element (area of interest). According to our knowledge, it is still largely unsupported by the existing eye-tracking software.

The *Eye Tribe*¹ that promises a cheap tracker comes with no software, only with API for developers. On the other hand, *Tobii Technologies*² offers a *Tobii Studio* that comes with a full support for planning user studies, tracking, visualization and evaluation. However, it works only with Internet Explorer and areas of interest can be added only as static rectangles or polygons which is unusable with dynamically changing Web content. The best support for Web 2.0 seems to have *Nyan 2.0*³ solution by *Eye Gaze, LC Technologies*. It can recognize different overlays and also visualize Web navigation paths. Areas of interest are, however, still defined as polygons. In addition, most of the existing solutions try to roll-out the Web pages to account for scrolling. This is, however, not enough for many modern applications, which can have different elements with their own scrollbars (e.g.

Facebook with its chat, activity stream etc.).

Other problem with existing solutions is support for only one tracking device, i.e. multiple users cannot be tracked at the same time with exception of *Eyeworks* software by *Eyetracking*⁴ when combined with their *Quad* server solution. Even so, the existing solutions for gaze data collection and visualization are developed by the eye-trackers' manufacturers and therefore, they are closed to one particular eye tracker brand and cannot be extended to work with devices from other manufacturers.

3. INFRASTRUCTURE FOR GAZE TRACKING

In order to address the problems of existing solutions discussed in the previous section, we propose an infrastructure for gaze tracking focusing on the dynamic Web applications. Its conceptual design can be seen in Figure 1. It consists of three main components, namely *Gaze Monitor*, *Web Browser AOI Logger* and *Gaze Presenter*, which in turn comprises of *Gaze Admin*, *Gaze Visualizer* and *Gaze API*.

Researchers define the areas of interest (AOI) using the *Web Browser AOI Logger* which are then stored on the server. They can set-up the whole experiment using the *Gaze Admin*, which is a part of *Gaze Presenter* component.

Then, the participants can connect using the *Gaze Monitor*, which communicates in the background with the eye-tracker, collects the gaze tracking data and sends them to *Web Browser AOI Logger* for enrichment. The data are enriched with the XPath⁵ of the element the user (i.e. participant) is looking at, based on the coordinates supplied by the eye-tracker. The URL of the current website is added as well. Enriched data are sent by the *Gaze Monitor* at

¹<https://theeyetribe.com/>

²<http://www.tobii.com/>

³<http://www.eyegaze.com/eyegaze-analysis-software/>

⁴<http://www.eyetracking.com/>

⁵<http://www.w3.org/TR/xpath/>

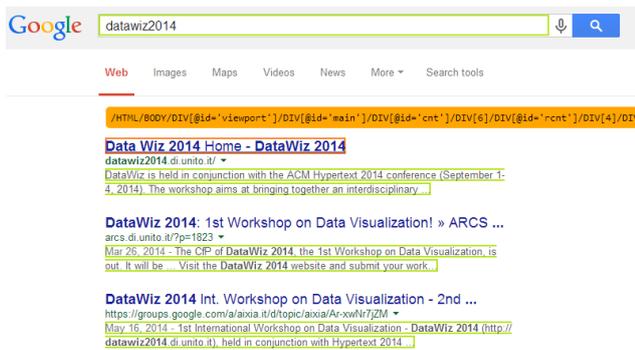


Figure 2: HTML elements highlighted during definition of areas of interest. Green ones have already been added (note that every snippet is a part of an area of interest definition), orange is highlighted upon mouse hover and can be selected by a mouse click.

specified time intervals to the *Gaze Presenter* for persistent storage.

They can be viewed and analysed by the researchers using the *Gaze Visualizer* component. The data can be also retrieved using the provided *Gaze API* and then manipulated by the third-party applications.

The individual components are further described in the following sections.

3.1 Gaze Monitor

Gaze Monitor connects to an eye-tracking device to receive gaze data from it. In order to transparently support devices from various manufacturers, we have implemented our own library that serves as a façade to the actual eye-tracker's API. Currently, we support devices from two manufacturers, namely *Tobii Technologies* and *The Eye Tribe*. In addition, we provide our own gaze data simulator that enables developers and researchers to develop applications for the eye-tracker without having one; gaze is simulated by the position and movement of the mouse cursor. Because it uses our provided library, the applications developed and tested with the help of the simulator can consume the simulated gaze tracking data as if they were from the actual eye-tracking device (i.e. using the same API calls).

The Gaze Monitor stores gaze data from the tracker in a queue. It communicates with the our provided browser extension - Web Browser AOI Logger, sends it the queued data and receives the enriched data. These are sent to the server in specified time intervals.

3.2 Web Browser AOI Logger

Web Browser AOI (Area of Interest) Logger is realized as an extension to the web browser. Its main functionality is to enrich data from the Gaze Monitor. Currently, we support both Google Chrome as well as Mozilla Firefox browser. The gaze tracker data contain normalized coordinates which are recalculated in order to identify the specific HTML element of the displayed Web page. The element is identified by its



Figure 3: XPath string. It can be customized by deselection of the specific path's elements (in gray).

unique XPath.

The extension is also used to define areas of interest on the Web page, which is in more detail described in section 4.1.

3.3 Gaze Presenter

The data sent from the Gaze Monitor are collected by the provided server application, i.e. the Gaze Presenter. It enables data collection from multiple connected users at once. We use two databases for storing the data; SQL database for storing the information about experiments (projects, sessions, users, areas of interest) and NoSQL document-based database *RavenDB* for storing the enriched gaze tracking data in JSON format. One of the considerations when choosing the data storage was velocity of the incoming data; the eye-tracker's frequency is (based on the actual model) at least 30Hz meaning that we have approximately 100,000 new data records per each hour's worth of tracking.

The collected data can be accessed and visualized by the users using the provided Web interface. In addition, we provide an API for third-party applications that can consume collected data (i.e. what users are looking at which elements at what time) and e.g. adapt (personalize) the visualized information based on the users' gaze, i.e. what they are (not) looking at. Thus, the gaze tracking can be used not only for the purpose of evaluating the interface (visualization), but can be considered as a form of implicit user feedback. This way, it can help to model interests of the users more precisely.

4. VISUALIZATION OF GAZE TRACKING DATA

Visualization of gaze tracking data is crucial for its understanding and usage for evaluation of the user interfaces. Complexity lies in the data's velocity, multidimensionality and time variability. We can significantly reduce the computational requirements, when we include only data for specific areas of the tested Web page that are of an interest for us (so-called areas of interest). Thus, instead of computing fixations for all the elements, we can do it for a handful defined by the user.

4.1 Definition of Areas of Interest

We enable users to define areas of interest (AOI) using our browser extension. After activation, the elements in the Web page are highlighted upon mouse hover (see Figure 2). After the highlighted element is clicked on, the pop-up appears, in which it is possible to customize the selected area (name it, described it) or to change the XPath (see Figure 3) to suite the user's specific needs.

It is, thus, possible to choose not only the actual clicked element, but e.g. every paragraph with the same parent, or

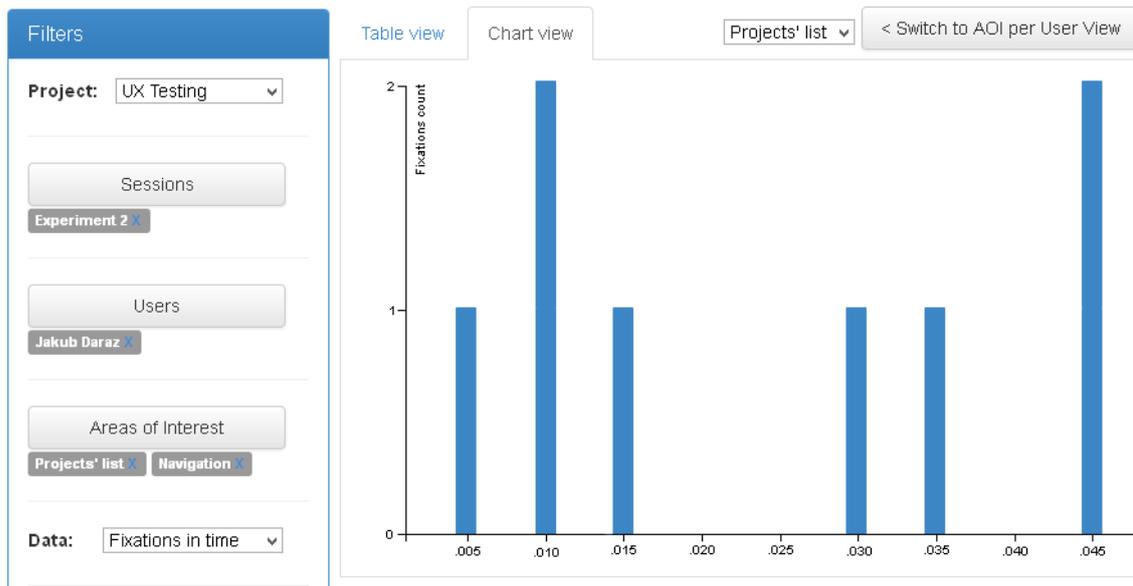


Figure 4: Visualization of fixations in time for the selected user and area of interest.

every element with the same class, etc. This can be used with advantage for the dynamically generated Web pages, where we do not know exact element’s path, but we can identify it by its relative position within the HTML DOM structure or by its other attributes. It also enables users to include to an area of interest elements which are generated on the fly and are therefore not present at the time of area of interest definition, but share the same attribute value.

4.2 Visualization of Metrics

The eye-tracker tracks the position and movements of each eye separately; however, we are interested, what a user is looking at (which is rarely two things at once). Therefore, we calculate the gaze position as an average of the two eyes’ coordinates. In addition, the tracker is not always precise and the gaze can seem to oscillate around a specific point, when the user actually looks at the same point the whole time. We use several smoothing techniques to account for this, especially a *moving average technique* by averaging N consecutive gaze coordinates from a moving window. The users (researchers) can also specify minimal time threshold for fixation, i.e. for how much time (e.g. 500 ms, 1 s, etc.) the user has to look at the area for it to count as a fixation. This way, we can filter out events, when the user moved gaze through the element without actually fixating on it.

The cleared data can be accessed and visualized by the users using the provided *Gaze Visualizer* component. Currently, we support the following metrics:

- *Number of fixations* - it counts, how many times the users looked at the specified areas of interest during the duration of the whole session
- *Dwell time* - similar to the first metric, but instead of the number of times the users’ gaze entered the areas of interest, it aggregates the spent time (how long the

users looked at the areas of interest during the whole session)

- *Fixation in time* - it shows, how the fixation count changed over the time of the experimental session (see Figure 4)

The users can aggregate and compare the data from multiple sessions, users and for multiple areas of interest using the provided filtering options. Data are shown in tabular view as well as visualized in the form of charts using the *D3.js*⁶ library. The charts can be exported and saved to disk.

5. EXPERIENCE WITH THE PROPOSED GAZE TRACKING INFRASTRUCTURE AND ITS USABILITY

In order to evaluate our proposed infrastructure, we carried out a user study with four participants. We chose participants who had previous experience with eye tracking in Tobii Studio, so that they could compare the functionality of the both systems.

The participants’ task was to set-up an experiment using our infrastructure, then collect the gaze data and lastly, to visualize and evaluate it. At the end, we asked them to fill in a questionnaire evaluating the different features.

The participants rated highly the provided functionality of defining the areas of interest. It was also rated as intuitive and easy to understand (4.25 on average from a five-point Likert scale). However, we observed problems with editing the XPath string, namely the participants did not intuitively find out that it is customizable. After explanation of how it works, they appreciated the flexibility. One of the participant suggested that he would be interested to define not

⁶<http://d3js.org/>

only a single area of interest as a combination of different elements (e.g. each result on search engine's results page), but also to explore the differences in gaze patterns with individual elements within this area of interest group.

The participants found the experiment easy to set-up, although they had in some cases problems to understand the difference between a project and its sessions. As to the visualization, it was again rated very positively (4.25 on average), even though we currently provide only visualization of the three metrics. On the other hand, these metrics are ones of the most often used as we also verified in the reviewed literature (they were used practically in all of the related works reported in this paper). The participants missed the most possibility of creating heat maps and fixation sequences (how the gaze moves from element to element).

Compared to Tobii Studio, the participants appreciated the flexibility of defining the areas of interest, support of multiple browsers and multiple concurrent users as well as possibility to manually set the preferred minimal length (threshold) of fixations. On the other hand, they lacked audio and video recording and support of data inputs other than gaze, such as mouse clicks (left and right button), scroll events, etc. They would also appreciate the possibility to export the data or to clean it within our application.

Lastly, two participants would use our solution alone and two in combination with others, such as Tobii Studio, mainly for the lack of audiovisual recording. Overall, we find the feedback positive and encouraging for future development.

6. CONCLUSIONS

In the paper, we proposed an infrastructure for collection and visualization of gaze data focusing on the dynamic Web applications. Our main contributions are:

- visual definition and support of dynamic areas of interest, the content of which as well as size and position can change over time
- support of multiple browsers and eye-trackers from different manufacturers by providing a unified and easily extensible API
- collection and automatic evaluation of the gaze data from multiple concurrent devices and users

We realized a prototype of the infrastructure and carried out an user study in order to gain feedback to its functionality and usability. Based on the collected user feedback described in previous section, we plan to provide heat maps as well as fixation sequences visualization in the future. More importantly, we would like to enhance the data manipulation techniques, such as cleaning the data, selecting time ranges, zooming in and out, etc.

Currently, it is possible to automatically annotate the gaze data based on the fixations within the areas of interest defined by the users. However, the users may wish to add other annotations of different types either manually or automatically based on a set of predefined rules. It can be in

a form of events, e.g. someone entered the room during the study, the participant looked away, the user study moderator provided a guidance, etc. These events represent useful metadata that can further explain the collected gaze data and provide new insights. In addition, it would be interesting to segment the data based on these events or compare the changes in gaze patterns or behaviour (e.g. before giving guidance and after it).

In order to support this kind of annotations, we have to solve several (also) visualization issues, namely visualization of gaze data stream in real-time and adding the annotations to a single point in data or a range. The easy to understand and intuitive visualization of the associated annotations in the data in the process of evaluation is also an open problem.

In addition, it is very likely that the eye-trackers will be a part of end-user devices in the near future. This will allow usage of gaze data as one of the implicit feedback factors of users' interest. When we combine our provided *Gaze API* with the events in the form of annotations, it can support new ways of personalized interactions on the Web.

7. ACKNOWLEDGMENTS

This work was partially supported by grants No. APVV 0208-10 and VG1/0971/11 and it was created with the support of the Research and Development Operational Programme for the project "University Science Park of STU Bratislava", ITMS 26240220084, co-funded by the European Regional Development Fund.

We would like to thank our colleagues who participated on the development of the presented prototype, namely Dominika Červeňová, Lukáš Gregorovič, Michal Mészáros, Róbert Kocian, Martin Janík and Kristína Mišíková. We thank also the *Tobii Technology* for kindly providing us with the eye tracker as well as Tobii Studio for evaluation purposes.

8. REFERENCES

- [1] F. Alt, A. S. Shirazi, A. Schmidt, and J. Mennenöh. Increasing the user's attention on the web. In *Proc. of the 7th Nordic Conf. on Human-Computer Interaction Making Sense Through Design - NordiCHI '12*, pp. 544–553, NY, USA, 2012. ACM Press.
- [2] S. T. Dumais, G. Buscher, and E. Cutrell. Individual differences in gaze patterns for web search. In *Proc. of the 3rd Symposium on Information Interaction in Context - IIX '10*, pp. 185–194, NY, USA, 2010. ACM Press.
- [3] M. Kemman, M. Kleppe, and J. Maarseveen. Eye tracking the use of a collapsible facets panel in a search interface. In *Proc. of the 17th Int. Conf. on Theory and Practice of Digital Libraries - TPD L '13*, pp. 405–408, Berlin, Heidelberg, 2013. Springer.
- [4] F. Papenmeier and M. Huff. DynAOI: a tool for matching eye-movement data with dynamic areas of interest in animations and movies. *Behavior Research Methods*, 42(1):179–87, Mar. 2010.
- [5] B. Steichen, G. Carenini, and C. Conati. User-adaptive information visualization. In *Proc. of the 2013 Int. Conf. on Intelligent User Interfaces - IUI '13*, pp. 317–328, NY, USA, 2013. ACM Press.