

## Mousetracking Visitors to Evaluate Efficacy of Web Site Design

Aleksandar Jevremović<sup>1</sup>, Saša Adamović<sup>1</sup>, Mladen Veinović<sup>1</sup>

**Abstract:** Objective assessment of website design is a complex task whose results are valuable for improving efficiency. Some of the commonly used methods, such as A/B testing, are relatively simple to implement but generally give only a rough summary of results. Eyetracking is one of the most efficient methods, but requires the use of expensive equipment and is not generally applicable to the web environment. Mousetracking is a far cheaper and implementable alternative to eyetracking and some studies indicate a high degree of correlation between the results of these two methods. This paper presents the architecture of a system for mousetracking, and the results of its use in a particular website.

**Keywords:** Mousetracking, Eyetracking, Design Evaluation.

### 1 Introduction

Objective evaluation of website design is a complex task whose results are valuable for improving its efficiency. Design in this case is not just graphic design, but the organization of content in combination with the components that make up the user interface.

Some of the commonly used methods, such as A/B testing, are relatively easy to implement but generally give only a rough summary of results. A significant failure of A/B testing, especially when it comes to sites with relatively low traffic, is the fact that it takes a long time to obtain statistically verified conclusions. For example, on a site that is visited daily by about 200 visitors, an efficiency of 20%, confirmation of an increase in efficiency of 10% would require 64 days of testing. This calculation refers to the testing of only two variants, the site or any part thereof. In the case of A/B using the ten tested embodiment, the time needed to produce statistically significant results is 320 days. In most real situations specified times are not acceptable because the results could be obsolete by the time they are produced.

Eye (pupils) tracking of users is one of the most effective methods, but requires the use of expensive equipment and is not generally applicable to the

---

<sup>1</sup>Faculty of Informatics and Computing, Singidunum University, Danijelova 32, 11000 Belgrade, Serbia;  
E-mail: ajevremovic@singidunum.ac.rs, sadamovic@singidunum.ac.rs, mveinovic@singidunum.ac.rs

web environment. Mousetracking is a far cheaper and more practicable alternative for eyetracking and some studies indicate a high degree of correlation between the results of these two methods.

This paper presents the architecture of a system for tracking the mouse cursor of visitors, as well as the results of its use in a particular website.

## 2 Eye tracking and Mousetracking

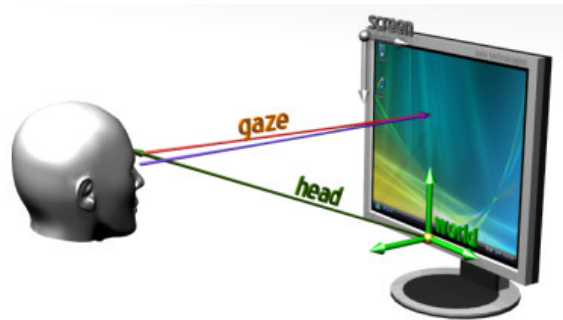
A number of parties, such as software companies, website owners, and the like, are keen to acquire information about how people use computer systems (hardware and software) and how their use can be optimized. This area is studied in engineering psychology, which examines the role of human factors [1].



**Fig. 1** – Example of eyetracking by using mobile openEyes open-source and open-hardware solution.

Eyetracking of users is a method that is widely used in research centers of large software companies to analyze the performance of users in dealing with their products [5, 7, 8]. This tracking is performed with special equipment (the price of which often amounts to several tens of thousands of dollars, but there are much cheaper alternatives [2]) that captures pupils of users and determines what they are looking at on the screen.

Eyetracking is often combined with “thinking out loud” in which the user speaks his/her mind in the process of solving a specific task (for example: “I’m looking for an option to import data,” “trying to save the current state,” etc.). The user’s view is located and his/her stream of thought matched and analyzed with the aim of finding errors in the design of the user interface or parts thereof which do not coincide with the intuition of the user.



**Fig. 2** – *A typical mode monitors tracking look at examples of IG-30 commercial solutions company Alea technologies.*

It is clear that such tracking is a complex method which gives very significant results but may not be widely applicable. In addition, because this method of testing is conducted under controlled conditions - usually in the room of the test organizers where the specific equipment is located - its application to the web in real conditions is practically impossible.

Basic alternatives to eyetracking include tracking the user's mouse cursor position. Some studies show that there is a very high degree of correlation - over 88% - between positions corresponding to the view and the cursor [3]. On the other hand, tracking the mouse cursor can be implemented in a real web environment with a larger sample of visitors.

### **3 Description of Developed System**

Mousetracking in the user base requires an active component on the computer to record the coordinates of the cursor in the form of time series data. Additional components of the task and when it comes to Web environment, and to send recorded data server. Components of this type that are used in traditional desktop environments produce results mainly recorded in the local log files.

The implementation of the aforementioned components on the client side in a web environment can make use of technologies such as JavaScript, Java applets or Adobe Flash. The system developed by the authors is based on JavaScript technology, which is benign and also supported on different platforms.

Implementation of server components was achieved by using the PHP programming language on an Apache HTTP server. MariaDB DMBS was used for storage of the data obtained. What should note is that the server side can represent a bottleneck in situations where there are a large number of active parallel visitors to a website to which the monitors, and accordingly to

determine the characteristics of the server environment (hardware components, the capacity of the channel, operating system, etc.).

All communication between the client and the server, when it comes to data related to the mouse cursor position and other parameters of the state of the environment, was performed within the To overcome the numerous problems encountered in practice, however, it proved necessary to develop a special sub-protocol (still within the HTTP protocol) with functionalities close to the TCP (transmission control protocol) protocol: segmentation data, retransmission of lost segments, correct order, etc.. In developing this protocol we took advantage of previous experiences of the implementation of different algorithms in HTTP message protocol [4].

### 3.1 Display structure

The mousetracking system we developed for our research is able to detect five levels of objects that are involved in presenting web content: screen, window, viewport, page and wrapper. The screen represents a physical display, a matrix of pixels with dynamic width and height.

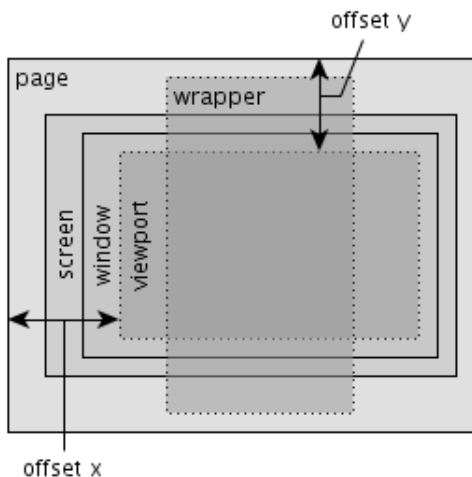


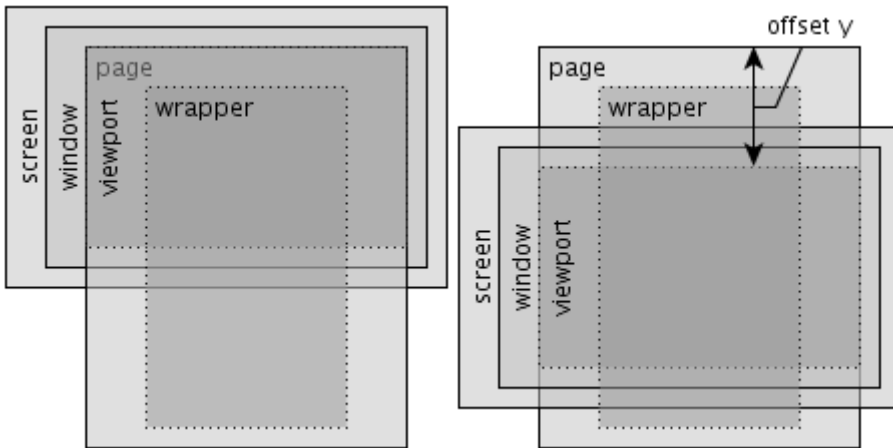
Fig. 3 – Display structure.

The window represents the web browser's GUI window object, with variable width, height and position on screen. All window pixels are usually displayed within the screen area and the system detects that most visitors use a web browser within a maximized window. Also, the system is able to detect if a window is active (in focus) or not (minimized, behind some other window...) which is also important for our research.

The viewport is a “useful” part of the web browser's window or where the loaded webpage is actually displayed. The size of this area is equal to that of the

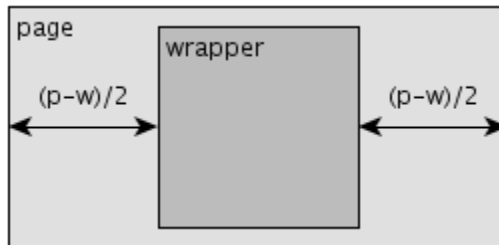
window minus the window title and border, toolbars and status bar. Scrollbars, if any, are included within the viewport area. When the web browser is in full-screen mode, the viewport size is equal to window size and screen size.

The page area is a representation of the body element within an HTML document. It can fit within the viewport area if the page's content does not require more space to be displayed. If the content's display size is less than the available viewport size, the page area expands to fit the viewport size. Conversely, if the display content requires more space than the viewport area, page size is adapted to content size and viewport scrollbar(s) are enabled.



**Fig. 4** – Page with no horizontal or variable vertical offset.

Offsets are a measure of how many pixels a page scrolls vertically or horizontally. Even if we are able to calculate mouse position relative to the page/wrapper directly, putting it in a viewport/offset context can provide additional useful data, because users interact differently with different viewport parts.

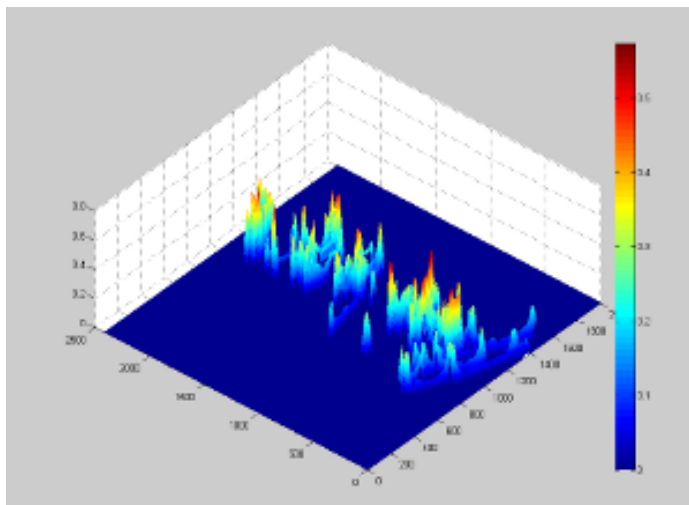


**Fig. 5** – Wrapper position with “auto” left and right margins.

The wrapper is the page's content part that our tracking is focused on. It can be any displayed part of the content, but usually it is a DOM element within the body element which contains all the page content. In our case, we used one div element with "TPL\_Wrapper" value of "ID" attribute which was the only child of body element and which contained all the page content within itself. The wrapper in our experiment had 878 pixels and both left and right margins were set to "auto," which resulted in a horizontally centered wrapper area. We consider this (horizontally centered content) as generally good practice, and it is especially useful in eye/mouse tracking experiments because content is displayed directly in front of the visitor.

## 4 Experimental Results

The role of the described system is to provide data for further analysis. The analysis includes cleaning and normalization of data, the application of appropriate filters, visualization and pairing with picture web pages and drawing conclusions.

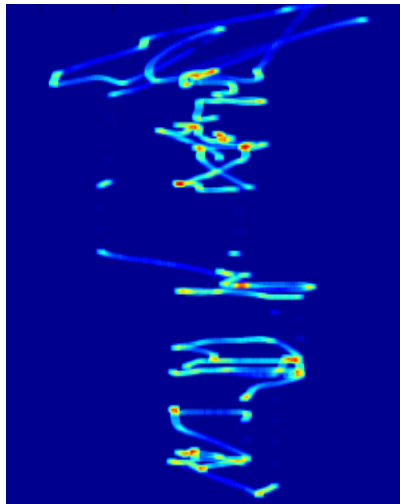


**Fig. 6** – *Three-dimensional heat map display allows easy retrieval of parts of the site where the mouse cursor is no longer kept.*

The cleaning and normalization of data are the most difficult part of the process because it is necessary to apply algorithms that will find and remove or correct data that are not of the view that the user saw, but are the result of random and non-core activities. For example, it is necessary to identify situations in which the user letting go of the computer web browser window with the active cursor to a position. On the other hand, this situation should not be confused with that where the user spends some time reading some text on the

page, while at the same time removing the mouse cursor to the side of the content. Because of the different representations of the same content in different web browsers prior to the generation of heat maps it is necessary to execute the appropriate mapping and correction of certain items obtained from the web browser where there are deviations from the chosen reference display.

The described introductory phase forms the matrix of relativized intensities, and time spent by the user in viewing different parts of rendered web pages. The most appropriate form of visualization of this matrix proved to be heat maps [6], both two-dimensional and three-dimensional.



**Fig. 7** – *Two-dimensional heat map obtained on the basis of the matrix position where the mouse cursor is kept.*

Although the three-dimensional heat map (Fig. 6) are easier to spot zone the highest intensity of observation, in practice, far more commonly used two-dimensional heat map (Fig. 7), because it is easier to compare with Web pages to which the map is made.

#### **4.1 Representative results**

In relatively well-organized and well-designed web sites most heat maps confirm the expected manner of their use. For example, Fig. 8. shows a typical user behavior in which more attention is paid to the content that is closer to the top of the page.

On the other hand, deviations from the standard and expected behavior patterns indicate the places that need improvement or that may be otherwise utilized. For example, the heat map in Fig. 9. shows that users spend most time watching the content located at the bottom of the page, especially the part that initially (immediately after loading) does not show the most popular screen.



Fig. 8 – In most sites the content is closer to the top, and visitors look at it for longer.

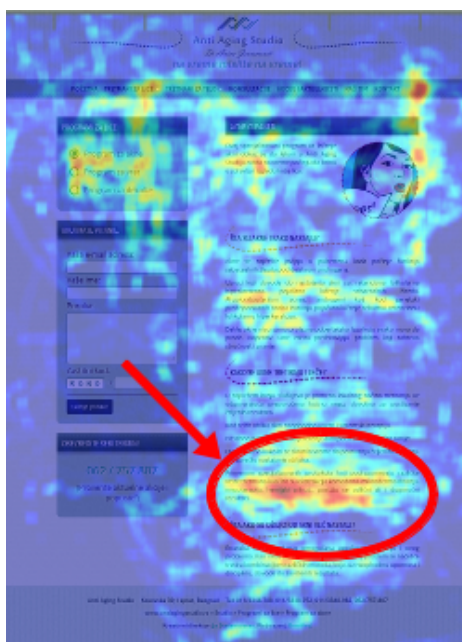


Fig. 9 – Poor organization of content on the page - the text on which visitors are given more time to work is below the page that is initially displayed after loading.



Another problem that was detected and resolved through the use of the described method is the main menu website that is designed to be next to each item thumbnail is active or passive bullets (Fig. 10).



**Fig. 10** – *This heat map suggests that visitors try to click on the bullets in the menu that initially were not part of the hyper-links.*

Given that the aforementioned images are not part of a hyper-link that should be clicked with the mouse, although the heat map showed that a large number of visitors do just that, appropriate changes were made to the area where the hyper-link extended. This change after the application resulted in an increase of the average number of open pages per visit and an increase in its average lifetime.

## 5 Conclusion

This paper describes the process of monitoring the mouse cursor of website visitors to evaluate the effectiveness of its design. The described method has been implemented using an original solution developed by the authors. The paper highlighted representative examples encountered while the authors applied the solution described in real projects.

Tracking mouse cursors is a less reliable method than eye tracking. On the other hand, mouse tracking can be implemented as software, which avoids the purchase of expensive devices. Additionally, tracking the mouse cursor can be implemented in a real web environment that is not possible when it comes to eye tracking that require controlled conditions.

One of the main advantages of the method described in relation to other popular methods for measuring the efficiency of design web sites (such as the A/B test) is the possibility of drawing conclusions on the basis of a relatively small number of visitors. On the other hand, tracking the mouse cursor in a web

site with a lot of traffic requires proper sampling or complex server infrastructure.

## **6 Acknowledgment**

This work was supported by the Serbian Ministry of Education, Science and Technological Development (projects TR32054, III 44006 and ON 174008).

## **7 References**

- [1] R.W. Proctor, K.P.L. Vu: Cumulative Knowledge and Progress in Human Factors, Annual Review of Psychology, Vol. 61, Jan. 2010, pp. 623 – 651.
- [2] D. Li, J. Babcock, D. J. Parkhurst: OpenEyes: A Low-cost Head-mounted Eye-tracking Solution, Symposium on Eyetracking Research and Applications, San Diego, CA, USA, 27 – 29 March 2006, pp. 95 – 100.
- [3] C.M. Chen, J.R. Anderson, M.H. Sohn: What Can a Mouse Cursor Tell us More?: Correlation of Eye/mouse Movements on Web Browsing, Human Factors in Computing Systems, Seattle, WA, USA, 31 March – 05 April 2001, pp. 281 – 282.
- [4] A. Jevremovic, R. Popovic, D. Zivkovic, M. Veinovic, G. Shimic: DMS - Improving Web Performance, 19<sup>th</sup> Telecommunications Forum Telfor 2011, Belgrade, Serbia, 22 – 24 Nov. 2011, pp. 166 – 169.
- [5] M. Nakayama, Y. Hayashi: Estimation of Viewer's Response for Contextual Understanding of Tasks using Features of Eye-movements, Symposium on Eye-Tracking Research and Applications, Austin, TX, USA, 22 – 24 March 2010, pp. 53 – 56.
- [6] F. Lettner, C. Holzmann: Heat Maps as a Usability Tool for Multi-touch Interaction in Mobile Applications, 11<sup>th</sup> International Conference on Mobile and Ubiquitous Multimedia, Ulm, Germany, 04 – 06 Dec. 2012, p. 49.
- [7] H.A. Maqbali, F. Scholer, J.A. Thom, M. Wu: Using Eye Tracking for Evaluating Web Search Interfaces, 18<sup>th</sup> Australasian Document Computing Symposium, Brisbane, Australia, 05 – 06 Dec. 2013, pp. 2 – 9.
- [8] J.H. Goldberg M.D. Stimson M. Lewenstein, N. Scott, A.M. Wichansky: Eye Tracking in Web Search Tasks: Design Implications, Symposium on Eye Tracking Research and Applications, New Orleans, LA, USA, 25 – 27 March 2002, pp. 51 – 58.