



HAL
open science

Remote Usability Testing Using Eyetracking

Piotr Chynal, Jerzy M. Szymański

► **To cite this version:**

Piotr Chynal, Jerzy M. Szymański. Remote Usability Testing Using Eyetracking. 13th International Conference on Human-Computer Interaction (INTERACT), Sep 2011, Lisbon, Portugal. pp.356-361, 10.1007/978-3-642-23774-4_29 . hal-01590555

HAL Id: hal-01590555

<https://inria.hal.science/hal-01590555v1>

Submitted on 19 Sep 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Remote Usability Testing Using Eyetracking

Piotr Chynał and Jerzy M. Szymański

Institute of Informatics, Wrocław University of Technology
Wyb. Wyspińskiego 27, 50-370 Wrocław, POLAND
Piotr.Chynal@pwr.wroc.pl, 157690@student.pwr.wroc.pl

Abstract. In the paper we present a low cost method of using eyetracking to perform remote usability tests on users. Remote usability testing enables to test users in their natural environment. Eyetracking is one of the most popular techniques for usability testing in the laboratory environment. We decided to try to use this technique in remote tests. We used standard web camera with freeware software. Our experiment showed that such method is not perfect, but it could be a good addition to the standard remote tests, and a foundation for further development.

Keywords: Eyetracking, Usability, Remote Usability Testing, Human-Computer Interaction.

1 Introduction

In modern usability testing, remote tests are becoming more and more popular. Because of lack of time and money, companies are looking for alternatives for standard tests with users. Those tests require a place to test, gathered users, moderator and equipment. This situation lead to idea of remote usability testing. Its main goal is to test users in their natural working place, without any sophisticated equipment. Users in their own environment are behaving more naturally, like they would normally do while using the given website [4]. Moreover we do not need to gather all the users at one time, we can work with them when they have the time to take part in the test. Also we can have participants from different cities or even countries that would normally not visit our laboratory. Furthermore comparisons of the results of standard laboratory and remote testing have shown that participants find the same usability issues on tested pages with both methods [2], [6], [9].

So far remote usability tests are evaluated using standard methods such as remote surveys and video conferences [1], [7]. We can also use some traditional laboratory usability testing methods in remote environment [8]. One of the most popular tool for standard usability testing is eyetracking [3], [5]. The biggest drawback of eyetracking technology is its cost. The equipment is very expensive and companies which perform such tests usually charge a lot of money for such tests. We tried to perform remote usability test with eyetracking, using low cost hardware, such as ordinary web camera and free software.

2 Tools used in experiment

The main equipment used during this research was a casual web cam Logitech Quick Cam Pro 9000. Software used in this experiment was:

1. Piotr Zieliński Opengazer¹ .Net port made by Przemysław Nibylowicz². This application enables gazetracking using ordinary webcam. It is freeware open source software. After selecting feature face points on the video image, user calibrates the program by looking at the appearing squares. Next, when calibration is finished, line of gaze is tracked by the program. We slightly modified this application, so it stores the data of all the points that users is looking at in the text file (Fig. 1).

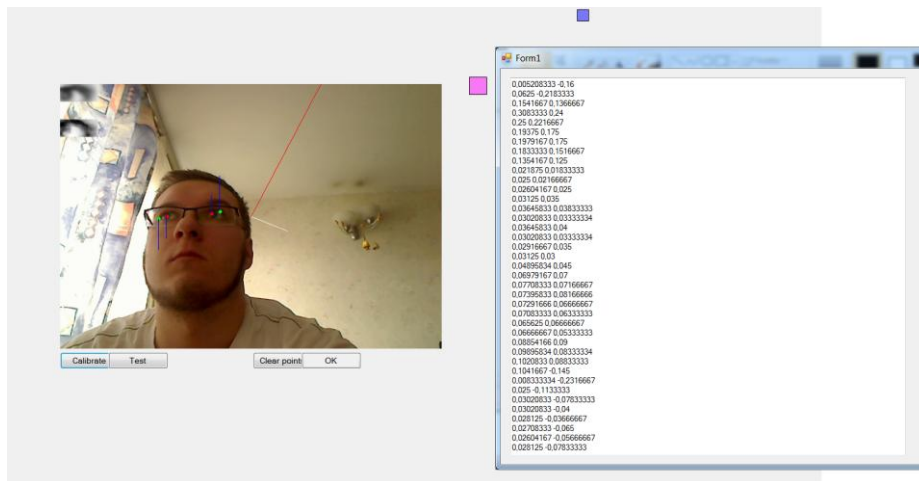


Fig. 1. Application after the calibration process. Blue square shows where the participant is looking and in the background we have a window that shows the coordinates of the gaze.

The points in the application have coordinates from -1 to 1 (float variable), so to store them as actual points on the screen, we are transforming them as shown below:

```
int okoX = (int)((eventArgs.EyeX + 1.0f) * (w / 2));
int okoY = (int)((eventArgs.EyeY + 1.0f) * (h / 2));
```

We add 1.0 to the float value (eventArgs.EyeX or eventArgs.EyeY) and then we multiply this value by height (h) or width (w) of the screen divided by two. Then the value is parsed to integer. Furthermore we needed to transform those points so

¹ <http://www.inference.phy.cam.ac.uk/opengazer/>

² <http://netgazer.sourceforge.net>

they could be used in the heatmap generator application (point 2 below). We needed to transform the screen parameters to image parameters:

```
int imageX = (int) ((1024*okoX)/w);
int imageY = (int) ((768*okoY)/h);
```

We received new coordinates for the image by calculating the given point with the image size and screen size.

- JavaScript application for creating heatmaps³. Created by Michael Dungan, released under the MIT license. This application creates heatmaps on images taking mouse movement as an input data (Fig. 2). However it has the functionality to import points, so we used it to generate heatmaps for our test. It takes as parameters the offset of the picture and a “mousemove mask” for smooth rendering, so we omitted this last parameter and put ‘3’ as a default value (center). Additionally we needed to run the browser in full screen mode, so the coordinates of the points calculated in Opengazer .Net port application would be adequate.

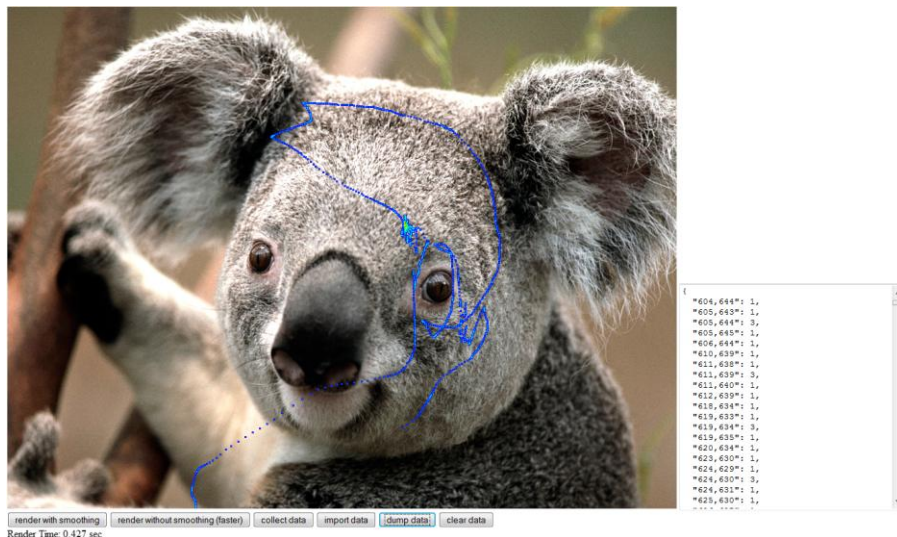


Fig. 2 Application for creating heatmaps from mouse input as well as from imported points (Source for koala bear picture: Microsoft Windows 7 sample images).

- Real VNC⁴ – simple application for desktop sharing.
- Skype⁵ – popular application that enables talking through microphone over the internet.
- CamStudio⁶ – simple application for recording actions from computer screen.

³ <https://github.com/xxx/heatmap>

⁴ <http://www.realvnc.com>

⁵ <http://www.skype.com>

3 Experiment

The goal of our experiment was to try to perform remote usability eyetracking test and analyze if our method is suitable for future development. We tested it with five users, on different computers but with the same webcam (Logitech Quick Cam 9000). During the test, first thing that we did was to connect to user's computer via Skype and Remote VNC. After that we instructed the user how to position the camera and how to calibrate the eyetracking application. This was the toughest part for all the users, because the calibration process in this application requires a lot of patience. Application sometimes crashes, and calibration often needs to be repeated few times until it is correctly set. When the results of calibration were satisfactory we asked our user to start CamStudio and perform some simple actions on google.com web page, such as to log in. While user was working with the page we were able to see where he is looking, because small blue square was showing that position (Fig. 3). It is a very helpful thing for the moderator, because he can observe at which elements the user is looking during the test and have more control over what user is doing.



Fig. 3. Google.com website during the test, small blue square shows the position of the participants gaze.

After the experiment we took the coordinates of the user's gaze on the screen (stored in a text file) and we imported them to the heatmap application. It allowed us to create some simple heatmaps for our test (Fig. 4).

⁶ <http://www.camstudio.org>

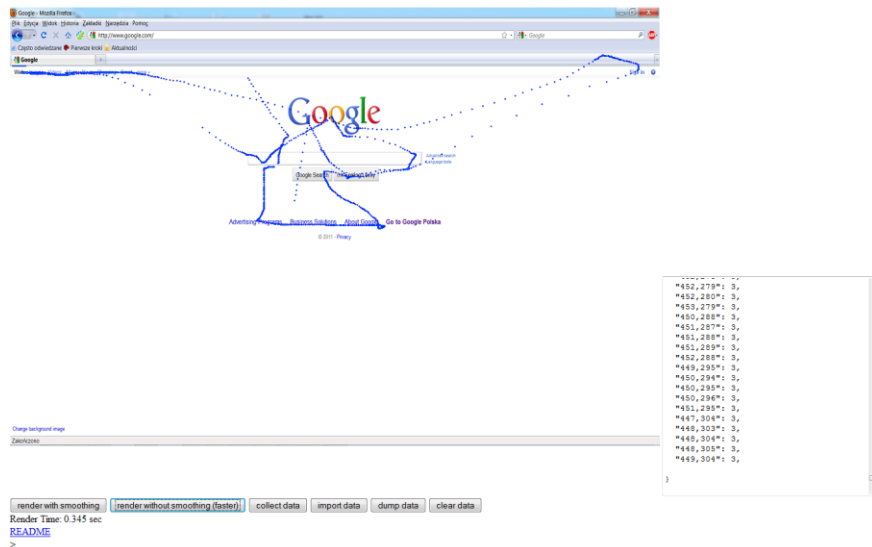


Fig. 4. Simple heatmap for google.com website created with JavaScript heatmap generator.

To sum up the experiment, our method of remote usability eyetracking test can provide the moderator with:

1. Verbal remarks and comments of the tested users, as in standard remote test.
2. Observation of user actions and where user was looking during the test via Remote VNC.
3. Recordings from the point 2 provided by CamStudio.
4. Text file with the coordinates of the points on the screen on which the user was looking during the test. Those points can be used to generate some visual reports.

4 Conclusions and future work

Our experiment has shown that it is possible to perform remote usability eyetracking tests. Moreover we obtained a lot of valuable data that could be processed for usability analysis. However our solution has some drawbacks that need to be addressed. First of all user has to perform many actions before the test, such as setting up the camera and calibrating the eyetracking software. In future we will try to improve this process, so the users will not need to perform so many operations. Secondly, using a web cam and Opengazer .Net port we need to perform calibration many times to obtain eyetracking data, which precision can be compared with professional eyetrackers in the laboratory environment. So far we managed to obtain the precision which is about two times worse than in professional eyetrackers, so there is still room for further development here. We will also try to improve calibration process, so it would be faster and more effective. Last thing is that we needed to put a lot of effort into creat-

ing a single heatmap for our test. We need to create an application that could quickly transform received points into heatmaps, gaze plots and other useful reports.

In conclusion our method needs a lot of improvements, but it is definitely a good starting point for creating a remote usability eyetracking testing methodology and a platform for such tests. Remote user testing is the future of usability tests, so introducing new methods and techniques to improve them is a very beneficial thing.

5 Acknowledgements

This work has been partially supported by the Polish Ministry of Science and Higher Education within the European Regional Development Fund, Grant No. POIG.01.03.01-00-008/08.

6 References

- [1] Andreasen M.S., Nielsen H.V., Schröder S.O., Stage J., What Happened to Remote Usability Testing? An Empirical Study of Three Methods, www.takebay.net/data/chi07/docs/p1405.pdf (20.03.2011)
- [2] Brush B., Ames M., Davis J., A Comparison of Synchronous Remote and Local Usability Studies for an Expert Interface, <http://delivery.acm.org/10.1145/990000/986018/p1179-brush.pdf> (21.03.2001)
- [3] Duchowski A. T., Eye tracking methodology: Theory and practice, London, Springer-Verlag Ltd, 2003, 205-300
- [4] Moha N., Li Q., Seffah A., Michel G., Towards a Platform for Usability Remote Tests via Internet, http://www.ptidej.net/Members/mohanaou/paper/OZCHI2004/OZCHI2004_Moha.pdf (20.03.2011)
- [5] Mohamed A.O., Pereira Da Silva M., Courbolay V., A history of eye gaze tracking, 2007, http://hal.archives-ouvertes.fr/docs/00/21/59/67/PDF/Rapport_interne_1.pdf (12.03.2010)
- [6] Oztoprak A., Erbug C., Field versus Laboratory Usability Testing: a First Comparison, www.aydinoztoprak.com/images/HFES_Oztoprak_.pdf (21.03.2011)
- [7] Petrie H., Hamilton F., King N., Pavan P., Remote Usability Evaluations with Disabled People, Montréal, Québec, Canada, CHI 2006 Proceedings, <http://www-course.cs.york.ac.uk/rmh/p1133-petrie.pdf> (20.03.2011)
- [8] Scholtz J., Adaptation of Traditional Usability Testing Methods for Remote Testing, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.2141&rep=rep1&type=pdf> (21.03.2011)
- [9] Tullis T., Fleischman S., McNulty M., Cianchette C., Bergel M., An Empirical Comparison of Lab and Remote Usability Testing of Web Sites, <http://home.comcast.net/~tomtullis/publications/RemoteVsLab.pdf> (21.03.2011)