Usability Aspects of Eye Gaze Tracking Systems

Stojance Panov[1], Saso Koceski[1], Natasa Koceska[1]

[1] Faculty of Computer Science, University „Goce Delcev"-Stip,
bul. Krste Misirkov bb. 2000 Stip, Macedonia

{stojance.panov, saso.koceski, natasa.koceska}@ugd.edu.mk

Abstract. With the rapid evolution in computer technology, there's an augmented need to eminently dedicate attention to the computer-aided interaction, including crucial design aspects, implementing and evaluating the interfaces that provide this type of communication. Various techniques for human-computer interaction have been used, commencing with keyboards, printers, moving on with gesture interaction, speech interaction, touch screens, eye gaze tracking and many more. Most of these techniques are still analyzed and examined if they could ensure an ease at performing given tasks, such as moving the mouse cursor, selecting menus, moving or dragging objects on the computer screen, thus helping users with disadvantages to interact with the workstation. This paper describes usability study of an existing eye-tracking system and evaluate its correctness and calculate its error percentage, which can lead us to the deduction if this system would present an efficient interaction and provide facilitation of performing certain tasks, such as reaching a goal, following a trajectory, steering through straight tunnels and steering through circular tunnels.

Keywords: Human-computer interaction, eye-gaze, tracking systems, usability, Fitt's law.

1 Introduction

Many recent computer interfaces have been designed in a fashion that enables users that have disadvantages to easily adapt to them and can flexibly interact with. That's precisely why alternative interaction techniques are seriously considered more frequently, such as controlling the main activities using eye gaze. Eye-tracking can be simply defined as a type of human-computer interaction where the system providing this functionality can calculate the coordinates where one's look is located on the screen. This type of system is also capable of making a movement between two locations on the screen obtained by the eye gaze of the user, and is also video based.
Concretely, eye-movements on the computer screen can be tracked and saved for later examination, which is a common task if we want to check whether given interface is user-friendly. Very frequent task that should be mentioned here is moving the mouse cursor with the user's gaze, which is essential to people with disabilities to have as a

way to interact with the workstation. This actually means that any user is capable of controlling with his eyes and uses them as an output which is processed by the computer and it's the workstation's task to determine the direction of the user's gaze.
Eye-tracking system should offer several benefits. The main goal it should reach is providing facility of use, which means that the user won't have to move his hands to move the cursor, and there is also no additional strain when moving the mouse cursor with one's eyes. The maintenance issues are reduced, which means all a user has to do is turn on a camera and he's ready to perform all the required activities. Since user's eyes can move quickly, this implicates that these types of interfaces must be fast and efficient. The eye-tracking system ensures that the user has the required attention through the human-computer interaction, which can be used to determine if the person is reading, how old a person is, etc.

Although a lot of systems for eye tracking were developed and a lot of scientific works have been published, to the best of the authors' knowledge just few of them are presenting some results of evaluation of these types of interfaces. This papers aims to propose an evaluation methodology and to presents evaluation results of eye-gaze tracking interfaces, which may be suitable for designers and developers at one side and for the users of these types of interfaces (mainly disabled persons).

2 Related work

The earliest researches made in the field of eye-tracking, according to Drewes [1], origin about 100 years ago, when there were several methods to describe methods for detecting an eye-movement, such as ghost images, coupling the eyes and the ears, electro-oculographic techniques [2] and using photo-graphic methods to record those kind of activities. Later, Fitts et al. have an interest of determining the usage of cockpit controls on planes, so he used a motion picture camera to track the eye gaze of the pilots [3].
The start of video-based eye gaze tracking systems origins back to 1981, when Bolt showed that computers were becoming efficiently enough to perform these tasks [4]. Since then, eye-tracking systems became widely useful, presenting a good alternative to manipulating user interfaces which provided disabled users with the ability to perform certain computer tasks, such as moving the cursor, object movement and manipulation etc.
There have been several studies and researches connected to this field of human-computer interaction. Many of them provide a good implementation of an eye-tracking system, and also some of them are open-source and free to use to perform deeper researching and evaluations. They perform in a way of detecting the pupil and recording its' movements. Such an algorithm that used this method was presented by Dongheng, which is called the starburst algorithm, using a head mounted eye-tracking system that had an accuracy of around one degree of visual angle [5]. This algorithm was also implemented in the OpenEyes low-cost eye-tracking system. There are also many other efficient systems that perform eye gaze detection by tracking the reflection of the cornea of the pupil [6]. Gustafson et al. [7] have presented a portable eye tracking device. Varsha et al. gave an example implementation of eye gaze

tracking by using neural networks. Recently there are many approaches for solving the problem of human-computer interaction for people with disabilities using even commercial eye-tracking systems [9].

3 The ITU gaze tracker system

The ITU Gaze Tracker [10] is a video-based eye-tracker, which means that the direction of the eye gaze will be easily determined by using a video-camera. It was developed at the IT University of Copenhagen by the Gaze Group. This system automatically detects the available video-camera and since it recognizes it, it provides a simple user interface which can facilitate the user experience and thus giving the ease of manipulating the cursor with the eye gaze. The ITU Gaze Tracker can obtain the desired functionality using any webcam or infrared night vision camera. This system was chosen to be evaluated because of its openness, possibilities for custom development and because it principles of functioning are the same as others.
Using the Gaze tracking library [11] which is an open source framework (developed by ITU) for eye tracking using off-the-shelf components, a special evaluation software has been develop. The application was develop in the programming language C# and was intended to work with a standard web-camera.
For the testing and evaluation purposes of this research, several different GUI designs were developed in order to test various tasks, such as reaching a goal, steering through straight and circular tunnels and following paths, using the eye gaze controlled mouse. The developed interfaces and correspondent tasks are given and thoroughly discussed in the following.

4 Experimental evaluation tasks and environment

4.1 Reaching a goal

In the first evaluating task, an elementary interface (Fig 1) was developed. It consists of a point on the left side and a resizable rectangle on the right. The user in this particular assignment is supposed to bring the cursor from the point to the rectangle as soon as possible. The calculations applied were followed by the Fitts' law, which defines the movement time of the cursor as follows:

$$MT = a + b * ID \quad (1)$$

where

$$ID = \log_2(A/W + 1) \quad (2)$$

A is the distance between the point and the goal and W is the width of the rectangle.

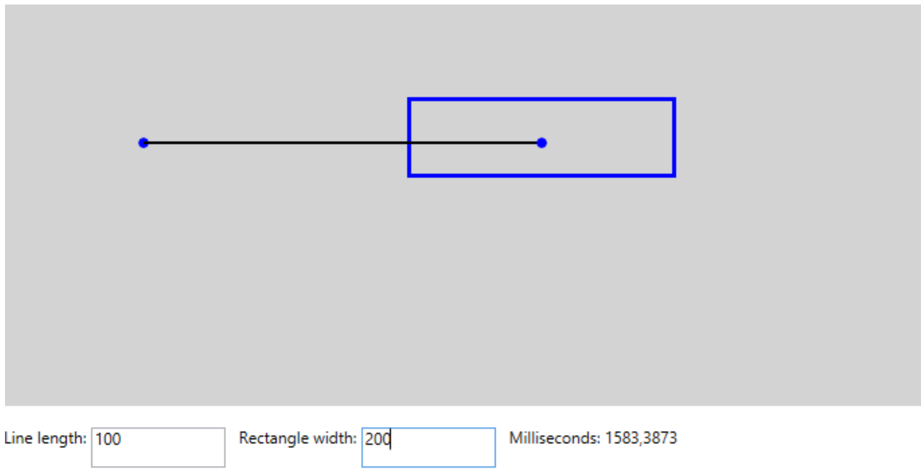

Fig. 1. GUI for evaluation of the goal reaching task.

In equation (1), the factors *a* and *b* are determined by calculating linear regression on the obtained samples. The distance between the point and the rectangle was moving in the interval [200; 600] with step of 50, and the width of the rectangle was ranging in the interval [50; 400] with step of 50. Every possible combination was taken into consideration.

4.2 Steering through straight tunnel

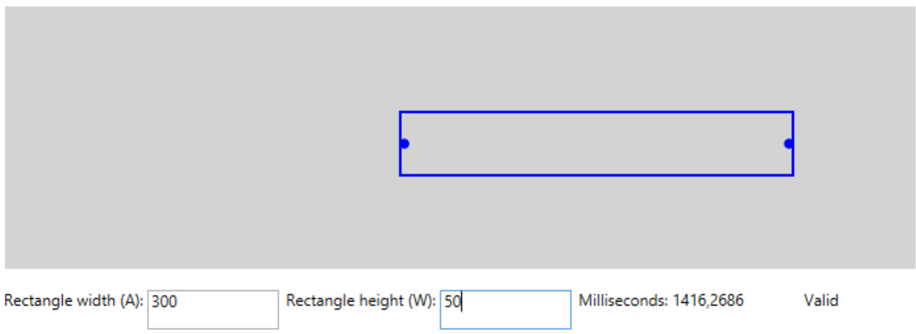

Fig. 2. GUI for evaluation of the steering through straight tunnel task.

In the second task, we designed a minimal interface which provided changeable width and height of the rectangle presenting the tunnel we need to use to reach from the left to the right side. Whenever the user went out of the tunnel before reaching the end, we counted it as an error and an invalid test.

For the purpose of this experiment, again we needed to gain results from the Fitts' law, by using the equation (1), but this time ID is obtained by:

$$ID = A/W \quad (3)$$

where A is the length of the tunnel, and W is the width of the tunnel. A was driving in the interval [200; 600] with a step of 50 and W was ranging in values [50; 300], also with a step of 50.

4.3 Steering through circular tunnel

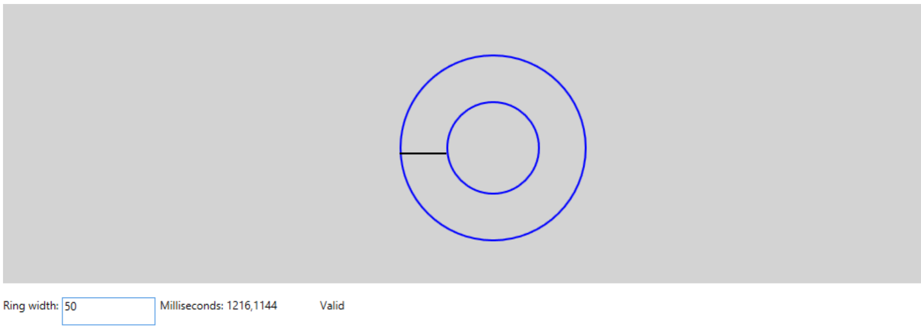

Fig. 3. GUI for evaluation of the steering through circular tunnel task.

Continuing with our research, we constructed a flexible interface which allowed us to easily change the width of the circular tunnel. The user was supposed to make one lap through the tunnel, so that we could measure the time of completing the task. Equation (1) was used here too, given ID which is calculated by (3), but this time the length of the tunnel A is determined as:

$$A= 2*R*\pi \quad (4)$$

where

$$R= (R_1+R_2)/2 \quad (5)$$

given R1 and R2 as the radiuses of the inner and outer circle respectively. Here, W was changing in the interval [50; 250] with a step of 20.

S. Loshkovska, S. Koceski (Editors): ICT Innovations 2015, Web Proceedings, ISSN 1857-7288

4.4 Path following

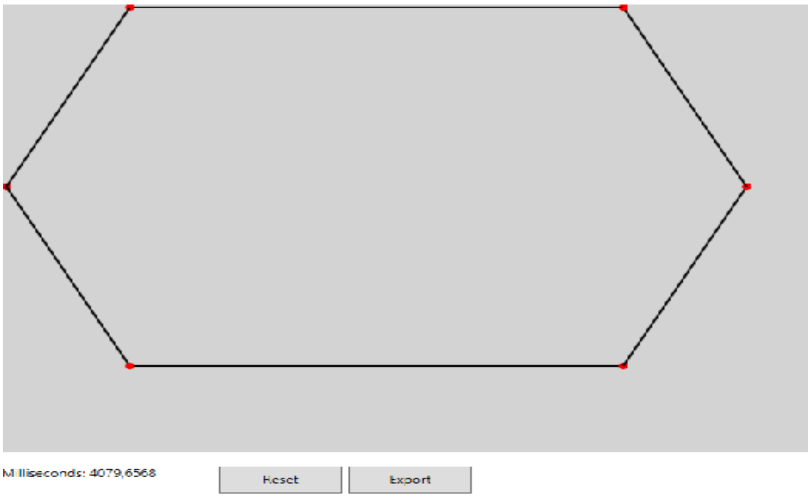

Fig. 4. GUI for evaluation of the path following task.

For the next step of this research, we designed a path of a hexagonal form to measure the effectiveness of the eye-tracking system in providing the user with the ability to give him facilitation in adapting to this method of human-computer interaction. In order to calculate the error of movement at following the path, we used the path repeatability formulas that follow:

$$RT_p = \max RT_{pi} = \max[\bar{l}_i + 3S_{li}];\ i = 1...m \tag{6}$$

Where

$$\bar{l}_i = \frac{1}{n}\sum_{j=1}^{n} l_{ij} \quad S_{li} = \sqrt{\frac{\sum_{j=1}^{n}(l_{ij} - \bar{l}_i)^2}{n-1}} \quad l_{ij} = \sqrt{(x_{ij} - \bar{x}_i)^2 + (y_{ij} - \bar{y}_i)^2}$$
$$\bar{x}_i = \frac{1}{n}\sum_{j=1}^{n} x_{ij} \quad \bar{y}_i = \frac{1}{n}\sum_{j=1}^{n} y_{ij} \tag{7}$$

m is number of calculated points along the path, and n is number of measurement cycles.

5 Evaluation methodology

We recruited 50 participants, 40 were male and 10 female; 38 were students and 12 non-students. All of the participants were aged 21-38. We recruited each person separately and ensured that none of them has eye problems or wearing glasses or lenses. None of them has previous experience with an eye-gaze tracking systems.
To evaluate how useful is the eye-gaze tracking system, the user study which including three stages was conducted. During the first stage, the players learned the interface basics and working principles. During this interface a calibration was performed and the developed application has "learned" how to adapt to the particular user's eye characteristics. During the second stage, the participants tried practically the developed GUIs' and tasks. Each participant was given at most three times to try. These sessions and users' behavior was observed and notes were taken. All the users were instructed to complete the tasks as fast as possible. In the final stage, the participants performed each of the given tasks for 10 times and average results for each of them were calculated.

6 Results and discussion

The results of experimental evaluation of given tasks and developed GUIs' are presented in the following. Results from the linear regression line fitting for the goal reaching and steering through straight tunnel tasks are presented in Fig 5 and Fig 6.

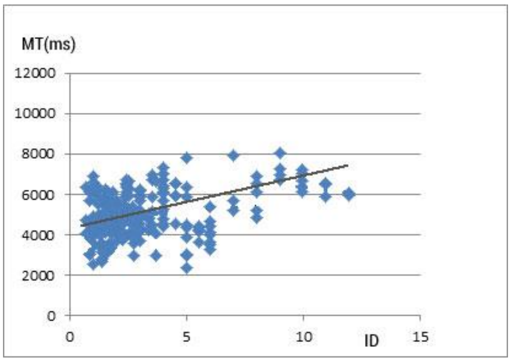

Fig. 5. Results from the linear regression line fitting for the goal reaching task.

In the first case the results obtained yield a regression line with movement time (ms) predicted as:

$$MT = 4300 + 116*ID \quad \textbf{(8)}$$

and in the second case:

$$MT = 4400 + 125*ID \quad \textbf{(9)}$$

with a correlation of r = .942 and r = .915 correspondingly. Correlations above .900 are considered very high for any experiment involving measurements on human subjects. A high r suggests that the model provides a good description of observed behavior. Both equations have large intercepts.

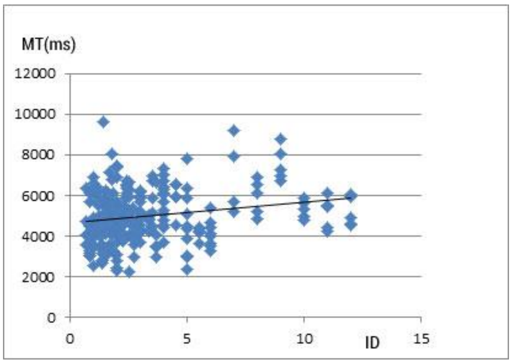

Fig. 6. Results from the linear regression line fitting for the steering through straight tunnel task.

The implication, of course, is that a movement task rated at ID = 0 bits will take 4.3 and 4.4 seconds correspondingly. The slopes of 116 ms/bit and 125 ms/bit translate into bandwidths of 8.6 bits/s and 8 bits/s correspondingly. This implies that this interface requires big effort from the user and could be classified as difficult to use, because the operations take relatively big amount of time.
Result from the linear regression line fitting for steering through circular tunnel task is presented in Fig 7. They yield a regression line with movement time(ms) predicted as:

$$MT = -2000 + 1000*ID \quad \textbf{(10)}$$

A negative intercept implies that, as tasks get easier, a point is reached where the predicted movement time is negative. This, of course, is nonsense and indicates a flaw in the application of the model or the presence of uncontrolled variations in the data.

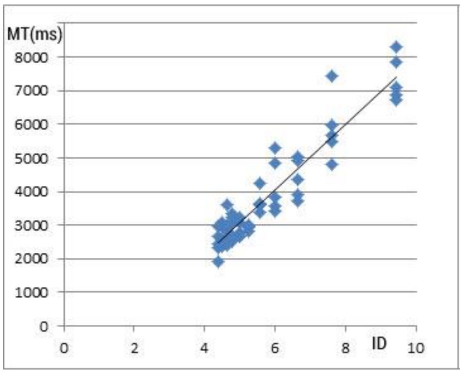

Fig. 7. Results from the linear regression line fitting for the steering through circular tunnel task.

The values for path repeatability task upon 10 measurements (averaged for all users) were calculated and they are given in Table 1.

Table 1. Repeatibility results.

Test number	Repeatability (pixels)	Test number	Repeatability (pixels)
1	32	6	35
2	45	7	41
3	28	8	44
4	38	9	37
5	52	10	39

The results imply that this interface should be considered very carefully when we have to deal with tasks that require precise actions. Moreover, in order to check the development of the user experience, i.e. how much time does one need to gain a better experience and to adapt to the eye gaze method of manipulating the cursor, the learning curve was calculated for the steering through a straight tunnel task (taking fixed values A=300pixels, and W=200. This experiment was conducted for 30 times and gained the results given in Figure 8.

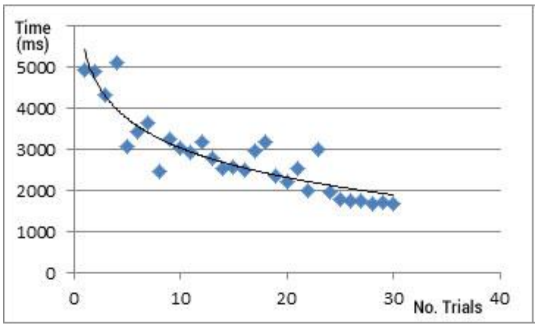

Fig. 8. Learning curve (power law of practice).

The results show an almost perfect fit with the power law of practice formula [12], with α=0.4. One can easily observe that the largest improvements in speed are made during the very first trials. Therefore, we should be careful with generalizing timing results from first-time users. The results are also useful to reach a conclusion regarding the number of trials after which there is no more significant learning. In our case for the given task it was determined that it corresponds to trial number 29.

5 Conclusion

This paper describes comprehensive usability study of eye-tracking system, which is presenting experimental results that may be useful both for developers and end users. The experimental validation was performed on a certain tasks, such as reaching a goal, following a trajectory, steering through straight tunnels and steering through circular tunnels.

References

1. Drewes, H. Only one Fitts' law formula please! In Proceedings of the 28th of the international Conference Extended Abstracts on Human Factors in Computing Systems. CHI EA '10. ACM Press (2010), 2813 – 2822
2. Duchowski, T. D.: Eye Tracking Methodology: Theory and Practice. London: Springer-Verlag (2002), ISBN 1-85233-666-8.
3. Fitts, Paul M., Richard E. Jones, and John L. Milton. "Eye movements of aircraft pilots during instrument-landing approaches." Ergonomics: Psychological mechanisms and models in ergonomics 3 (2005): 56.
4. Bolt, R.A. Eyes at the Interface. In Proceedings ACM Human Factors in Computer Systems Conference, ACM Press (1982), 360 – 362.
5. Dongheng Li, Winfield, D. and Parkhurst, D. J. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Volume 3, (2005), 79
6. Ohno, T., and Mukawa, N. A Free-head, Simple Calibration, Gaze Tracking System That Enables Gaze-Based Interaction. In Proceedings of the 2004 Symposium on Eye Tracking Research & Applications. ETRA '04. ACM Press (2004), 115 – 122
7. Gustafsson, S., Andersson, J. Engwall, E., Hainzl, R., Leksén, L., Cederlund, M. "Portable eye tracking device." U.S. Patent Application 14/281,587, May 19, 2014.
8. Varsha, P., Jasmeen Patel, and Nishit Gautam. "Eye Gaze Tracking System using Neural Networks." IJRECE 2, no. 2 (2014): 68-70.
9. Benjamin, David P., Ann M. Mastergeorge, Andrea S. McDuffie, Sara T. Kover, Randi J. Hagerman, and Leonard Abbeduto. "Effects of labeling and pointing on object gaze in boys with fragile X syndrome: An eye-tracking study." Research in developmental disabilities 35, no. 11 (2014): 2658-2672. 5. San Agustin, J., Skovsgaard, H., Mollenbach, E., Barret, M., Tall, M., Hansen, D.
10. W., and Hansen, J. P, Evaluation of a low-cost open-source gaze tracker, In Proceedings of the 2010 Symposium on Eye-Tracking Research&Applications, 2010.
11. ITU Gaze Tracker: http://www.gazegroup.org (last visited March 2015).
12. T. Stafford, M. Thirkettle, T. Walton, N. Vautrelle, L.Hetherington, et al. (2012) A Novel Task for the Investigation of Action Acquisition. PLoS ONE 7(6): e37749.