# Comparison and Analysis of the Eye Pointing Methods and Applications

Piotr Chynał and Janusz Sobecki

Institute of Informatics, Wroclaw University of Technology
Wyb.Wyspianskiego 27, 50-370 Wroclaw, Poland
Piotr.Chynal@student.pwr.wroc.pl, Janusz.Sobecki@pwr.wroc.pl

**Abstract.** In the paper we present comparison and analysis of the selected eye pointing methods and applications. Eye pointing enables communication with the computer system using human eye movements tracking. We analyze not only available eyetracking solutions but also those developed at our laboratory (i.e. ljo_gazetracker) are using specialized gazetracking devices. Those solutions were compared by means of series of experiments carried out with a group of users. These experiments showed that Dasher is the most effective gaze-writing application, and that ljo_gazetracker is the most precise eye pointing system. However, it needs application of quite sophisticated and pretty expensive technology. Thus Opengazer seems to be a good freeware alternative.

**Keywords:** Eyetracking, Human-Computer Interaction.

## 1 Introduction

In modern applications, interfaces are set to enable fast choices and selections, and the most popular control devices are a computer mouse, a touch-pad and a touch-screen [9]. However the development of the technology enables us to deliver new ways of interaction with users. The choice of element using a computer mouse is limited just to moving over the target and pressing a button. With eyetracking device it is even more simple. The line of gaze goes over a desired target, and it is automatically selected. The research shows that this way is faster and more effective than using a traditional computer mouse [8]. Therefore more and more applications with eyetracking interfaces are being developed. Using eyetracking instead of computer mouse enables people with physical disabilities or busy hands to gain access to the computer [7]. For some people it might be the only way to contact the outside world. This work presents chosen eyetracking solutions in human-computer interaction, and their recommendations, based on needs and financial conditions. To reach this target we needed to analyze existing eyetracking applications, projects and also create our own solutions. Then we have carried out two experiments with users that enabled us to choose the best applications and methods for writing with gaze, controlling the computer and for gaze detection. Additional information about eyetracking, its history and other applications can be found in [1], [3], [4].

J.-S. Pan, S.-M. Chen, and N.T. Nguyen (Eds.): ICCCI 2010, Part I, LNAI 6421, pp. 30–38, 2010.

## 2 Eyetracking Tools Used in the Experiments

Experiments were conducted in the Software Quality Laboratory, at the Wrocław University of Technology. The main equipment used during this research was ASL 6000 eyetracking module [2]. It consists of two computers (one for user, and one for controlling with EyeTrac 6000 software), Pan Tilt module with two cameras, and ASL module with monitor. One of the Pan Tilt eyetracking cameras was using infrared to detect pupil and cornea reflection, and the second one was tracking face and head movement. Other equipment used in the experiments was casual web cam Logitech Quick Cam Pro 9000. Software used in those experiments:

1. Piotr Zieliński Opengazer[1] .Net port made by Przemysław Nibyłowicz[2]. 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.
2. MyEye. It is an application developed by Marcelo Laginestra[3]. It enables gazetracking and controlling of the pointing device. To enable eyetracking in this application we need to set different parameters so that we get blue circle around the pupil. After looking in every corner of the screen the calibration is complete. This program is dedicated to the infrared cameras, so we had some problem to get the blue circle around the pupil right with ordinary Logitech Quick Cam Pro 9000. Therefore we decided to use Opengazer .Net port during the experiments.
3. Dasher. This application was created by Interfence Group[4]. Version of Dasher used in experiments was 4.11. In this application user can input text using any controlling device, such as mouse, touchpad, touch screen etc. Writing with Dasher is very easy. User chooses letters appearing on the screen, with cursor. Dasher uses zooming interface, so the letters, on which the cursor is pointing are zooming in. Those letters form words. System uses probalistic methods to give the most probable letters more space [5].
4. GazeTalk5. This application is a complex system that enables to control many different elements, such as browsing internet, listening to the music and many more. It was developed by Eye Gaze Interaction Group at IT University in Copenhagen[5]. Writing in GazeTalk5 is performed by choosing the adequate field, which contains letter or option. System uses prediction mechanisms to suggest the words that he thinks we want to write. Choosing the letter can be done by one of the three ways. First of them is dwell time, which means, that when we look at desired letter, it will be accepted after some time. We can set that time in the options menu. Other options of choosing are standard clicks, and by scanning all the fields, and selecting the highlighted letter when clicked. Besides its rich functionality GazeTalk5 enables running Dasher straight from the main menu, so we can write using that application, and send the text straight to GazeTalk5.

[1] http://www.inference.phy.cam.ac.uk/opengazer/
[2] http://netgazer.sourceforge.net
[3] http://myEye.jimdo.com
[4] http://www.inference.phy.cam.ac.uk/dasher/
[5] http://www.gazegroup.org/research/15

5. FlashKeyboard created by Piotr Chynał. This application enables writing without using a keyboard. It was developed in Adobe Flash CS4 with ActionScript 3.0. Selecting the letters is possible in two ways. User can move cursor over desired letter and wait for two seconds or press space key during that time, to make faster selection. Therefore this application can be used by disabled people and by casual users.
6. Ljo_gazetracker, the application created by Dorota Molik and Wojciech Pietrowski in Software Quality Laboratory, at the Wrocław University of Technology. It was developed and improved by Piotr Chynał. It consists of two parts, first that gathers the data from ASL6000 module and the second which is a Java program that transforms received parameters from the first component and determines the position on screen user is looking at. Furthermore, this program enables control of the cursor with line of gaze.


**Fig. 1.** Interface of the FlashKeyboard

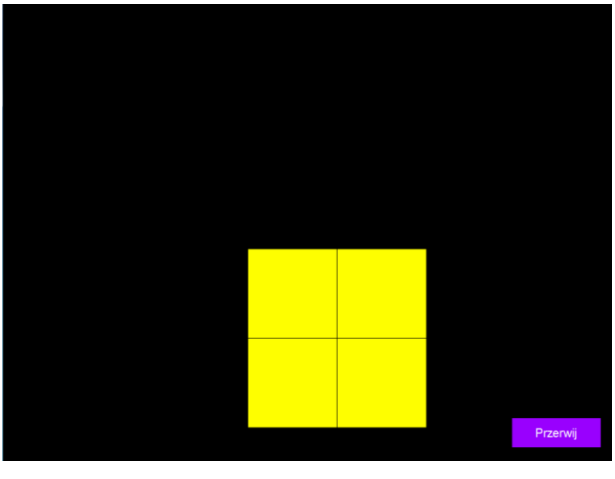


**Fig. 2.** Precision_test screen shot, showing the first square of the test

7. Precision_test. This program, likewise ljo_gazetracker was created by Dorota Molik and Wojciech Pietrowski. It was implemented in Adobe Flash CS3 with ActionScript 2.0. During working with this program, yellow squares are appearing on the screen. The idea of precision_test is the following, first the user look at yellow square displayed on the screen, so the cursor position is within this square, then after 1,5 seconds, the square is selected, and next the smaller one appears in different place. When square is so small that the user is not able to point gaze within it, the program stops, and prints the size of last selected square in pixels.

## 3 Experiments

We conducted two experiments. Goal of the first of them was to select the most effective application for writing with gazetracking. In the second experiment the objective was to analyze different eye pointing solutions.

The first experiment was carried out with ten people – students from different universities and faculties. For gazetracking we used Logitech Quick Cam 9000 Pro camera and .Net port of Opengazer. The test consisted of writing three short phrases by the respondents. The phrases were the following: “test”, “hello world” and “i am writing with my eye”. They were typed using the following tested programs: GazeTalk5, Dasher and FlashKeyboard that were used by each user in a random order. During the experiment the time and the number of mistakes were recorded. After the test, users were asked to describe verbally their reflections on the used programs.


**Fig. 3.** ASL 6000 eyetracking module in Software Quality Laboratory, at the Wroclaw University of Technology

The second experiment was also conducted on ten users. They all were students of the Wroclaw University of Technology. We have created two stands for this experiment. In the first of them, the respondents worked on Pan Tilt camera with ljo_gazetracker program. Second stand was with Logitech Quick Cam pro 9000 camera and .Net Opengazer port.

In this experiment, users were asked to perform identical tests on both stands. The first test was performing precision test with Flash precision test application. Next test was to write phrase "eye writing test" with Dasher. Last task was to input the web address of one of the Polish Internet portals in GazeTalk5, and then go to mailbox bookmark. During those tests, the results were noted. For precision test it was the result printed by the application. For Dasher and GazeTalk5 we measured time. After the experiment users were asked to express their thoughts on the tested applications and cameras.

## 4 Results of the Experiments

We present the results of both tests in Tables 1 to 4. They present particular results and comparisons between programs and eyetracking solutions.

**Table 1.** Results of the first experiments, with sum of time from all three tasks, sum of mistakes made by users in all three tasks and average time of writing

| | Dasher | | GazeTalk5 | | FlashKeyboard | |
|---|---|---|---|---|---|---|
| | Time [m:s:ms] | Amount of mistakes | Time [m:s:ms] | Amount of mistakes | Time [m:s:ms] | Amount of mistakes |
| Person 1 | 01:29:91 | 0 | 05:41:93 | 1 | 04:11:49 | 1 |
| Person 2 | 03:24:50 | 0 | 07:20:76 | 2 | 04:16:67 | 0 |
| Person 3 | 03:03:71 | 0 | 06:42:97 | 3 | 05:07:54 | 1 |
| Person 4 | 02:37:82 | 0 | 07:46:98 | 1 | 05:13:58 | 0 |
| Person 5 | 02:00:72 | 0 | 07:12:04 | 2 | 05:58:25 | 2 |
| Person 6 | 02:45:33 | 0 | 08:18:26 | 4 | 05:12:99 | 0 |
| Person 7 | 02:25:25 | 0 | 09:00:04 | 3 | 06:17:64 | 1 |
| Person 8 | 02:48:30 | 0 | 07:38:68 | 1 | 05:55:44 | 2 |
| Person 9 | 02:19:78 | 0 | 07:24:06 | 3 | 04:54:97 | 0 |
| Person 10 | 02:45:68 | 0 | 06:59:04 | 2 | 06:32:73 | 1 |
| Average total time[m:s:ms] | 02:25:31 | | 07:33:66 | | 05:32:71 | |
| Sum of mistakes | 0 | | 22 | | 8 | |

**Table 2.** Comparison of the applications tested during the first experiment

| | Dasher | GazeTalk5 | FlashKeyboard |
|---|---|---|---|
| Speed of writing | Very fast, sum of average times for all phrases was – 02:25:31minutes | Slow, sum of average times for all phrases was - 04:52:03 minutes | Fast, average time for all phrases was 03:32:54 minutes |
| Number of mistakes | No mistakes | Lots of mistakes, caused mainly by selecting wrong letters | Little amount of mistakes, caused by low precision of cursor |
| Complexity of usage | At first it was hard, but after some time very easy | Easy, but sometimes it is hard to find particular letter, when application suggests other letters, than those that we want to write | Simple and intuitive, created on base of computer keyboard |
| Ease of use | Very comfortable | Not too comfortable, it wears eyes and the „Midas touch" problem occurs | Comfortable |
| Interaction technique | Selecting further letters with cursor, combining them to words | Selecting by moving the cursor on desired letter and waiting for two seconds | Selecting letters by pressing space key or by moving the cursor over letter and waiting for two seconds for its selection |
| Other remarks | Best ranked application by respondents, very good algorithms suggesting most adequate words, working even with worse calibration and precision | Not the best of interfaces. This application has many modules, and the possibility to write using Dasher | All the letters were visible during the work, intuitive interface |

**Table 3.** Results of the second experiment with results of the precision test, time of writing test, time of Internet portal test and average values of all of them

| | Camera Pan Tilt + ljo_gazetracker | | | Camera Logitech + Opengazer | | |
|---|---|---|---|---|---|---|
| | Result [px] | Writing time [m:s:ms] | Mailbox time [m:s:ms] | Result [px] | Writing time [m:s:ms] | Mailbox time [m:s:ms] |
| Person 1 | 33 | 01:05:05 | 02:05:07 | 75 | 02:08:15 | 03:07:25 |
| Person 2 | 25 | 01:16:98 | 02:31:80 | 120 | 01:15:37 | 03:21:47 |
| Person 3 | 33 | 00:52:14 | 03:12:95 | 150 | 00:47:41 | 02:54:34 |
| Person 4 | 120 | 01:35:18 | 02:34:60 | 150 | 00:51:07 | 03:18:03 |
| Person 5 | 40 | 00:44:33 | 01:49:38 | 66,67 | 00:49:18 | 04:01:13 |
| Person 6 | 75 | 01:10:07 | 02:55:41 | 120 | 00:56:66 | 03:12:83 |
| Person 7 | 37,5 | 01:15:85 | 01:29:04 | 75 | 02:40:82 | 02:47:71 |
| Person 8 | 40 | 01:05:70 | 02:11:15 | 120 | 00:49:95 | 02:52:31 |
| Person 9 | 40 | 00:50:89 | 02:50:28 | 150 | 01:09:13 | 03:10:70 |
| Person10 | 33 | 01:02:56 | 01:46:17 | 200 | 00:40:21 | 03:21:24 |
| Average precision [px] | 47,65 | | | 122,67 | | |
| Average writing time [m:s:ms] | 00:59:86 | | | 01:12:79 | | |
| Average time to get to Onet.pl mail box [m:s:ms] | 02:18:59 | | | 03:12:70 | | |

**Table 4.** Comparison of the eyetracking solutions tested during the second experiment

| | Camera Pan Tilt + ljo_gazetracker | Camera Logitech + Opengazer |
|---|---|---|
| Precision | Very precise | Quite precise |
| Gaze-tracking | Very long and arduous setting of the camera: to get it directly on the eye, and set up pupil and cornea reflection | After selecting the feature points on the face and quick calibration it is ready |
| Calibration | Fast, user looks at nine points on the screen | Fast, it is possible to set up the custom number and coordinates of the calibration points. |
| Losing of the tracking | It does not loose image of the eye, small head movement does not make any problems. | Often loses points on the face even during little head movement and light changes. |
| Other remarks | After it is calibrated correctly once, other people can start working immediately | Calibration needs to be repeated few times for better results. Losing of gaze tracking is very burdensome |

## 5 Summary

After the experiments we gathered all results and observations, and analyzed them. Comparing the time and the amount of mistakes made by users in the first experiment, we can see that the fastest and most reliable application for gaze-writing is Dasher. GazeTalk5 was the slowest and had the most mistakes, mostly due to the "Midas touch" problem [6]. In the opinion of the respondents, Dasher was also the best application. Firstly writing in it was hard to manage, but after a while, it was easy to write in it for everyone.

Using the FlashKeyboard application, we were able to compare selecting letters by using space key and by moving cursor over the target and waiting for two seconds. The result was that using a key is much faster. Still, in case of people with motor disabilities usually they can only use applications which use dwell time for selection. Using the key could be used by people who have the ability to move at least one finger. The keyboard key could be replaced by a button.

Depending on needs and financial predispositions we can present some eyetracking recommendations. In case we want to control the pointing device with large precision and speed, we should get a professional eyetracking camera worth tens of thousands of Euro. There are many professional solutions available, like for example Tobii products. Still such solutions are recommended for companies that use eyetracking for professional purposes, that need its equipment to work fast and reliably. Therefore for handicapped and casual users it is recommended to use freeware solutions and webcam. The precision of such solution is not high, but as the research showed, using such solution we can write in Dasher as fast as with expensive professional camera. We should suspect, that using more advanced camera than Logitech Quick Cam 9000 Pro with, for example infrared LEDs could produce better results. In conclusion of experiments conducted in this work, we can say, that the best solution for handicapped or casual user is usage of combination Opengazer plus GazeTalk5. Opengazer is the best freeware software for eyetracking with ordinary webcam. GazeTalk5 enables complex control of PC with eyetracking. Using it we can browse the Internet, listen to the music etc. Moreover, it allows the user to run Dasher directly from the main menu. Thanks to that we can write with GazeTalk5 using the fastest available gaze-writing applications and be able to do many other things besides writing.

Comparing the different methods of gazetracking, clearly the best one is that used in Pan Tilt camera. It obtains simultaneously the position of pupil and cornea reflection, and traces the line of gaze on the screen very precisely. In case of freeware solutions we have few different approaches. After the experiments we can say that the best of them is the solution used in Opengazer. It captures the key points on users face and extracts them from video image. This solution enables fast calibration and descent precision. There are some different solutions, that use for example cameras with night vision, or like myEye infrared light, but they have no appliance with ordinary web cameras.

## References

[1] Mohamed A.O., Perreira 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 (March 12, 2010)

[2] Applied Science Group, Eye Tracking System Instructions. ASL Eye-Trac 6000, Pan Tilt/Optics (2006)

[3] Duchowski, A.T.: A Breadth-First Survey of Eye Tracking Applications, http://brm.psychonomic-journals.org/content/34/4/455.full.pdf (March13,2010)

[4] Duchowski, A.T.: Eye tracking methodology: Theory and practice, pp. 205–300. Springer, London (2003)

[5] Information about Dasher project, http://www.inference.phy.cam.ac.uk/dasher/DasherSummary2.html

[6] Jacob, R.: Eye Tracking in Advanced Interface Design, Washington D.C., Human-Computer Interaction Lab Naval Research Laboratory, http://www.cs.tufts.edu/~jacob/papers/barfield.pdf (March12, 2010)

[7] Sesin, A., Adjouadi, M., Cabrerizo, M., Ayala, M., Barreto, A.: Adaptive eyegaze tracking using neural-network-based user profiles to assist people with motor disability. Journal of Rehabilitation Research & Development 45(6), 801–818 (2008)

[8] Zhai, S., Morimoto, C.H., Ihde, S.: Manual and gaze input cascaded pointing, ACM SIGHCI-Human Factors Comput. Syst. Conference, http://www.almaden.ibm.com/u/zhai/papers/magic/magic.pdf (March 19, 2010)

[9] Zhang, X., MacKenzie, S.: Evaluating Eye Tracking with ISO 9241 - Part 9, Department of Computer Science and Engineering York University, Toronto, http://www.yorku.ca/mack/45520779.pdf (March 12, 2010)