

# Low cost eye tracking for human-machine interfacing

W Lemahieu, B Wyns

Department of Electrical Energy, Systems and Automation, Faculty of Engineering, Ghent University,  
Technologiepark 913, Ghent, BELGIUM

[wouter.lemahieu@ugent.be](mailto:wouter.lemahieu@ugent.be)

<http://www.eesa.ugent.be>

## ABSTRACT

One of the main challenges for developers of new human-computer interfaces is to provide a more natural way of interacting with computer systems, avoiding excessive use of hand and finger movements. In this way, also a valuable alternative communication pathway is provided to people suffering from motor disabilities. This paper describes the construction of a low cost eye tracker using a fixed head setup. Therefore a webcam, laptop and an infrared lighting source were used together with a simple frame to fix the head of the user. Furthermore, detailed information on the various image processing techniques used for filtering the centre of the pupil and different methods to calculate the point of gaze are discussed. An overall accuracy of 1.5 degrees was obtained while keeping the hardware cost of the device below 100 euros.

## 1. INTRODUCTION

The use of computers and embedded controllers has increased dramatically in the past few decades. This growth is most noticeable in consumer electronics. Where the use of computer systems integrated in industrial environments is common practice since quite some time (for example in automotive industry, process industry), the use of such (embedded) systems in commercial applications is still rapidly evolving and continuously new products are brought to the consumer market (for example smart phones). One of the main challenges of these computer systems is the way they interact with the user. In an industrial environment trained technicians use mobile teach pendants (for example in robotic manipulator control (Biggs and MacDonald, 2003)), or advanced touch screen technology. In commercial applications, mouse and keyboard are still most often used for interacting with the computer system. Clearly more advanced ways of communicating with electronic devices are needed, not only because of space limitations such as in mobile (smart) phones but because humans tend to communicate and interact in a more intuitive and simplified way. For that reason many researchers have already developed new methods to communicate in a natural way with a computer. Well known examples of this tendency are voice controlled applications, touch screen technology, motion sensing remotes like the Wii-mote and even brain computer interfaces (BCI, Devlaminck et al., 2009). Another solution, which is by far the most natural one, is to register and track the point of gaze of a person. In this paper the design of a low cost eye tracker is described. With only a simple webcam and an infrared light source an apparatus was developed that allows moving a mouse cursor using pupil tracking.

This apparatus is not only very handy for healthy computer users. Also for people with severe motor disabilities it could be the only possible alternative to communicate with their environment. Diseases like amyotrophic lateral sclerosis (ALS), which consists out of a paralysis of the muscles, eyes are the only way to control the environment (Donegan et al., 2006). With the use of an eye tracker these people (partially) regain the capability of controlling their wheelchair or other elements of their environment.

Until now, a general problem with commercially available eye trackers was the unit cost. Prizes often only included the eye tracker itself. Adaptation to the user usually results in an increased cost of the total system. Lowering the cost of the eye trackers would equally result in an acceleration of the development of applications where the eye tracker can be used. It would also make the introduction of eye trackers in the everyday use of computers possible. For these reasons the main goal of our research was to develop a remote eye tracker containing only cheap off the shelf components. An important concern was to investigate if the accuracy that could be obtained with these fairly simple components would be good enough to allow users to interact with a computer. Therefore, various image processing techniques were

used for filtering the centre of the pupil. Different methods to calculate the point of gaze are discussed. The paper is organized as follows. Section 2 gives an overview of the literature on the subject of eye tracking. In sections 3, 4 and 5 the hardware setup, the image processing software and the gaze extraction algorithm are discussed. The results can be found in section 6. Finally, section 7 summarizes the conclusions of this work.

## 2. LITERATURE OVERVIEW

Eye tracking can be split up into two groups based on the location of the camera with respect to the user. The first group is formed by head mounted eye trackers. Here the needed equipment is mounted directly on the head of the user with some help of a helmet or eyeglasses. These eye trackers are most used in on-site applications (Franchak et al., 2010) but will not be discussed in this contribution.

The second, and largest, group of eye trackers, are remote eye trackers. This type of eye trackers doesn't require the need to wear additional hardware because the camera is mounted at some distance from the user. Remote eye trackers are generally used to interface with a computer. This kind of eye tracker shows great promises for becoming a widely accepted eye-gaze tracking interface.

When focussing at hardware setup in remote eye trackers, three different popular approaches can be found. A first approach is the use of a single regular camera in combination with a calculation technique for extracting the point of gaze (Baluja and Pomerleau, 1994; Stiefelhagen et al., 1997; Qun Xu et al., 1998; Sewel and Komogortsev, 2010). Firstly, the eye is detected and then a small image of the eye is taken (for example 30 x 20 pixels). This information is used as input to a neural network approach. This method limits the head movement to the viewing angle of the camera and needs an exhaustive calibration procedure to learn the neural network.

The second approach uses two cameras, one wide angle camera and one camera on a pan-tilt-zoom system. The eye is detected with the wide angle camera and with an infrared sensitive camera on the pan-tilt-zoom system a close-up of the eye is taken. This setup is combined with two or four infrared light groups which makes it simpler to detect the pupil (Yoo et al., 2002 and 2004; Pérez et al., 2003; Ohno et al., 2004). Using a pan-tilt-zoom unit allows the user to move his head, although within certain limitations due to the use of simple servos.

The third approach uses one fixed camera together with one or more infrared lighting groups (Ohno et al., 2002; Morimoto et al., 2002; Coutinho and Morimoto, 2006; Meyer et al., 2006; Hennessey et al., 2006). The head movement is again restricted to the viewing angle of the camera. By use of only one camera the cost and the dimensions of the eye tracker can be kept small. Ohno et al. (2002) achieved the best results with this setup and reached an accuracy of 1 degree.

Ohno et al. (2002), Coutinho and Morimoto (2006) concluded that when calculating the point of gaze, the use of multiple infrared reflections increases the algorithms' performance. Obtaining good accuracy with just one infrared reflection necessitates the use of a calculation model based on a 3D-model of the eye. Researchers using more infrared reflections can simplify their algorithm to a more simple transfer function (Yoo et al., 2002 and 2004). In this contribution we adopt the approach of Yoo et al. (2004) in combination with a single infrared lighting group and add extra calibration data to obtain good overall accuracy.

## 3. EYE TRACKER HARDWARE

The hardware concept was based on a study of Hansen et al. (2004). The authors theoretically discuss some aspects that have to be taken into account to reduce the cost of an eye tracker. They claim that one of the biggest difficulties but also the biggest advantage of eye trackers made with off the shelf components is the exchangeability of different cameras in different lighting conditions. When programmers succeed in making a user friendly eye-tracker (software) capable of working with different camera's (hardware) the distribution of eye trackers could be much cheaper. By only downloading the software and connecting the computer to a commercial webcam (already built-in in many laptops) the user could already have a working eye tracker.

The developed eye tracker is a remote eye tracker as shown in figure 2.1. The user is seated at 40-50 cm from the computer screen. The position of the head is fixed by the use of a simple headrest providing support for the chin and forehead. The frame is built using wooden beams separated using long bolts and nuts and clipped onto the desk where the computer is placed. The camera and infrared lighting groups

are located near the screen of the computer. The exact location of the camera and lightning isn't important but to achieve the largest flexibility the position under the centre of the screen is optimal. With this alignment the user is hardly restricted in his movements (of course the design remains remote) and the largest possible comfort is offered.

The used camera was a cheap USB-webcam with a resolution of 1024×768 pixels at a frame rate of 25 frames per second. The output of the camera is an RGB-valued image. The infrared filter and lens were modified on this webcam. Firstly, the infrared filter was removed to allow the use of infrared lighting to increase the contrast between the pupil and the rest of the eye as described in (Yoo et al., 2002; Pérez et al., 2003; Yoo et al., 2004). Additionally, the position of the reflection of this infrared light is used to determine the point of gaze of the user. A detailed explanation of this calculation method is given in section 5.

Secondly, the original lens of the webcam with a focal length of 3.58 mm was replaced by a lens with a focal length of 8 mm. These lenses, originally intended for cameras, contain a M12 screw-thread which makes sure they can be replaced easily. The focal length of 8mm allows to obtain a more accurate image although the use of a cheap off the shelf camera.



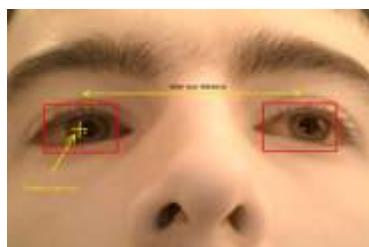
**Figure 1.** (a) Overview of the hardware design. Note that the camera and the infrared light source are placed near the computer screen. (b) A picture of the real setup.

## 4. IMAGE PROCESSING ALGORITHM

To calculate the point of gaze two features have to be extracted from the image. Firstly, the exact position of the pupil is detected. Secondly, the same procedure is repeated to extract the position of the reflection of the infrared light. With these two features the point of gaze of the user is calculated as described in section 5.

### 4.1 Eye detection

Due to the use of the head fixation the position of the eye in the image never changes (or very slightly). Before calibration begins, the user is asked to position the camera such that the subject's right eye is positioned within the red rectangle shown on the computer screen as shown in figure 2 (left half of the image). Next, a rectangular area of 200x150 pixels around this position is taken. The surface of this rectangle is large enough to ensure capturing the eye even if the subject moves a little during the experiment. The image of the second (left) eye is obtained by drawing a second rectangle at the same height as the first image but with a horizontal displacement. This displacement is equal to the inter eye distance and depends on depends on the used lens. The exact distance value was empirically measured. This seems to be a subject depending parameter, however, the inter subject difference has little or almost no influence on the point of gaze because the rectangles capturing both eyes are large enough.



**Figure 2.** *Determination of the position of the eye using a camera with a focal length of 8 mm. Due to the use of a head fixation the eye can be found on the same spot in every image*

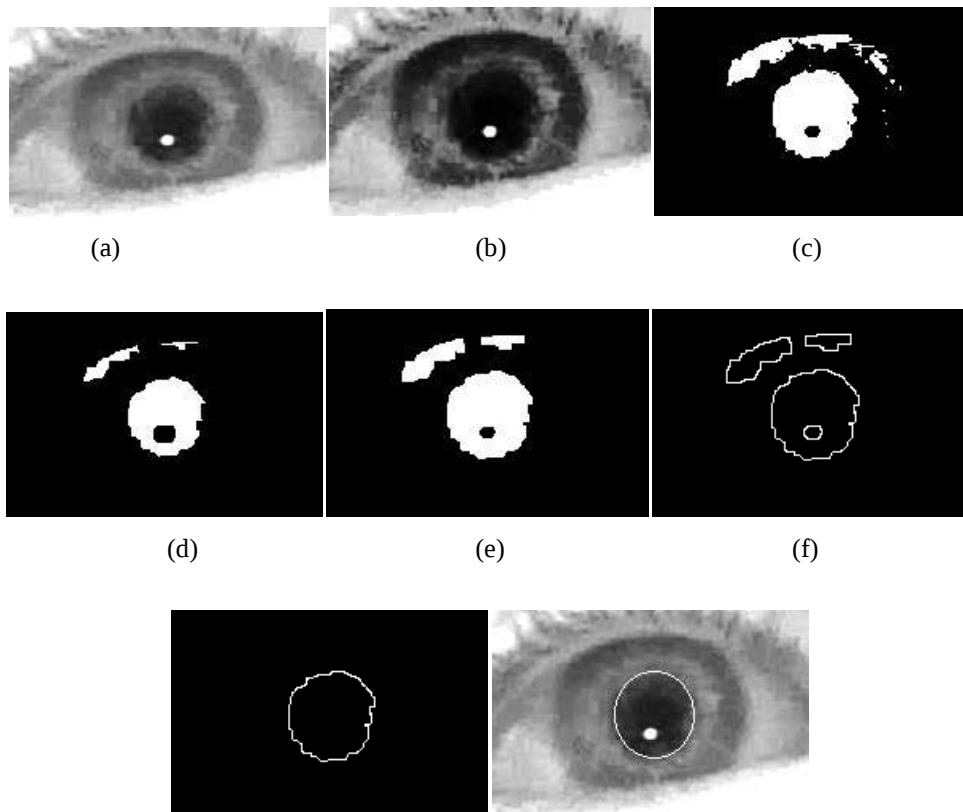
#### Glint detection algorithm

The position of the reflection of the infrared light source (glint) is used for calculating the point of gaze. The detection of this glint is quite straightforward as the glint is the brightest spot in the eye. After taking a threshold function of the red channel of the source image the glint is found. Several tests in different lighting conditions showed that the contrast between the glint and the rest of the image was the highest in the red spectrum. The level of this threshold was set quite high at 220 (scale from 0 to 255) to isolate the reflection as accurately as possible, avoiding noisy pixels. Empirical tests proved that the exact value of this threshold wasn't very influential as long as the value was quite high (above 200). The ellipse fitting function provided by the OpenCV library is used to determine the centre of the glint

#### Pupil detection algorithm

Extracting the centre of the pupil from an image is more difficult and is done in seven steps. In the first step the blue and the green color channels are separated from the source image (figure 3a). Again, based on the authors observations in different lighting conditions, the contrast between the pupil and the rest of the eye is higher in green and blue color channels than in the red color channel. The second step applies histogram equalization to reduce the influence of lighting conditions (figure 3b). With a simple threshold function in the third step, the image is converted to a binary image (figure 3c). To reduce the influence of noise dilate and erode conversions are applied (figure 3d and 3e). In the fifth step the edge points of the pupil are determined as the points that surround the white regions in the binary image (figure 3f). As can be seen three loops are defined by connecting these points. Next, the loop corresponding to the pupil is selected based on the length of the loop. On forehand a set of loop dimensions for several subjects were measured, and with this data loops representing the glint or eyebrows can easily be eliminated because these loops are clearly smaller or shorter than the pupil (figure 3g).

In the last step, the calculation of the centre of the pupil is done by representing the pupil by an ellipse. For the calculation of this ellipse two methods were investigated. A method based on the RANSAC method (Fisher et al., 1981) and a method based on the least squares ellipse fitting algorithm of the OpenCV image processing library.



(g)

(h)

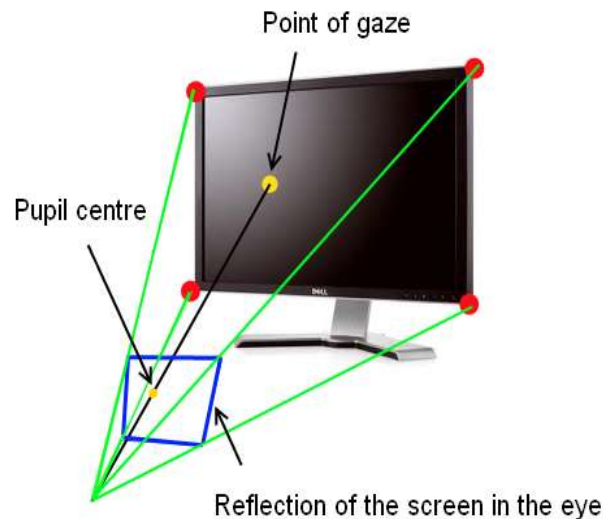
**Figure 3.** An overview of the pupil detection algorithm. The original image (a) is transformed with a histogram equalisation (b). The image is converted with a threshold function to a binary image (c). To reduce the noise in the image dilate (d) and erode (e) functions are adapted to this image. With an edge detection function the edge is calculated (f) and based on the contour of the resulting shapes the pupil is selected (g). The pupil is then calculated using an ellipse fitting procedure (h).

To increase robustness of pupil detection two additional filters were used: an eyebrow filter and a glint filter. Please note that the position of the glint was already determined as described in section 4.1. Here the glint is considered as noise. The eyebrow filter removes the upper (eyebrow) and lower (edge of the eye) strip of the founded edge points. Therefore, first the lowest and highest point of the pupil is searched. Afterwards every point that lies in the upper 10% or lower 10% of the pupil is removed from the dataset. The glint filter removes the points of the pupil edge that are within the near proximity of the glint. Thus for every edge point the distance between this point and the centre of the glint is calculated. When this distance between the pupil centre and the glint centre smaller is than two times the radius of the glint, the point is removed of the pupil edge.

As result of this image filtering process the x and y coordinates of the pupil in the image and the glint are obtained. This is sufficient to calculate the point of gaze as discussed in the next section.

#### CALCULATION OF THE POINT OF GAZE

As described in section 2, literature reveals many methods using multiple infrared lighting sources. The main benefit of this approach is that a method with two (or more) infrared lighting sources avoids the need of a calibration procedure. As shown in figure 4, when we assume that the surface of the eye is nearly flat, and we place different infrared lighting groups at the corners of the computer screen, then no calibration is needed to determine the direction of gaze, even no head fixation is needed (Yoo et al., 2004). With the fact that the computer screen and the camera are placed horizontally, only infrared lighting groups are needed at two diagonal corners of the screen.

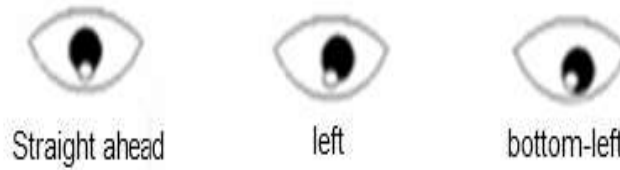


**Figure 4.** Image showing the determination of the point of gaze without a calibration action with four infrared lighting groups

But using multiple infrared lighting sources also suffers from several drawbacks. Firstly, when using more reflections, the chance that the reflection occurs near the border of the pupil increases. And therefore it becomes more difficult to detect the pupil in an accurate way. Secondly, when using a single lighting group, this group can be placed at the bottom of the screen. When sitting too close to the screen the angle of the illumination becomes too high and the reflection appears at the outside of the iris. It is already known that in that case the calculation is incorrect (Pérez et al., 2003). So using more infrared reflections necessitates placing the user further away from the computer screen.

For a more robust eye tracker we chose to concentrate on making an eye tracker with a single infrared lighting source in combination with a calibration procedure. Due to the geometric differences between the eyes of different users a calibration procedure is common when using eye trackers. During such a procedure the user is asked to look a predefined number of points on the screen. This number can vary from two to up to more than 5 depending on the calculation method that is used. The data (x, y coordinates) obtained during this calibration trial is then used to calculate the point of gaze. This is formulated as an equation with some parameters depending on the chosen calculation method. The parameters are usually calculated using a least square minimization problem.

With the position of the pupil and the reflection known, the point of gaze can now be calculated as illustrated in figure 5. These three images of an eye are taken by a camera mounted at the same fixed position. Each eye is lit with an infrared light from the same spot. Now it is easy to see that the position of the pupil compared to the position of the reflection, the pupil-glnt vector, can be used to guess the viewing direction.



**Figure 5.** These three images show the correlation between the viewing direction and the position of the pupil in reference to the infrared reflection.

With the hardware setup from figure 1 (right) four different calculation methods were implemented to transform the pupil glint vector of the right eye to the point of gaze. Unless stated otherwise, all methods use data from a single eye (left eye). All methods were compared to each other. The first two methods (figure 6.a and 6.b) consist out of scaling the obtained pupil-glnt vector. These scale factors were calculated a priori with a calibration procedure with two calibration points for the first method and three calibration points for the second method. Equations 1 and 2 represent the transformation of the coordinates of the pupil glint vector  $K$  to the point of regard  $\bar{K}$ . The parameters  $\alpha_1$ ,  $\alpha_2$ ,  $\alpha_3$  and  $\alpha_4$  are the scaling parameters calculated with the calibration data. Equations 3 and 4 represent the second method in a similar way. Due to the use of an extra calibration point compared to the first method, two extra parameters are used to calculate the point of regard.

$$\bar{K}_x = \alpha_1 K_x + \alpha_2 \quad (1)$$

$$\bar{K}_y = \alpha_3 K_y + \alpha_4 \quad (2)$$

$$\bar{K}_x = \alpha_1 K_x + \alpha_2 K_y + \alpha_3 \quad (3)$$

$$\bar{K}_y = \alpha_4 K_x + \alpha_5 K_y + \alpha_6 \quad (4)$$

A third method (figure 6.c and eq. 5-8), which uses four calibration points, was based on the cross ratio of a line. This method was previously described in (Yoo et al., 2002), but the authors used their method to calculate the point of regard with four infrared reflections without a calibration requirement. In our implementation we use only one infrared reflection but four calibration points instead to calculate the point of gaze.

$$CR_x = \frac{(A_x R_y - R_x A_y)(S_x B_y - B_x S_y)}{(A_x S_y - S_x A_y)(R_x B_y - B_x R_y)} \quad (5)$$

$$CR_y = \frac{(B_x T_y - T_x B_y)(U_x C_y - C_x U_y)}{(B_x U_y - U_x B_y)(C_x T_y - T_x C_y)} \quad (6)$$

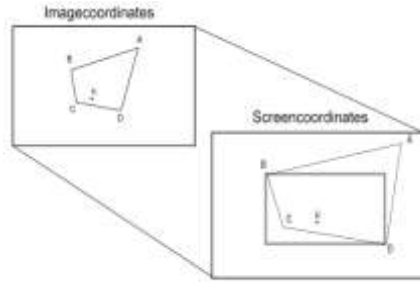
$$\ddot{K}_x = \frac{\text{Screen height} \cdot CR_x}{1 + CR_x} \quad (7)$$

$$\ddot{K}_y = \frac{\text{Screen width} \cdot CR_y}{1 + CR_y} \quad (8)$$

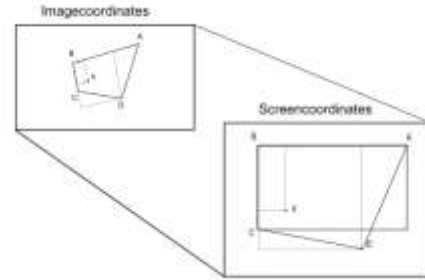
The last method (figure 6.d and equations 9 and 10) was based of the projective transformation and uses also four calibration points. The theory of the projective transformation (Rothwell et al., 2005) wasn't used yet in other researches about eye tracking. It uses geometric algebra to transform the pupil-glint vector into the point of regard on the screen. Thereby we assume that the four calibration points are located in the same plane (or the eye has an almost flat surface). With these four points a so-called vanishing point can be calculated that is the link between the image coordinate system and the screen coordinate system

$$\rho \begin{bmatrix} \ddot{K}_x \\ \ddot{K}_y \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \alpha_4 & \alpha_5 & \alpha_6 \\ \alpha_7 & \alpha_8 & \alpha_9 \end{bmatrix} \begin{bmatrix} K_x \\ K_y \\ 1 \end{bmatrix} \quad (9)$$

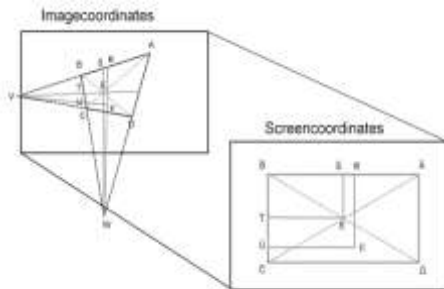
$$\begin{aligned} \alpha_1 K_x + \alpha_2 K_x + \alpha_3 - \alpha_7 K_x \ddot{K}_x - \alpha_8 K_x \ddot{K}_x &= \ddot{K}_x \\ \alpha_4 K_y + \alpha_5 K_y + \alpha_6 - \alpha_7 K_y \ddot{K}_y - \alpha_8 K_y \ddot{K}_y &= \ddot{K}_y \end{aligned} \quad (10)$$



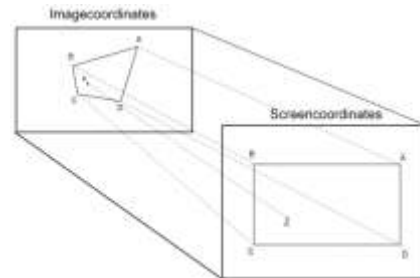
(a)



(b)



(c)

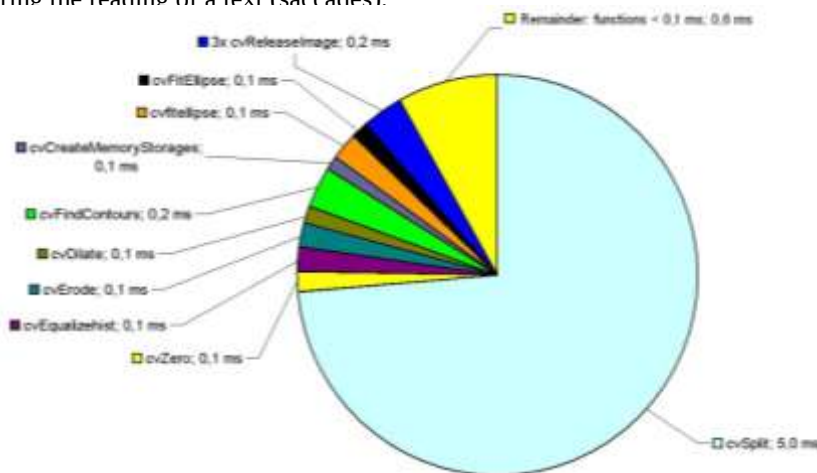


(d)

**Figure 6.** The visual presentation of the four methods compared in this survey. Methods (a) and (b) are based on simple scaling functions. Method (c) was based on the cross ratio method (Yoo et al., 2002), while the last method (d) was based on the projective transformation (Rothwell et al., 2005).

## 5. RESULTS

The image processing software was implemented in the C++ language with the OpenCV image processing library. Timing analysis of the image detection software on a 1,83 GHz Intel Duo Core processor, 1GB RAM laptop with Windows XP showed that reaching a frame rate of 25 frames per second (1 frame every 40ms) was possible (figure 7). This allows controlling the computer quite accurately although this speed is not sufficient for measuring high speed eye movements that occur for instance during the reading of a text (saccades).



**Figure 7.** The total time consumption of the eye tracker was analysed. The point of gaze calculation algorithms were much faster ( $<0,1$  ms) than the image processing algorithms and were therefore not included in this figure. In the graph we see the distribution the time between the OpenCV functions for the detection of the pupil and the glint in one eye. CvSplit witch divides the whole source image into three color channels takes the most time because of size of the image and the time consuming memory operations. Other functions take far less time because they handle the reduced image (rectangles as described in section 3).

To evaluate the performance of the different methods a group of 15 subjects were asked to perform a simple task, i.e. looking at 14 predefined spots on the computer screen (fig. 9, predefined spots are denote by encircled x symbols). Afterwards the difference in degrees between the calculated viewing angle and the correct viewing angle was computed. The results when using measurements from a single eye are shown in Table 1.

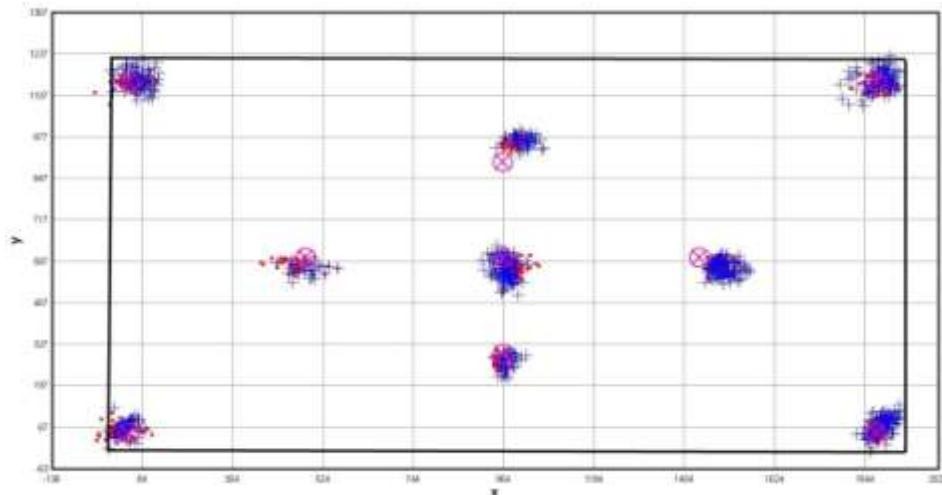
**Table 1.** Accuracy of the different methods.

| Method | Accuracy (in degrees)<br>(standard deviation) | Accuracy (in cm with 40 cm<br>between user and screen) |
|--------|-----------------------------------------------|--------------------------------------------------------|
| 1      | 2.3 (0.83)                                    | 1.61                                                   |
| 2      | 2.4 (0.65)                                    | 1.68                                                   |
| 3      | 2.4 (0.75)                                    | 1.68                                                   |
| 4      | 2.0 (0.73)                                    | 1.40                                                   |

The method based on the projective transformation clearly gives the best results.

By using both eyes the accuracy of the eye tracker improved to 1.5 degrees. The combination of the data of both eyes is done by first calculating the point of gaze for both eyes separately. Afterwards the average of these two points is taken. If accurate detection of the pupil or glint in one of the eyes was not possible (due to noise, or missing of a glint), the prediction of the other eye was taken.

Figure 8 clearly shows that this accuracy of 1.5 degrees is sufficient to determine where the person is looking on the screen. This graph was measured with a distance of 40 cm between the user and the screen.



**Figure 8.** Results of the calculation of the point of gaze. The circles with a cross are the targets that have been looked to. Red points are the calculated points of regard for the right eye, blue points for the left eye. The black rectangle depicts the contour of the screen.

The points close to the calibration points result in better accuracy. It is noted that large deviations (bad performance) can occur when one of the calibration points is poorly measured. The cause of this can be a user that is unfocused or inaccurate pupil detection due to large changes in room illumination (see also section 5). Therefore it is important to take extra care when performing calibration procedure.

## 6. USABILITY TESTS

In the introduction we stated that the reason for designing an eye tracker was to assist people suffering from severe motor disabilities in their everyday life. With the use of an eye tracker these people (partially) regain the capability of communicating with others or even controlling their wheelchair or other elements of their environment. The required accuracy to do so is below what is achieved by state-of-the-art commercial eye trackers.

To demonstrate that the developed eye tracker's accuracy is good enough to be used as an alternative human computer interface, another experiment was done. For that experiment an interface was written to communicate with the existing typing software package Dasher (Ward and Blackwell, 2000). This interface was implemented by simply linking the coordinate of the mouse cursor to the calculated point of gaze.

With Dasher a word is typed by successively selecting the different characters of the word. When looking at a character, Dasher automatically zooms in on the region of interest. Hereby, the influence of the limited accuracy of the eye tracker (see Table 1) is reduced. After zooming in on a single character a new series of characters appear on the right side of the computer screen. Remark that no clicking is acquired to use Dasher; starting and stopping the program can be done by looking at the centre of the screen for three seconds. Dasher also includes some predictive spelling using a dictionary as well. Dasher automatically displays different words based on characters that were already selected by the user (Figure 9). Using such probability calculations, variable character size and the use of a colour schedule typing speeds as fast as keyboard input could be obtained.

After getting acquainted to the Dasher program, several users were capable of typing on average 40 characters per minute, which is a very good result. Optimal typing speeds could only be obtained after that the user gained some experience with the system. It is recommended to introduce new users to the eyetracker and the Dasher software separately. When the test subjects did not experiment with the two systems separate it took much more time to obtain proper typing speeds than when the users were granted some time to experiment with the control of the mouse with an eyetracker and the use of the Dasher software.



**Figure 9.** Example of the usage of the Dasher software for typing the word “Test”. In this example the user already selected the character T, and aims toward the letters e and s. The second figure is one instance later when the program zoomed into the region of interest and the character e is already selected. The software varies the size of the boxes of the characters based on the probability that a character can occur. For this example the probability of the character “s” after the characters “Te” is much higher than the other combinations which results in a high typing speed.

## 5. CONCLUSIONS

The goal was to provide a low cost alternative for the expensive commercially available eye trackers. In this contribution we concentrated on the hardware design, the image processing algorithm and the calculation algorithms for the point of regard. The careful choice of all of these elements resulted into a low overall cost of the hardware of the eye tracker. The cost of all of these components is kept well below 100 Euros. This proves that for making a usable eye tracker no expensive camera is needed.

We succeeded in achieving an accuracy of 1.5 degrees. Compared to other non-commercial eye trackers this is a good result. An experiment with a text-input software learned that typing speeds of 40 characters per minute were capable in ideal circumstances. This outcome gives prospects to making eye trackers that are affordable for everybody.

Future work will focus on using a 3D model of the eye to further enhance performance. In the setup described above the gaze point calculation models assume that the eye is a flat surface, while in reality this is not the case. Prediction accuracy can be increased in points situated away from the calibration points. Another important aspect is to avoid the necessity of a headrest. By adjusting the calculation algorithm and adding a pan-tilt-unit a head fixed setup could be eliminated without very high costs.

## 6. REFERENCES

- Baluja, S., & Pomerleau, D. (1994) *Non-intrusive gaze tracking using artificial neural networks*. Carnegie Mellon University.
- Biggs G., & MacDonald, B (2003). A Survey of Robot Programming Systems. *Proceedings of the Australasian Conference on Robotics and Automation (CSIRO)*, Brisbane
- Coutinho, F., & Morimoto, C. (2006). Free head motion eye gaze tracking using a single camera and multiple light sources. *Computer Graphics and Image Processing 19th Brazilian Symposium (SIBGRAP '06)*, Manaus, 171-178.

- Devlaminck, D., Wyns, B., Boullart, L., Santens P., & Otte, G. (2009). Brain-Computer Interfaces: from theory to practice. *Proceedings of the European symposium on artificial neural networks*. Evere, 415-424.
- Donegan, M., Oosthuizen, L., Bates, R., Istance, H., Holmqvist, E., Lundalv, M., Buchholz M., & Signorile I. (2006). D3.3 report of user trials and usability studies. *Communication by Gaze Interaction (COGAIN)*.
- Fisher, M. & Bolles, R. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24, 381-395.
- Franchak, J., Kretch, K., Soska, K., Babcock J., & Adolph, K. (2010). Head-mounted eye-tracking of infants natural interactions: A new method. *Proceedings of the 2010 Symposium on Eye Tracking Research & Applications*. Austin, TX, USA: ACM
- Hansen, D., MacKay, D., Hansen J., & Nielsen M. (2004). Eye tracking off the shelf. *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications (ETRA '04)*, Austin, 58.
- Hennessey, C., Nouredin, B., & Lawrence, P. (2006). A single camera eye-gaze tracking system with free head motion. *Proceedings of the 2006 symposium on Eye tracking research & applications (ETRA '06)*, Austin, 87-94.
- Meyer, A., Böhme, M., Martinetz, T., & Barth, E. (2006). A single-camera remote eye tracker. *Perception and Interactive Technologies. Artificial Intelligence*, 4021, 208-211.
- Morimoto, C. H., Amir, A., & Flickner, M. (2002). Detecting Eye Position and Gaze from a Single Camera and 2 Light Sources. *Proceedings of the International Conference on Pattern Recognition*, Quebec, 4, 314-317.
- Morimoto, C., & Mimica, M. (2005). Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding*, 98(1), 4-24,
- Ohno, T., Mukawa, N., & Yoshikawa, A. (2002). Freegaze: a gaze tracking system for everyday gaze interaction, *Proceedings of the 2002 symposium on Eyetracking research & applications (ETRA '02)* New Orleans, 125-132.
- Ohno T., & Mukawa, N. (2004). A free-head, simple calibration, gaze tracking system that enables gaze-based interaction, *Proceedings of the 2004 symposium on Eye tracking research & applications (ETRA '04)*, Austin, 115-122.
- Pérez, A., Córdoba, M., García, A., Méndez, R., Muñoz, M., Pedraza J., & Sánchez, F. (2003). A precise eye-gaze detection and tracking system. *Proceedings of the 11th International Conference in Central Europe of Computer Graphics, Visualization and Computer Vision*, Plzen – Bory, 105-108.
- Qun Xu, L., Machin, D., Sheppard, P., Heath, M., & Re, I. (1998). A novel approach to realtime non-intrusive gaze finding. *Proceedings of the British Machine Vision Conference*, Southampton, 428-437.
- Rothwell, C., Zisserman, A., Forsyth, D., & Mundy, J. (1995). Planar object recognition using projective shape representation. *International Journal of Computer Vision*, 16(1), 57-99.
- Sewell, W., & Komogortsev, O. (2010). Real-time eye gaze tracking with an unmodified commodity webcam employing a neural network. *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems (CHIEA '10)*, NY, 3739-3744.
- Stiefelhagen, R., Yang, J., & Waibel, A. (1997). Tracking eyes and monitoring eye gaze. *Proceedings of the Workshop on Perceptual User Interfaces*, San Francisco, 98-100.
- Ward D., & Blackwell, A. (2000). Dasher - a data entry interface using continuous gestures and language models. *Proceedings of the 13th annual ACM symposium on user interface software and technology (UIST '00)*, NY, 129-137
- Yoo, D., Kim, J., Lee, B., & Chung, M. (2002). Non-contact eye gaze tracking system by mapping of corneal reflections.. *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, Washington, 101.

Yoo D., & Chung, M. (2004). Non-intrusive eye gaze estimation without knowledge of eye pose, Automatic Face and Gesture Recognition. Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG'04), Seoul, 785.