The human eye as Human-Machine interface

Fernando Ribeiro · Merylin Santos · Gil Lopes

fernando@dei.uminho.pt

Abstract — Eye tracking as an interface to operate a computer is under research for a while and new systems are still being developed nowadays that provide some encouragement to those bound to illnesses that incapacitates them to use any other form of interaction with a computer. Although using computer vision processing and a camera, these systems are usually based on head mount technology being considered a contact type system. This paper describes the implementation of a human-computer interface based on a fully non-contact eye tracking vision system in order to allow people with tetraplegia to interface with a computer. As an assistive technology, a graphical user interface with special features was developed including a virtual keyboard to allow user communication, fast access to pre-stored phrases and multimedia and even internet browsing. This system was developed with the focus on low cost, user friendly functionality and user independency and autonomy.

Index Terms—Eye tracking, human machine interface, assistive technologies.

I. Introduction

The difficulties encountered by people with physical disabilities can be minimized through a human machine interface, where the interaction with a machine could be advantageous in everyday tasks. Especially for people who cannot speak, the communication through written conversation is very important for mental health and social inclusion reasons.

The system described in this paper consists of a non-contact eye tracking device that makes the interface between a user and a computer, by reading the movement of one eye and using it to move and position the computer cursor in a graphical interface based window. The system was developed for people with physical disabilities that only (or at least) have full control of one eye.

The system has also a graphical user interface dedicated to people with paralysis that are not able to talk. System features can be easily added or removed in the future.

Much work has been performed to develop systems that use the eye for human-machine interface [1-3]. Non-invasive systems are also being developed by authors in order to avoid physical contact with the user [4, 5]. This prevents skin problems when prolonged used of any sort of apparatus is attached to the user head. This is more important when a head attached system is used by an upper limbs disabled person that cannot remove the device on its own whenever it feels uncomfortable.

There are four known techniques for eye movement measurement, involving the use or measurement of: Electro-OculoGraphy (EOG), scleral contact lens/search coil, Photo-OculoGraphy (POG) or Video-OculoGraphy (VOG), and video-based combined pupil and corneal reflection [6]. A video-based technique is divided in dark pupil and bright pupil detection [7]. This technique is more suitable for use in this system since it needs to be low cost and open-framework. There are various eye tracker systems developed using this technique for human-machine interaction [8, 9] and some that have the same purpose and provide a virtual keyboard [10, 11].

This paper initially presents the methodology used by the authors in the developed system, followed by the algorithms that were implemented and a brief description of the graphical user interface that was created in order to be easily operated by any user.

II. Methodology

Even though there are various methods for reading eye movement, in this work the eye tracking is based on video image analysis, more specifically infrared (IR) video image reflection to enhance the contrast between the pupil and the iris. One advantage of using IR light is due to being invisible to the human eye and therefore it will not disturb the user. Another advantage of IR light is its resilience or stability to variations in room illumination.

A Sony *PlayStation® Eye* camera was used for the video capture. Since this camera does not originally support the capture of IR light, modifications were required. The infrared blocking filter was physically removed and a visible light filter was added. The lens used is a 50 mm focal distance with S-mount and M12 thread.

An IR source that generates bright corneal reflections when illuminated is also used. It is based on a commercial circular array of 12 IR LEDs (JG-22) consuming a total power of 3 W. The position of the IR source was essential to attain pupil detection. If the light source is near the optical axis of the camera, the reflection of IR rays occur towards the camera, resulting on a bright pupil. If the light source is away from the optical axis of the camera, the IR reflection rays do not occur towards the camera, resulting in a dark pupil [1]. Another important aspect of this technique is the invariance to room light changes especially at night. Camera, lens and the LED array can be seen in **Fig. 1**.

Fig. 1. Camera, lens and LED array used in this work

Although the graphical user interface was defined and developed without any major alteration, different methods were tried on the vision processing algorithms to interface with the system in order to attain the best results for a stable operation, as described in the following chapters. All the artificial vision programming described in this paper was based on the *OpenCV* library methods [12].

III. Hough transform Method

This first method was based on the glint generated by the IR light and by the use of the Hough transform for the pupil detection.

A. Pupil and glint detection

In this method, the pupil is detected using the dark pupil method, since the bright pupil does not offer a contrast as high as desirable (

Fig. 2).

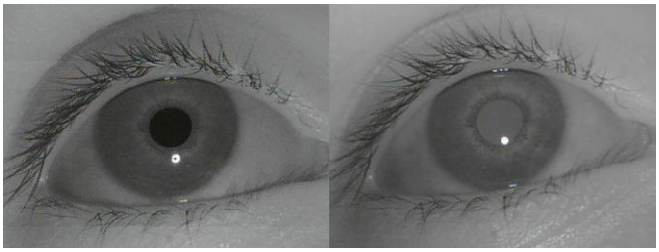

Fig. 2. Dark pupil (left) and bright pupil (right) methods

For the pupil center detection the Hough transform is used for circle detection. At first, the image contour is detected so that the transform is applied correctly. The contour is detected by abrupt changes in brightness or by discontinuities. For this purpose, a Canny operator is used.

In the next step a voting procedure is carried out. In the canny image, each time a white pixel is found, a vote is held on an array of votes. The votes are held around the coordinates of the white pixel at a distance *r* (radius). In

Fig. 3 two images are shown; on the left side a resulting canny image and on the right side the array of votes.

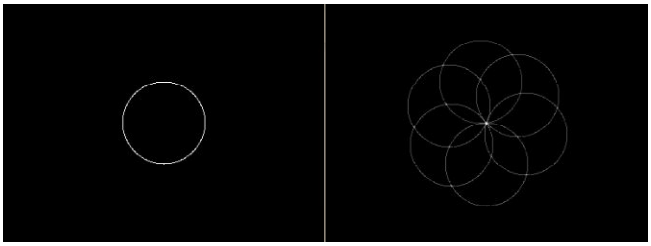

Fig. 3. Canny image (left) and array of votes (right)

A maximum on the array of votes will correspond to the center of the circle of radius *r*. If the radius of the circle is unknown, then an absolute maximum will not be found. This is due to the fact that the radius used in the voting procedure is not equal to the radius of the circle to detect. Therefore it is important to know the radius of the circle.

Since the pupil does not have a constant radius, the solution to this problem lies in the analysis of multiple arrays of votes, for different radii of the pupil. The votes are made to 12 radii, from 18 pixels up to 30 pixels. In this way it is possible to detect the pupil center.

The glint is detected by the same method, but there is no need to make 12 different arrays of votes since the glint radius is constant.

B. Data collection and treatment

At first, only relative pupil center coordinates were collected (static reference point with coordinates (0, 0)). The first tests showed that even the breathing movements of the user change the reference point, disabling the calibration carried out and preventing the normal functioning of the system. It was then decided to use a moving reference position, the glint.

At this point, the data collected is the pupil center in relation to the glint, as seen in

Fig. 4, which presents values not influenced by small head movements.

Understanding that the user has motor disabilities, it can be assumed that the distance between the user and the camera remains constant. If this was not true, it would be another parameter to take into account.

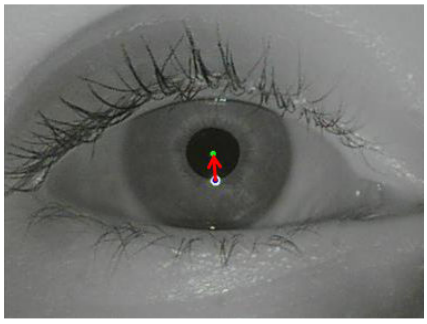

Fig. 4. Data vector

C. *Data filter*

The data obtained cannot be used directly, since the constant change in the pupil radius can also change the pupil center, which despite being small is aggravated due to the low resolution image.

The *Kalman* filter is an optimal estimator that predicts and corrects the estimated state of a system and consists on the application of two groups of equations. The first group, called *time update*, is responsible for the advance in prediction time. The second group, called *measurement update*, incorporates a new measurement in the estimated state in advance to correct a *posteriori* estimation.

The implementation of this filter was also carried out using the *OpenCV* library by changing the values of the noise covariance of the process, Q, and the covariance matrix of measurement noise, R.

Fig. 5, **Fig. 6** and **Fig. 7** shows the different results for different values of R and Q. The values chosen to be used in this system were $Q = 1x10^{-3}$ and $R = 0.1$. On each graph the blue line represents the original values and the red line represents the Kalman filter output.

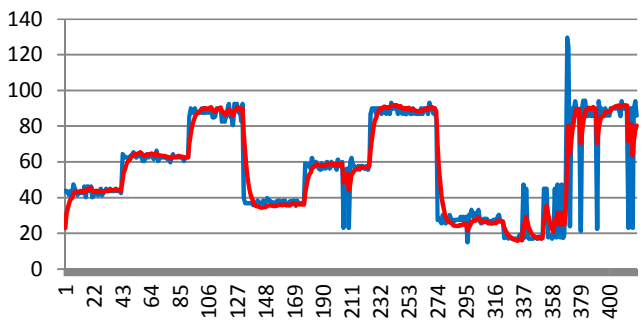

Fig. 5. Filter response (red), Q = $1x10^{-2}$ e R = 0,1

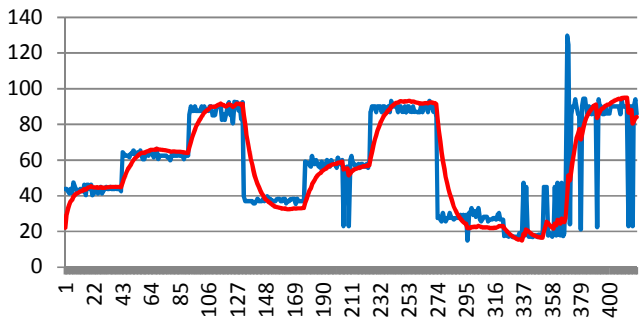

Fig. 6. Filter response (red), Q = $1x10^{-3}$ e R = 0,1

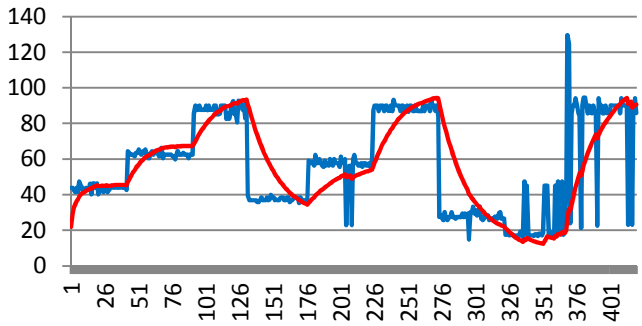

Fig. 7. Filter response (red), Q = $1x10^{-4}$ e R = 0,1.

D. *Mapping*

After detecting the pupil and treating the data, it is necessary to match the coordinates of the pupil to screen coordinates. This is called mapping and it consists of two parts. The first part is the calibration, which collects the data required for calculations. On the second part, the correlation between the pupil coordinates and the screen coordinates is calculated.

1) *Calibration*

The calibration consists of collecting the data from the pupil that matches the fixed coordinates of the screen, and these are the points of calibration. For this method 20 points of calibration were used, which results in the pupil points as seen on the example of **Fig. 8**.

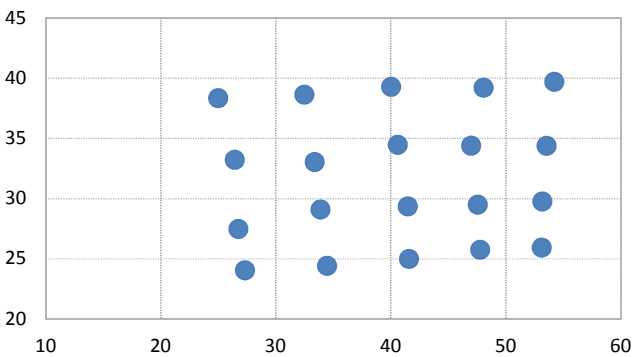

Fig. 8. Pupil calibration points

2) *Coordinates transformation*

The result from the calibration generates 32 straight lines and 12 quadrants, as seen in **Fig. 9**. When a pupil data is obtained, first the system checks on which quadrant it belongs. Then, the relation between the pupil coordinates and the quadrant straight lines (**Fig. 10**) are applied to the screen quadrant, resulting in a cursor position (**Fig. 11**).

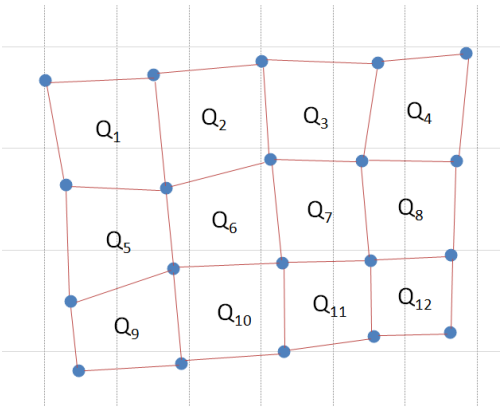

Fig. 9. Straight lines and quadrants

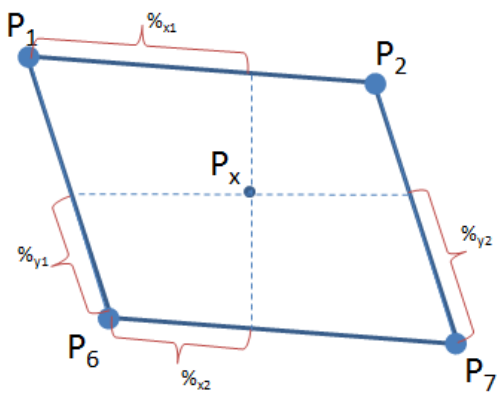

Fig. 10. Pupil relation

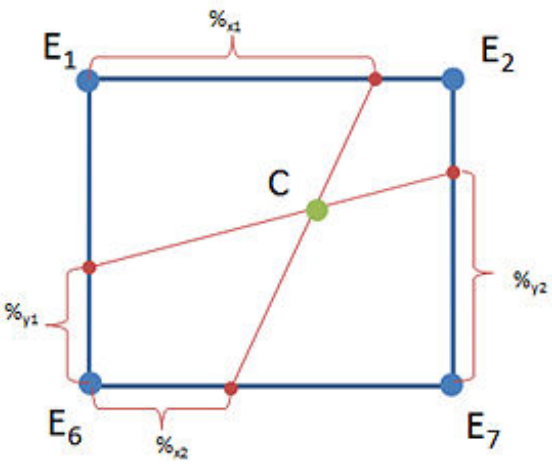

Fig. 11. Screen relation

E. Mouse click generation

Since the user is limited to the movement and blinking of the eyes due to its physical condition, there are two methods to perform the mouse click, either through an eye blinking or by a time out when staring at a particular point. In other words, the time out method consists of counting the time the cursor position is fixed. If the cursor stays in a certain coordinate (within a certain pre-defined value of hysteresis) during a pre-set time, then a mouse click is simulated.

For the eye blink method, it cannot be confused with the natural blink, and therefore, the eye blink should be slightly longer, about 1 second, although this time can be adjusted accordingly. The lack of glint is detected as the eye blinks and that triggers a mouse click event.

F. Resolution

Screen size and the distance between the user and the camera/IR cause variations on the resolution. The farther the user is from the system, the lower the resolution.

For a distance of about 60 cm from the screen and 30 cm from the camera and for a screen diagonal of 39.6 cm (15.6”), the system only recognizes 14x29 points. This means that when moving slightly the eye a big cursor jump on the screen is generated. This also influences the size and amount of onscreen buttons on the graphical user interface. For small buttons the user has great difficulty reaching them since the long jumps of the cursor do not ever coincide with the button.

G. Method flowchart

The flowchart of **Fig. 12** demonstrates the system’s behavior for the presented method and using eye blinking for mouse clicking.

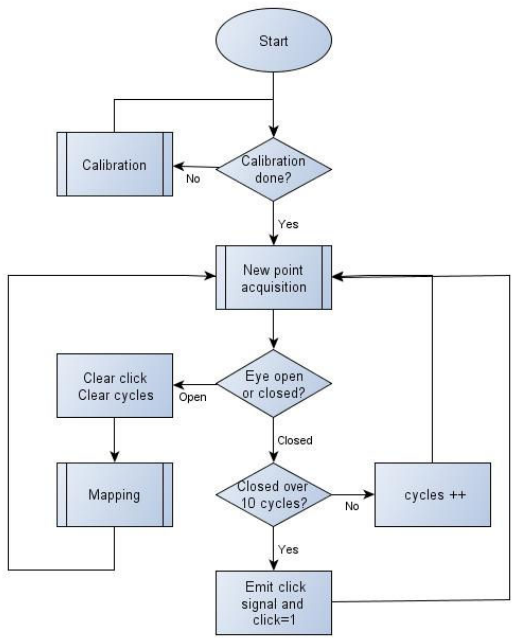

Fig. 12. Flowchart

H. Method discussion and conclusion

The complete system was developed in the laboratory and tested with different users which demonstrated a positive

feedback of the whole system's work. When the system was tested with a patient suffering from tetraplegia caused by an amyotrophic lateral sclerosis disease, some difficulties were found though. In this particular case, the user had low control on his eye blinking and in most cases his eyelid was half open. The latter undermined the circular detection of the pupil using the Hough transform method. A new approach had to be taken for these two important tasks. Also, the use of 20 calibration points has revealed tiring and time consuming for the user increasing his level of stress on the system operation. This had also to be altered in order to make it more practical. In general, the whole system proved its value, feasibility and usefulness once the patient, as a user of the system, perceived its potential and demonstrated its control over the computer mouse cursor and a way of communicating to the world again.

IV. Histogram Method

Based on the previous approach of pupil detection and tracking, this new method had to be robust enough to operate when the eyelid is half closed but still detecting the pupil and its position. After some tests it was found that the pupil area had a consistent low value in the grayscale values showing a strong potential for pupil segmentation.

A. *Segmentation by Threshold*

The Threshold method was applied to the original grayscale image in order to create a binary image with the low hysteresis defined with values slightly above the ones returned by the eye pupil. After the Threshold has been applied an Erode and Dilate methods are applied to reduce some possible noise found on the image. The final obtained image showed only the pixels that fulfill the pupil as shown in **Fig. 13**

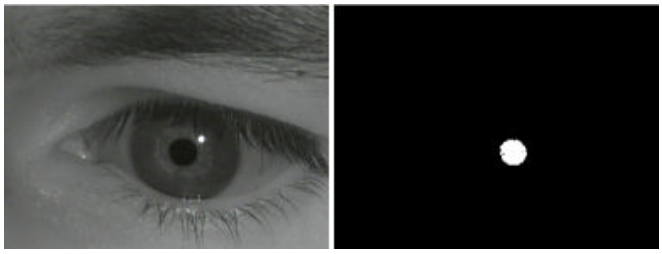

Fig. 13. Original grayscale image (left), binary image (right)

Fig. 14 shows the same Threshold method applied when the eyelid is half opened. It is clear from the image that only pixels from the eye pupil are captured. Some eyelashes crossing the pupil can also be observed in the resultant image. This method has proven robustness for pupil segmentation even in poor conditions.

Fig. 14. Original grayscale image with eyelid half opened (left), binary image of the obtained image (right)

B. *Pupil tracking by histograms*

Tracking of a blob can be achieved by different computer vision methods and histogram calculation is a light processing method that produces reliable and fast responses for this type of tasks. Vertical and horizontal histograms on each captured image were created counting the total amount of white pixels in the binary image and their maxima in both axes thus representing a rough center of the blob. A low pass filter from this output was used to smooth the sudden variations created by the movements of the pupil. This new method was implemented and tested successfully.

Even in poor conditions with the eyelid almost closed the few resultant white pixels from the eye pupil can be detected and tracked successfully using this method.

C. *Five point calibration*

As referred before the 20 point calibration method has revealed unacceptable and therefore a five point calibration was developed, based on a technique by Fang and Chang from Texas Instruments® [13], as shown in **Fig. 15**.

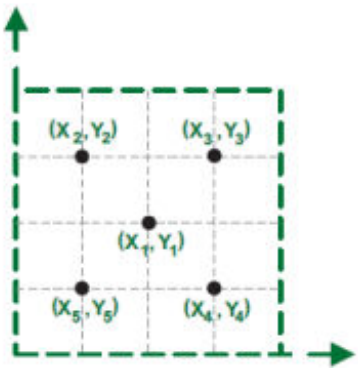

Fig. 15. Five point calibration technique [9]

Experiments were conducted in order to assess the technique's reliability and easiness of use and it was demonstrated that it performed flawlessly.

D. *Method discussion and conclusion*

This new method was tested with the same tetraplegia patient and the implemented changes confirmed a much better system behavior. The calibration technique has proven to be less painful to perform and the results were more satisfactory than previously. It has also shown that the difference of the pupil position in pixels between left and right positions (X axis) is on average 54 pixels whereas top and bottom positions (Y axis) is on average 23.

The Threshold method was found accurate but the low hysteresis had to be manually adjusted to adapt to a different environment (laboratory versus the house where the patient lives). An automatic hysteresis level calculation had to be developed.

Some instability though on the mouse cursor was found created by some fluctuation on the histogram results. The resultant histogram values on each axis is in most cases narrow and a single pixel change can make a considerable difference between the output coordinates hence creating the instability. The low pass filter helps smoothing this behavior but does not solve it. In conclusion it was found that the histogram method does not produce a stable and accurate detection of the pupil center. A new method was tried by calculating the center of mass of the pupil instead as described next.

V. Center of Mass Method

This last method was based on developing the automatic low hysteresis and center of mass calculation in order to achieve a better and automatic response to changes in new environments.

A. *Automatic hysteresis calculation*

The automatic level calculation of the low hysteresis was developed based on the amount of white pixels found on the binary image after the application of the Threshold method. The total amount of white pixels in the image is counted and by experimentation an ideal amount was calculated to achieve the best algorithm results.

On each captured image the amount of white pixels obtained is compared to the ideal amount returning an error difference. The new level is then calculated based on this error and the result passes through a low pass filter in order to avoid an abrupt change. This new hysteresis level is then used on the next captured frame until a stable and constant level is achieved. This process has a typical stability time of less than 5 frames (250 ms).

B. *Center of Mass or Centroid calculation*

In order to obtain the center point of the eye pupil blob on the binary image, the calculation of image Moments was utilized as it is available by the OpenCV library. Raw image Moments can be calculated as shown on Eq. 1 [14],

$$\boldsymbol{M_{ij}} = \sum_x \sum_y \boldsymbol{x^i y^j I(x,y)} \qquad \textbf{Eq. 1}$$

where *i* and j are the order and *I(x,y)* is the pixel's intensity on a grayscale or binary image. From the raw Moments a Centroid can be calculated by dividing the order 1 values from order 0 as shown on Eq. 2.

$$\langle \bar{x}, \bar{y} \rangle = \langle \frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \rangle \qquad \textbf{Eq. 2}$$

The resulting Centroid gives a rough estimate of the pupil center and the conducted experimentation has proven its accuracy and stability.

C. *Method discussion and conclusion*

The automatic hysteresis level calculation has performed well inside the desired parameters and it has shown its relevance since no adjustments were needed to changes in the user environment.

Also, the Center of Mass calculation has demonstrated a higher degree of stability when compared to the Histogram method. The mouse cursor became more stable when the eye pupil was steady and the user was able to get more control over the cursor and its position reducing his levels of stress.

It was then concluded at this stage that no further changes to the eye tracking algorithms were necessary.

VI. Graphical User Interface

An user friendly and easy to use software was developed to use the eye tracking system and to improve the user communication with other people. The software was developed using Nokia QT IDE platform for Linux. It was developed in the Portuguese language but it can be easily translated to any other language. The following figures show two examples of the graphical user interface (GUI) developed.

Fig. 16 shows the Main Menu with the accessible applications such as a Keyboard to write sentences, Phrases where the user can recall predefined sentences, Multimedia where multimedia sound and video files can be played, Internet where a browser like window is shown with large navigation buttons and Configurations where some settings can be defined such as the eye calibration. The latter is called by default when the program runs for the first time. The last button quits the program.

Fig. 16. Main menu

Fig. 17 shows the GUI of the virtual keyboard with some special buttons like a "Bell". When this button is clicked, a loud alert sound is reproduced and played through the computer speakers (optional) calling for the attention of someone nearby.

Fig. 17. Keyboard layout

The large buttons makes the system easier to use for the untrained eye and a short staring at a button presses it to activate a function or to write a letter or a digit.

VII. PROTOTYPE STAND AND LIGHT CONDITIONS

A stainless steel pipe prototype stand was developed that is made of a five castor swivel base with extra weight, a central pillar and a set of smaller pipes to hold the 17" flat screen, the camera and the IR light (

Fig. 18**) keeping a steady balance to avoid tipping. The stand was positioned in order to allow one eye of the user to clearly see the entire flat screen at the necessary distance between the user eye and the camera (around 60 cm from the screen and 40 cm from the camera). It also allows the user to watch TV that stands on the wall (not visible in**

Fig. 18) when he turns the eyes to the left side.

The user also has a 3 m x 2 m glazed door in front of him where sunlight goes in the room as it can also be slightly noticed from the left side of

Fig. 18. As one could expect this sunlight could create some visible variations in the captured images mainly when the room goes dark at night. With the developed system these variations were not significant since the use of the IR LED light was sufficient to illuminate the eye thus reducing any variation caused by the room illumination.

The conducted tests have demonstrated that in different light scenarios the system behaved properly and was consistent. No perturbation was found during operation from a completely dark room to a completely bright room. Although a small brightness variation around the user eye was noticed, the pupil segmentation was still possible allowing the system to follow it.

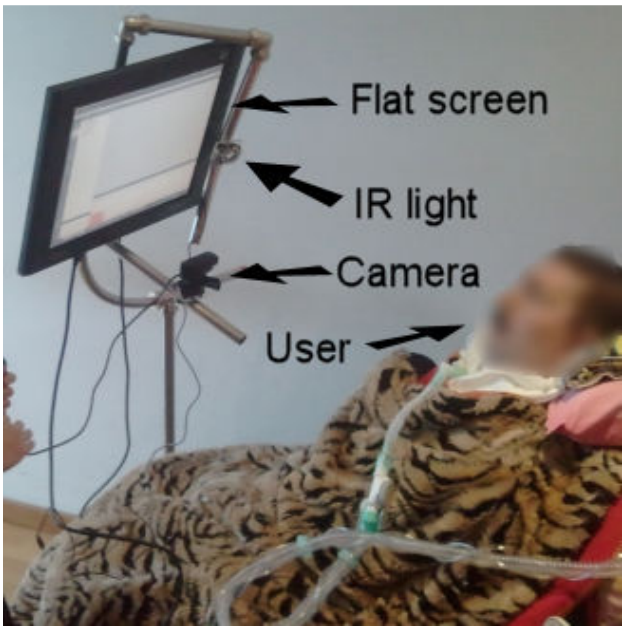

Fig. 18. System stand with a flat screen, camera, IR light and the user

VIII. CONCLUSION AND FUTURE WORK

The developed system was tested successfully and small variations of the distance between the user, the camera and screen are allowed. Using this system with other configurations will not have a predictable behavior because if the screen and/or camera are farthest from the user the eye movement between two points is smaller, thus decreasing the resolution. The optimal distances are the ones defined.

This system is also limited to users with no vision problems and it was not tested with users with asymmetric pupils, dry eyes, nystagmus or strabismus. It has been tested with people with glasses and contact lenses without being detected any problem although some reflection may occur due to backlight position such as sunlight when standing against a window. However, the system was not tested with people with light eyes, but it is anticipated that the system's behavior will not change, because a light iris contributes to a greater contrast between the iris and pupil.

It is a low cost system and defined initially and open to hardware changes without radically changing the original characteristics, i.e. another camera can be used, or a different computer screen, with same resolution and size, respectively. As for the software side, even though the computer must run the Linux operating system (Ubuntu version 4.10 in this case [15]), it is free and available everywhere in different languages. The computer in use is a small factor laptop known in Portugal as Magalhaes V2 (Magellan) due to its low price and enough computing power for the task in hands (1.6 GHz Atom processor and 1 GB of memory).

The three approaches described in the paper where necessary to accomplish the desired robustness but nevertheless only the consistent use of the system will show necessary improvements to be implemented in the future. The authors intent to increase the resolution of the system using a higher focal distance lens and possibly testing other low cost cameras with higher resolution. Although communication is of major importance at this stage for a person with amyotrophic lateral sclerosis, it is already planned the development of other interfaces with the environment controlled by the user such as the actuation of the room lights, doors, television, radio, among others, allowing greater independence to the user.

ACKNOWLEDGMENTS

The authors would like to thank the important contributions of Mr. Abel, his wife and Mr. Sampaio for the success of this work. This work was supported by the Automation and Robotics Laboratory from the Algoritmi Research Center at the University of Minho in Guimaraes. This work is funded by FEDER through the Operational Competitiveness Programme — COMPETE — and by national funds through the Foundation for Science and Technology — FCT — in the scope of project: FCOMP-01-0124-FEDER-022674

REFERENCES

[1] S. Gihan and C. Junchul, "Vision-Based Multimodal Human Computer Interface Based on Parallel Tracking of Eye and Hand Motion," in *Convergence Information Technology, 2007. International Conference on*, 2007, pp. 2443-2448.

[2] E. C. Lee, J. C. Woo, J. H. Kim, M. Whang, and K. R. Park, "A brain,Äìcomputer interface method combined with eye tracking for 3D interaction," *Journal of Neuroscience Methods,* vol. 190, pp. 289-298, 2010.

[3] E. C. Lyons, A. B. Barreto, and M. Adjouadi, "Development of a hybrid hands-off human computer interface based on electromyogram signals and eye-gaze tracking," in *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, 2001, pp. 1423-1426 vol.2.

[4] A. De Santis and D. Iacoviello, "Robust real time eye tracking for computer interface for disabled people," *Computer Methods and Programs in Biomedicine,* vol. 96, pp. 1-11, 2009.

[5] B. Noureddin, P. D. Lawrence, and C. F. Man, "A non-contact device for tracking gaze in a human computer interface," *Computer Vision and Image Understanding,* vol. 98, pp. 52-82, 2005.

[6] A. Duchowsky, *Eye tracking methodology theory and practice* 2nd ed. London: Springer, 2007.

[7] C. H. Morimoto, D. Koons, A. Amir, and M. Flickner, "Pupil detection and tracking using multiple light sources," *Image and Vision Computing,* vol. 18, pp. 331-335, 2000.

[8] J. B. Hiley, A. H. Redekopp, and R. Fazel-Rezai, "A Low Cost Human Computer Interface based on Eye Tracking," in *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, 2006, pp. 3226-3229.

[9] Z. Zhu and Q. Ji, "Eye and gaze tracking for interactive graphic display," *Machine Vision and Applications,* vol. 15, pp. 139-148, 2004/07/01 2004.

[10] B. Ashtiani and I. S. MacKenzie, "BlinkWrite2: an improved text entry method using eye blinks," presented at the Proceedings of the 2010 Symposium on Eye-Tracking Research \&\#38; Applications, Austin, Texas, 2010.

[11] J. S. Agustin, H. Skovsgaard, J. P. Hansen, and D. W. Hansen, "Low-cost gaze interaction: ready to deliver the promises," presented at the CHI '09 Extended Abstracts on Human Factors in Computing Systems, Boston, MA, USA, 2009.

[12] W. Garage. (2010, January 2010). *OpenCV Library - Wiki*. Available: http://opencv.willowgarage.com/wiki/

[13] W. Fang and T. Chang, "Calibration in touch-screen systems," in *Analog Applications Journal* vol. 3Q, ed: Texas Instruments Inc., 2007.

[14] H. Ming-Kuei, "Visual pattern recognition by moment invariants," *Information Theory, IRE Transactions on,* vol. 8, pp. 179-187, 1962.

[15] C. Ltd. (2010, January 2010). *Ubuntu OS*. Available: http://www.ubuntu.com