Eye-controlled mouse cursor for physically disabled individual

Conference Paper · February 2018

DOI: 10.1109/ICASET.2018.8376907

5 authors, including:

Mohamed Nasor
Ajman University

Mujeeb Rahman K k
Ajman University

Maryam Mohamed
Prince Sattam bin Abdulaziz University

Farida Mohamed
Alriyada college for health sciences

Eye-Controlled Mouse Cursor for Physically Disabled Individual

[1]Mohamed Nasor, [1]Mujeeb Rahman K K, Maryam Mohamed Zubair, Haya Ansari, Farida Mohamed.
[1]Department of Biomedical Engineering
Ajman University, Ajman, UAE
Email: m.nasor@ajman.ac.ae

***Abstract*— This paper presents a novel algorithm for controlling the movement of a computer screen cursor using the iris movement. By accurately detecting the position of the iris in the eye and mapping that to a specific position on the computer screen, the algorithm enables physically disabled individuals to control the computer cursor movement to the left, right, up and down. The algorithm also enables the person to open and close folders or files or applications through a clicking mechanism.**

Keywords—MATLAB; human computer interface; graphical user interface; iris;

I. INTRODUCTION

Personal computers were initially used for solving mathematical problems and word processing. In recent years, however, computers have become necessary for every aspect of our daily activities. These activities range from professional applications to personal uses such as internet browsing, shopping, socializing and entertainment.

Computers are designed to be readily accessible for normal individuals. However, for individuals with severe physical disabilities such as cerebral palsy or amyotrophic lateral sclerosis, usage of computers is a very challenging task. There have been many research studies on human computer interface (HCI) to improve the interaction between the user and the computer system. Most of these are applicable only to normal individuals. These interfacing methods include a touch sensitive screens, speech recognition methods and many others. Despite the success of these techniques, they were not suitable for the physically disabled individuals.

Many researchers have tried to develop methods to help the disabled to interact with computers by using signals such as electroencephalography (EEG) from the brain, facial muscles signals (EMG) and electro-oculogram (EOG) [1-3]. Other methods include limbus, pupil and eye/eyelid tracking [4-5], contact lens method, corneal, pupil reflection relationship [6] and head movement measurement [7]. These methods require the use of attachments and electrodes to the head, which makes them impractical. Other high end techniques [8] that are based on infrared tracking of the eye movements to control computers were exceptionally expensive and were not affordable for those who need them.

The method described in this paper is distinctive because unlike existing methods we did not use electrodes, infrared, or any other light source to track the eyes. The only hardware that is required is a PC or laptop along with a webcam, which makes it practical and feasible. By taking consecutive snaps of the user from the camera, the program is designed to process these frames individually at very high speed of processing and compare the iris shift in each frame with respect to the initial frame. The frame undergoes several stages of processing before the eyes can be tracked. After obtaining the processed image, the iris shift is calculated and the program prompts the cursor on the screen to move to the respective location.

II. METHODOLOGY

The first step was to use a face detection algorithm locate the face on an image frame captured by an ordinary webcam. The next step was to detect only the eyes from this frame. We consider tracking only one eye movement for faster processing time. Then the iris movement was tracked. Since the color of the iris is black, its image has a significantly lower intensity compared to the rest of the eye. This helps us in easy detection of the iris region. Taking the left and right corners of the eye as reference points, the shift of the iris as the person changed his eyes focus was determined. The shift was then used to map cursor location on the test graphical user interface (GUI).

III. IMPLEMENTATION

The algorithm for controlling the cursor by the eye iris movement was achieved through the following steps:

A. Face detection

In order to capture the face image accurately, the user sat upright with the eye level parallel to the webcam as shown in the Fig. 1.

The image of the user's face is captured using a MATLAB tool called vfm [9], using which images can be captured with the help of a webcam or any other imaging tools attached to the PC. Fig. 2 shows the image captured by this tool

Fig. 1: System setup for eye mouse.

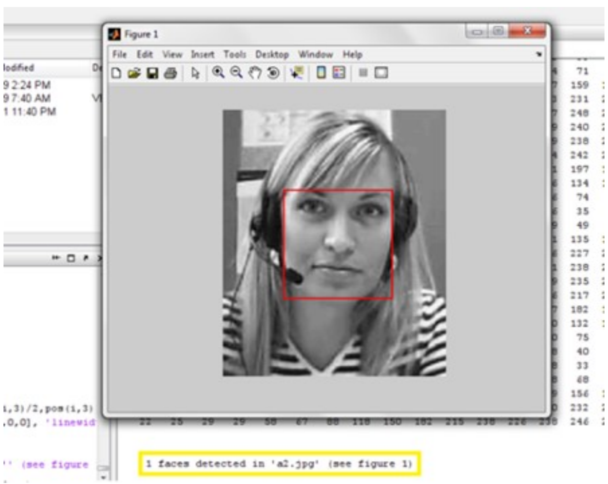


Fig. 2 Image detection and image capture

B. Extrating eye area

In order to extract the eyes region, the image of the face is divided into three equal horizontal areas. The upper third where the eyes are located is extracted as shown Fig. 3.

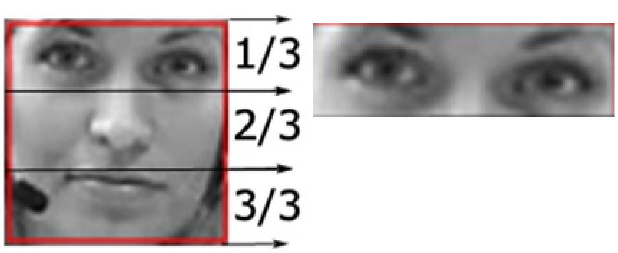
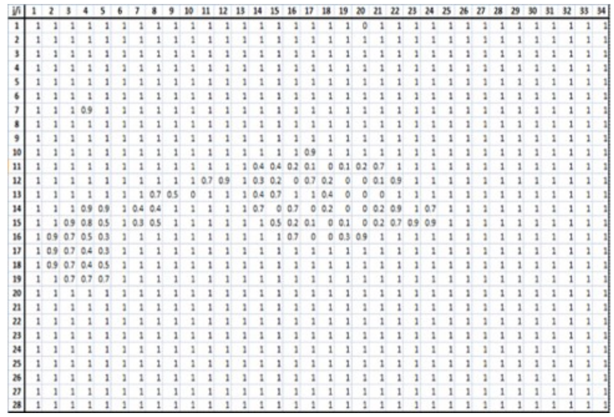


Fig. 3 Eye region extraction

C. Extracting eye location

The image of one eye was then extracted and normalized in order to remove the background noises and then it was converted to binary image to enhance contrast. The normalized pixel (0 – 1 scale) values of the left eye image is shown in Fig. 4.



j/i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	0.9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	0.4	0.4	0.2	0.1	0	0.1	0.2	0.7	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	0.7	0.9	1	0.3	0.2	0	0.7	0.2	0	0	0.1	0.9	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	0.7	0.5	0	1	1	1	0.4	0.7	1	1	0.4	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	0.9	0.9	1	0.4	0.4	1	1	1	1	1	0.7	0	0.7	0	0.2	0	0	0.2	0.9	1	0.7	1	1	1	1	1	1	1	1	1	1
15	1	1	0.9	0.8	0.5	1	0.3	0.5	1	1	1	1	1	1	0.5	0.2	0.1	0	0.1	0	0.2	0.7	0.9	0.9	1	1	1	1	1	1	1	1	1	1
16	1	0.9	0.7	0.5	0.3	1	1	1	1	1	1	1	1	1	1	0.7	0	0	0.3	0.9	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	0.9	0.7	0.4	0.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	0.9	0.7	0.4	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	1	1	0.7	0.7	0.7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
25	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
28	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fig. 4 Normalized pixels values of left eye image

D. Extacting iris region

Since the iris region is black, corresponding pixels' values were very low and when the image was normalized, these values were approximated to zero. Hence the boundaries of iris region were determined as shown in the Fig. 5.

iris =

0	0	0	0.1000	0.4000
0	0	0	0	0
0	0	0	0	0
0	0	0	0	1.0000
0.9000	0	0	1.0000	1.0000

Fig.. 5 Pixel's intensity values in the iris area

E. Tracking eye movement

In order to avoid the head movement effect on the iris location, the two corners of the eye region were taken as reference, thus the iris movement was tracked accordingly.

F. Calculating iris shift

In order to calculate the movement within the eye as the user gaze on the screen, the user was asked to look at a central pattern on the screen and the iris boundaries are determined. The user was then asked to look at other pattern to the left, right up and down with respect to the central pattern. The shift in the iris was then calculated and was used to decide the direction and position of the cursor. Fig. 6 shows the binary images of the eye when looking in different directions. In the Fig. 6, the image at the top was captured when the user was looking at the center of the screen while the middle image was captured when the user was looking to the right and lower image when the user was looking to the left. The relative shift in the iris position is used to determine the cursor position on the test GUI shown in the Fig. 7.

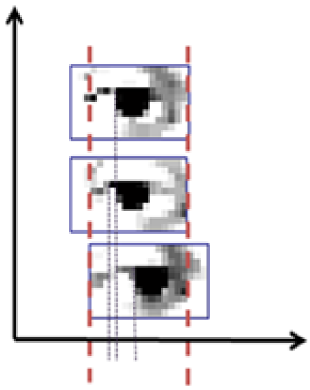
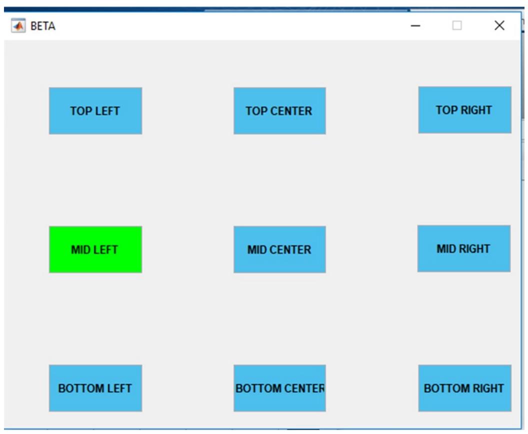


Fig. 6 : Shift in the Iris to left and right



A prolonged blink was used as a signal for clicking the pattern. The principle is that when the eyes are closed, the image variable 'iris' will no longer contain the actual iris. As a result, the average of the sum of the pixels in 'iris area will be higher than that of the actual iris (since eyes closed will not give any darker region than with open eyes). The icon on the GUI where the blink occurred will change its color, indicating a click.

H. Algorithm

The flow chart of the MATLAB program that was developed to perform the various stages discussed in the sections 3.1- 3.7 is given in the Figs. 8 and 9.

IV. RESULS AND DISCUSSION

A MATLAB based algorithm for face detection, eye region extraction, iris tracking and cursor control was successfully developed and tested. A test GUI comprised of nine boxes on the computer screen was used for the validation of the cursor

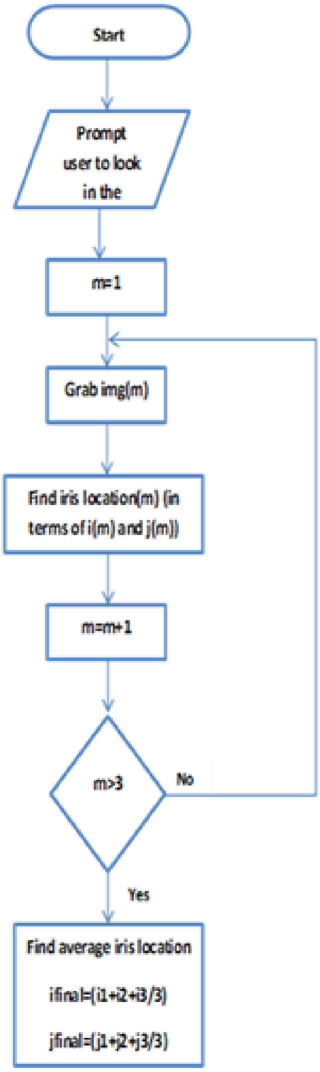


Fig. 8 Flowchart of iris detection algorithm

movement as shown in Fig. 7. The iris movement was accurately tracked and the cursor was successfully moved to all nine boxes in the test GUI. Up and down movement however, was not highly consistent and the system requires further enhancement. Clicking action was also successfully tested through a change in the color of boxes.

IV. CONCLUSION

A system that enables a disabled person to interact with the computer was successfully developed and tested. The method can be further enhanced to be used in many other applications. The system can be adapted to help the disabled to control home appliances such as TV sets, lights, doors etc. The system can also be adapted to be used by individuals suffering from complete paralysis, to operate and control a wheelchair. The eye mouse can also be used to detect drowsiness of drivers in order to prevent vehicle accidents. The eye movement detection and tracking have also potential use in gaming and virtual reality.

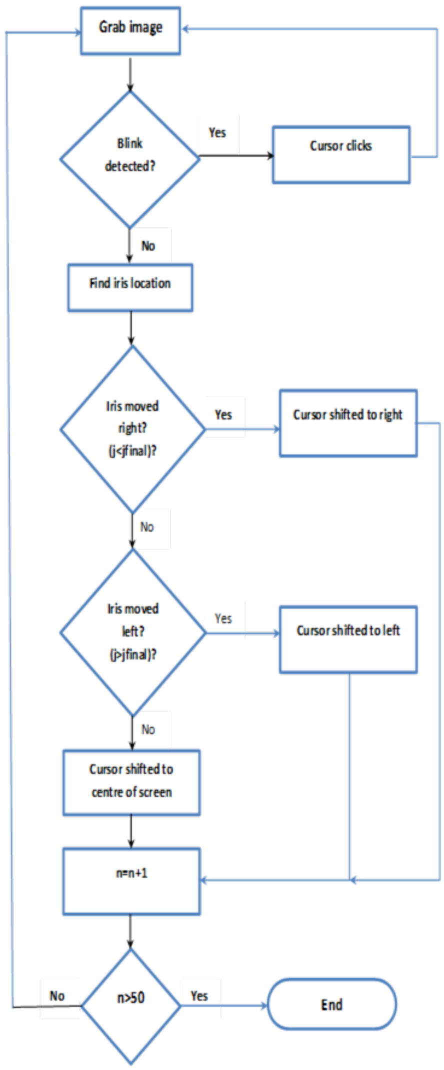


Fig. 9 Flowchart of blink detection algorithm

REFERENCES

[1] B. Rebsamen, C. L. Teo, Q. Zeng, M. Ang. Jr. "Controlling a wheel chair indoors using thought" IEEE Intelligent Systems, 2007, pp. 18-24.

[2] C. A. Chin "Enhanced Hybrid Electromyogram / Eye gaze tracking cursor control system for hands-free computer interaction", Proceedings of the 28th IEEE EMBS Annual International Conference, New York City, USA, Aug 30-Sept 3, 2006, pp. 2296-2299.

[3] J. Kierkels, J. Riani, J. Bergmans, "Using an Eye tracker for Accurate Eye Movement Artifact Correction", IEEE Transactions on Biomedical Engineering, vol. 54, no. 7, July 2007, pp. 1257-1267.

[4] A. E. Kaufman, A. Bandyopadhyay, B. D. Shaviv, "An Eye Tracking Computer User Interface", Research Frontier in Virtual Reality Workshop Proceedings, IEEE Computer Society Press, October 1993, pp. 78-84.

[5] T. Kocejko, "Device which will allow people suffered from Lateral Amyotrophic Sclerosis to communicate with the environment", MSc thesis, January 2008.
G. A. Myers, K. R. Sherman, L. Stark, "Eye Monitor", IEEE Computer Magazine, Vol. March 1991, pp. 14-21.

[6] C. Collet, A. Finkel, R. Gherbi, "A Gaze Tracking System in Man-Machine Interaction", Proceedings of IEEE International Conference on Intelligent Engineering Systems, September 1997.

[7] B. Hu, M. Qiu, "A New Method for Human-Computer Interaction by using Eye-Gaze", Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, October 1994.

[8] P. Ballard, G. C. Stockman, "Computer operation via Face Orientation", Pattern Recognition vol. 1. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference, 1992.

[9] https://www.mathworks.com/matlabcentral/fileexchange/247-vfm