A HEAD-MOUNTED SENSOR-BASED EYE TRACKING DEVICE: EYE TOUCH SYSTEM

Cihan Topal, Ömer Nezih Gerek and Atakan Doğan
Anadolu University
{cihant, ongerek, atdogan}@anadolu.edu.tr

Abstract

In this study, a new eye tracking system, namely Eye Touch, is introduced. Eye Touch is based on an eyeglasses-like apparatus on which IrDA sensitive sensors and IrDA light sources are mounted. Using inexpensive sensors and light sources instead of a camera leads to lower system cost and need for the computation power. A prototype of the proposed system is developed and tested to show its capabilities. Based on the test results obtained, Eye Touch is proved to be a promising human-computer interface system.

Keywords: Human-computer interface, eye tracking, IrDA sensors, data acquisition.

1. INTRODUCTION

Eye tracking systems have been developed to provide many different solutions for the problems in human-computer interface systems. They are used for either providing the usability of a computer or a system for disabled people, or improving the human abilities in controlling an external system.

This work aims for providing an alternative interface method for tracking the eye gaze of humans. Several research projects and products were developed for this purpose [1-10]. However, most of the available devices work with a camera sensor and capture the human face to detect and interpret the human gazes or gestures to recognize where the person looks at or what eye gesture he/she performs. Available literature mostly reports algorithms regarding the eye gaze analysis from the captured camera video, indicating that camera sensor usage has become a traditional solution. Obviously, such an image analysis system requires high amount of computational power to perform video and image analysis. Although camera systems become relatively cheap, the complexity of the algorithms and the computational power requirements directly reflect to the price of such systems.

A solution of cheap and few-in-number sensor use is proposed in this work. The system is desired to be low-cost, with low computational requirements, and without a mobility compromise. For this purpose, the wearable interface idea is adopted and an eye-glasses-like apparatus is developed. Similar studies can be found in [9], [10].

2. EYE TOUCH SYSTEM

Eye touch system (ETS) is designed to recognize human commands given by a pair of eyes. In order to achieve the recognition of human eye commands, ETS must determine the eye gaze direction in real-time. Consequently, ETS is developed to be capable of discovering the eye gaze direction in real-time by acknowledging following simple facts:

- Human eye consists of mostly white color, with a central iris part which has darker tones than white such as brown, black or blue.
- Different colors reflect the light in different amounts and around different wavelengths.

Considering these basic light reflection principles and the spherical structure of the eye surface, it is possible to acquire reflected light intensities from specific portions of the eye surface and then find out the eye gaze direction based on the measured reflected light intensities.

Eye touch system, as shown in Figure 1, is composed of components such as infrared light sensitive apparatus (ILSA), data acquisition device (DAQ), computer software and external power supply. The function of each ETS component is briefly described as follows.

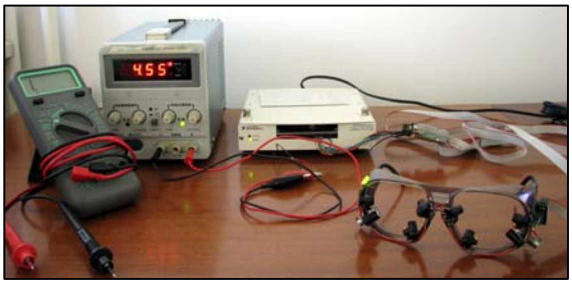

Figure 1. Eye Touch System – from right to left; ILSA, DAQ, and power supply.

Infrared light sensitive apparatus (ILSA): ILSA resembles eyeglasses without any lenses. It includes eight IrDA sensitive sensors and eight IrDA light sources (LEDs) mounted on its frame to detect the movements of the iris. Each LED is attached to one IrDA sensitive sensor, which results in eight sensor-

LED pairs. On the apparatus frame, there are four sensor-LED pairs on the left and four on the right, each

Figure 2. A closer look to ILSA – front and rear views.

of which is focused on a specific portion of the eye. Figure 2 shows the designed apparatus.

IrDA sensitive sensors used are TSL262R light-to-voltage optical sensors, which are manufactured by TAOS. TSL262R responses the light in 800~1100 nm wavelength range with a sharp peak at 940 nm. In addition, IrDA LEDs emit light at about 940 nm wavelength.

ILSA is connected to the data acquisition device and power supply over a 10-channel cable. This cable carries 8 channel analog voltage signals to the DAQ as well as the power from the power supply which is necessary to feed the sensors and LEDs. Note that the current ILSA weighs about 150 grams.

Data acquisition device (DAQ): DAQ deployed is NI DAQPad-6015 of National Instruments. DAQ samples the output of the sensors on the ILSA with the resolution of 16-bit and sampling rate of 10 Hz, and serves the sensor information over an USB interface to the application running on a computer. The available DAQ device currently receives a reference voltage from a power supply.

In the final ETS, a microcontroller powered over a USB interface will replace DAQ - the microcontroller will sample the sensor outputs and send the collected data over the USB interface to the computer. The sensors and LEDs will be powered by the USB as well. It should be emphasized that all the required hardware can be put on the ILSA, resulting in a portable device. As far as the cost of the finished ILSA is concerned, it will not probably exceed 100 USD, thanks to inexpensive microcontrollers and IrDA components.

Eye touch software: The application software is developed in the Microsoft Visual Studio 2003 environment where DAQ device's own function library is used to pass the collected data to the application.

Eye touch software consists of two main schemes: training and tracking. The first scheme performs calibration and training of the system. In order to calibrate the system, the software requires the user to look at specific regions on the screen or blink his eyes by the help of a simple interface as shown in Figure 3. While the user is looking at the specific locations or blinking his eyes, the software collects the necessary information to adapt the system to the physical properties of the user's eye and prepares the algorithm parameters which will be used in tracking. The ETS training takes about a minute.

The second scheme, tracking, decides for the mouse action using the acquired information from the sensors via DAQ. That is, it provides the control of the Windows mouse cursor with the click operations. Specifically, three main mouse operations are supported: standing still, moving in one of the four directions, and click operations. For the first two operations, the current state of the prototype is not able to exactly track the gaze within a real mouse sensitivity. Therefore, a cursor-like usage is adopted. When the user looks "inside" the monitor frame, the mouse pointer does not move. Whenever the gaze is out of the monitor boundaries (right-left-up-down), the cursor operations are activated. Such an interface was found to be practical for several applications such as reading text from the monitor, where the user would not want the mouse pointer constantly occluding the area that is being looked at. If the user is looking at (or over) one of the boundaries of the screen, it assumes that the user wants the cursor to move. As a result, the cursor is moved towards that boundary horizontally or vertically. For the left and right click operations, the user must close its respective eye for about 500 ms (an adjustable parameter by the user) while looking at the screen. Once this case is detected, it will perform the desired action. The computational load of the tracking is 90 floating point (FP) subtractions, 48 FP multiplications, and 42 FP additions after each of the sampling operation.

In Figure 3, a case of left eye blink is shown. If the user keeps his eye closed for a while (i.e., 500ms), the application calls for the left click API function of Windows OS with the current coordinates of the cursor. It should be noted that there has not been any confusion between normal blinks and eye closure events during the trails. Since the application continues to run even if it is minimized, ILSA enables the usage of the Windows computer (i.e., browsing around the folders, open or close files or applications, performing operations which require click operations) by means of the glances of its user. The operation accuracy and usability of the tracking application is still under research.

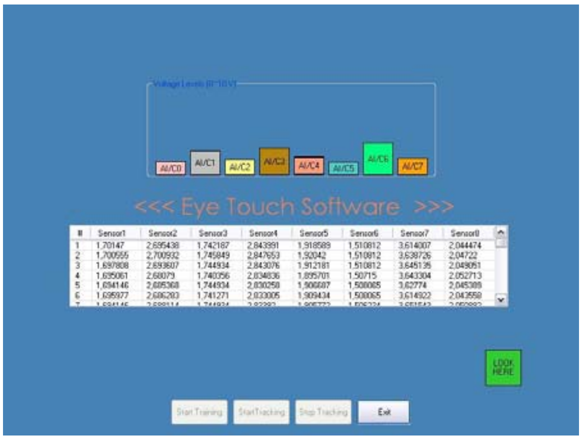

Figure 3. A screenshot from the ETS training.

Figure 4. A screenshot from the ETS tracking.

3. TRAINING & TRACKING ALGORITHMS

As explained in Section 2, eye touch software includes a training algorithm followed by a tracking algorithm. For the beginning, relatively simple training and tracking algorithms were implemented to test the system; yet, they have yielded satisfactory results in terms of replacing a mouse by ETS.

The training application runs as follows. Initially, it requires its user to look at 13 different points: 9 are on the screen and the others are towards out of the screen (right-left-up-down). Next, the user is asked to blink his right, left and both eyes, respectively. As a result, the application includes 16 different events, each of which corresponds to a class in the training/tracking algorithm. For each class, 10~20 samples are acquired over 8 channels, leading to a set of 8-dimensional vectors per class. Then, each sensor's predetermined dark voltage values are subtracted from the acquired voltage values for that sensor to obtain more accurate information. After that, some of these vectors need to be filtered and eliminated since a person cannot look at a specific point for a long time (i.e., 5 seconds). That is, the gaze moves in an involuntary manner and the vectors which are acquired when the user's eye lost its focus must be filtered out from the class. The algorithm devised for filtering out such "outlier" vectors is shown in Figure 6.

According to Figure 5, a set of N samples are collected first for the related class. Then, the mean vector of the vector set is computed and the Euclidean distances between each vector and the mean vector are found. After that, the vectors are sorted in ascending order in terms of their Euclidean distances and the farthest vectors are considered as outliers which should be eliminated. This leaves us with half of the vectors. Finally, the mean vector of these remaining "robust" vectors is computed and it is taken to be the identifier vector of that class.

After the identifier vector is computed for each class, the vector table becomes ready for the tracking scheme. During the tracking, the application collects vectors and calculates mean square error (MSE) values between the current acquired vector and each vector in the vector table. Then, the algorithm finds the closest class with the minimum MSE and performs the corresponding action of that class. Such a nearest neighbor algorithm works well under most of the conditions. However, the issue of consistency in the cursor movements requires a fine tuned set of "rules" which are considered as future research issues. The tracking algorithm runs in a continuous manner as long as the goggle apparatus is worn.

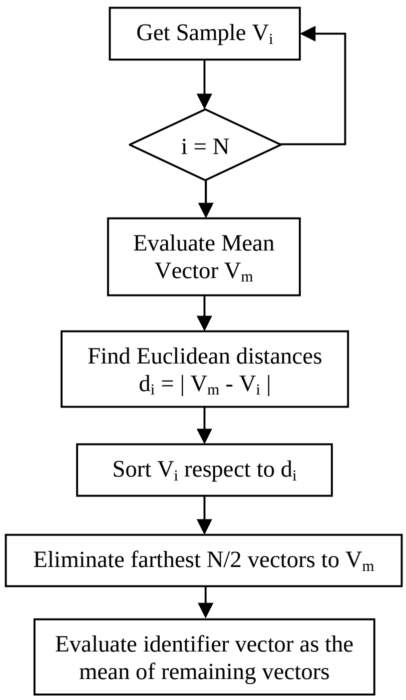

Figure 5. Flow chart of the training algorithm.

4. EXPERIMENTAL RESULTS

Satisfactory test results are obtained during the experiments. Figure 6 shows the sensor output voltage values that were acquired during a training scheme with six different classes. Each class corresponds to a specific location on the screen. In the figure, each vertical column stands for one of the classes. Each line represents one of the 8 sensors and 10 samples were collected for each class. It is easy to see the boundaries of each class where the signal levels change abruptly.

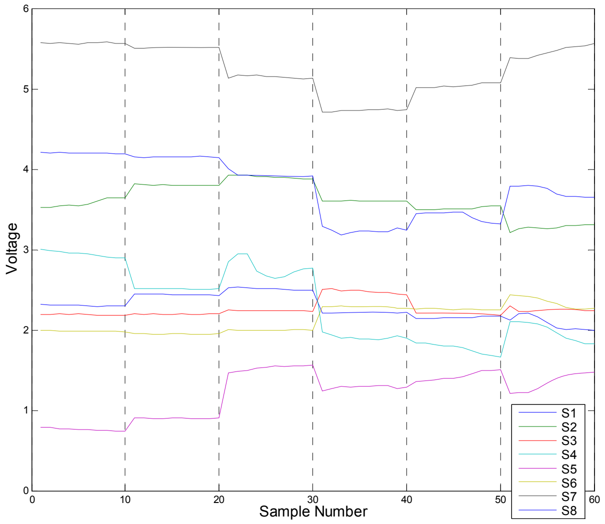

Figure 6. Voltage vs. sample sketch of the data acquired during the training scheme.

Figure 7 renders the output voltage data of three sensors, namely Sensor 2, Sensor 4 and Sensor 7. The 3-D plot clearly shows the suitability of the data for the classification purposes. Based on Figure 7, the apparatus is working fine – that is, the reflected light intensities acquired by the sensors for the six different classes are easily separable. Although, classes 1 and 6 have a scattered vector distribution, other sensor outputs exhibit sufficiently compact regions.

It should be noted that ETS was presented in a national conference where the conference participants were allowed to try it. During these trials, a success rate of 80% was easily achieved in terms of its usability.

5. CONCLUSIONS

In this study, a portable and cheap eye tracking system is proposed and implemented, and experimental results are presented. The main differences of this system from the similar ones can be summarized as non-existence of a camera sensor, cost of the system and portability.

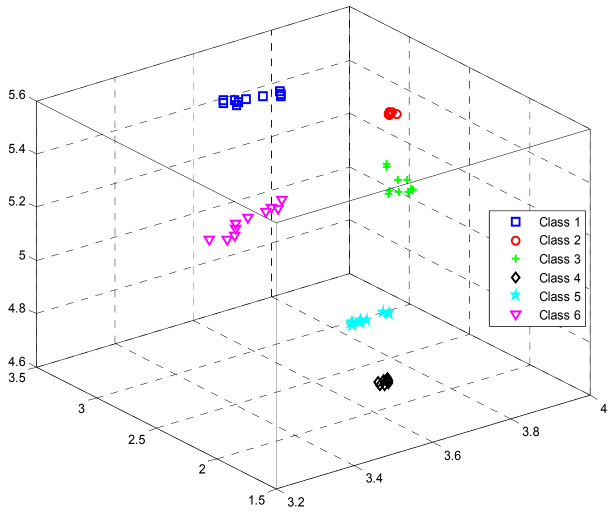

Figure 7. Output voltage data of sensor 2, 4, and 7.

REFERENCES

[1] COGAIN NoE (in EU IST 6th framework): Communication by Gaze Interaction, online at: http://www.cogain.org.

[2] Lawrence H. Yu, Moshe Eizenman, “A new methodology for determining point-of-gaze in head-mounted eye tracking systems,” *IEEE Trans. on Biomedical Engineering,* 51(10), 2004.

[3] Metin Akay, et al, “A system for medical consultation and education using multimodal human/machine communication,” *IEEE Trans on Information Tech. in Biomedicine*, 2(4), 1998.

[4] A. Amir, et al “An embedded system for an eye-detection sensor,” *Computer Vision and Image Understanding,* vol. 98, pp.104–123, 2005.

[5] R. S. Allison, M. Eizenman, B. S. K. Cheung, “Combined head and eye tracking system for dynamic testing of the vestibular system,” *IEEE Trans. on Biomedical Engineering,* 43(11), 1996.

[6] D. Droege, T. Geier, D. Paulus, “Improved low cost gaze tracker,” *The 3rd COGAIN Conference on Communication by Gaze Interaction,* 2007.

[7] D. Agrafiotis, N. Canagarajah, D. R. Bull, M. Dye, “Perceptually optimised sign language video coding based on eye tracking analysis,” *Electronics Letters*, 39(24), 2003.

[8] D. Li, J. Babcock, D. J. Parkhurst, “openEyes: A low-cost head-mounted eye-tracking solution,” ACM Eye Tracking Research and Applications Symposium, 2006.

[9] J. P. H. Reulen et al, “Precise recording of eye movement: the IRIS technique. Part 1,” Medical and Biological Engineering and Computing, 26(1), 20-26, 1988.

[10] Skalar IRIS IR Eyetracker, online at: http://www.crsltd.com/ catalog/skalar/index.html.