DESIGN AND IMPLEMENTATION OF A HUMAN COMPUTER INTERFACE TRACKING SYSTEM BASED ON MULTIPLE EYE FEATURES

[1]Shazia Azam, [2]Aihab Khan, [3]M.S.H.Khiyal

[1]Software Engineering, Fatima Jinnah Women University, Rawalpindi, PAKISTAN
[2]Department of Computer, Fatima Jinnah Women University, Rawalpindi, PAKISTAN
[3]Department of Computer Science/Software Engineering, Fatima Jinnah Women University, Rawalpindi, PAKISTAN
E-mail: [1]sheeze14@yahoo.com, [2]Aihabkhan@yahoo.com, [3]m.sikandarhayat@yahoo.com

ABSTRACT

This research is aimed at designing and implementing a human computer interface system that tracks the direction of the human gaze. The motion and direction of the iris is used to drive the interface by positioning the mouse cursor accordingly. The location of the iris is done in batch mode. This implies that the frames are stored in a permanent storage device (hard disk, USB drive etc) and are retrieved one by one. Each of the frames is processed thus finding the location of the iris and thereby placing the mouse cursor accordingly. Such a system that detects the iris position from still images provides an alternate input modality to facilitate computer users with severe disabilities.

Keywords: *Iris Detection, Gaze Tracking*,

I. INTRODUCTION

Human eyesight is one of mankind's most major senses. The eye is different from the other body parts that make up the human's sensor array (see Shaviv, 2002). A person's eyes convey a great deal of information with regards to the meaning behind certain facial expressions. Also, the direction in which an individual is looking shows where his or her attention is focused. By tracking the position of the irises, useful interfaces can be developed that allow the user to control and manipulate devices in a more natural manner. Traditional interfaces are typically one-sided, with the bandwidth from computer to user far greater than that from user to computer. The movement of a user's eyes can provide a convenient, natural, and high-bandwidth source of additional user input, to help redress this imbalance (Grauman et. al., 2003 and Magee et. al., 2004). Some people are so severely paralyzed that they only have the ability to control the muscles in their eyes. For these people, eye movements or blinks are the only way to communicate. This research aims in developing a system that can aid the physically challenged by allowing them to interact with a computer system using only their eyes.

A. Background

Traditional human-computer interfaces demand manual dexterity. Unfortunately, people with physical disabilities cannot fully enjoy the benefits provided by computer systems. This is because the conventional mouse and keyboard were designed to be used by those who are able bodied. The task of reducing or eliminating the communication barriers between man and machine is an arduous task (Kaufman et. al., 1993 and Magee et. al., 2005). There is little motivation in providing an alternative communication tool for those whose physical abilities are extremely limited. Human eye movements have the potential to be a convenient, natural, high bandwidth and fast input mode of computers due to their communication power.

B. Analysis of existing techniques

There is no universal method to track the movement of the eyes. In any study, the choice of the method rests with the particular demands of the application. During the analysis phase of this research, three techniques were analyzed; the Limbus tracking, Pupil tracking, and

Electrooculography. Each technique has its own strong points and drawbacks.

Limbus Tracking explains a method of tracking the eye using the limbus. The limbus is the boundary between the white sclera of the eye and the darker iris.

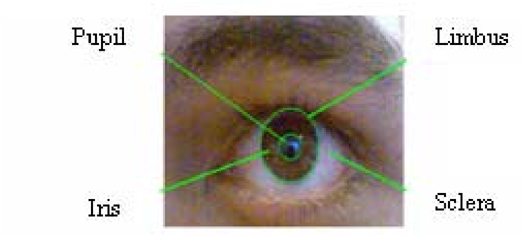

Fig.1. Diagram of eye

As the sclera is white and the iris is darker, this boundary can easily be optically detected and tracked. This technique is based on the position and shape of the limbus *relative to the head*, so either the head must be held quite still or the apparatus must be fixed to the user's head (Kyung-Nam and Ramakrishna, 1999). This technique is negatively affected by the eyelid often concealing all or part of the limbus (Morimoto et. al., 2000). This makes its uses restricted to horizontal tracking. Generally this method does not involve the use of infra red light.

Pupil tracking is a method of gaze detection that is commonly used often in conjunction with other forms of tracking. There are several reasons for this, but the main advantage is the notion of the "bright spot" (Grauman et. al., 2003). Similar to the case of red eye when taking flash photographs at night, infrared can be used in pupil detection to create a high intensity bright spot that is easy to find with image processing. This bright spot occurs when infrared is reflected off the back of the pupil and magnified by the lens. The main advantage of pupil tracking is that as the border of the pupil is sharper than the limbus, a higher resolution is achievable. Also, as the pupil is never really covered by the eyelid, x-y tracking is more feasible as compared to Limbus tracking. The disadvantage is that the difference in contrast is lower between the pupil and iris than between the iris and sclera-thus making the border detection more difficult (Kawato and Tetsutani, 2002).

Electrooculography is based on electrodes attached to the human skin. Due to the higher metabolic rate at the retina compared to the cornea, the eye maintains a constant voltage with respect to the retina. This is approximately aligned with the optical axis. Voltage rotates with the direction of gaze and can be measured by surface electrodes placed on the skin around the eyes (Basu et. al., 1996). This system is easily mounted elsewhere other than directly in front of the person as compared to other methods.

Electrical skin potential tracking is often used in medicine and optometry to diagnose certain conditions. For example, researchers from the department of Ophthalmology at the University of Vienna (Kyung-Nam and Ramakrishna, 1999) have used EOG to diagnose sixth nerve palsy. From their research it can be seen that while a clinical orthoptic examination is still the best method of diagnosis, Electrooculography provides a suitable replacement in the follow-up stage of treatment programs (Point Grey Research, 2003). While these uses are beneficial, the use of electrodes makes this method of gaze tracking unsuitable for use in everyday applications.

Table 1. Analysis of Eye Detection Techniques

Technique	Limbus Tracking	Pupil Tracking	EOG
Face Access	Poor to good	Good	Good
Subject Contact	Headmount/ chinrest	None	electrodes
Subject Variety	Low	Reasonable	-
Real time response	-	6-12ms delay	Yes
Accuracy	h=0.5-7° v=1-7°	0.003°	±1.5-2°

The table shows that the use of each technique varies from one application to the other. EOG equipment is best suited for lab use. Pupil tracking uses infrared which is not good for everyday applications. Limbus tracking is best in case of horizontal and not vertical tracking.

The goal of this research is to develop a human computer interface system that is based on iris tracking. The iris is widely used as the starting point for detection and tracking. It is an important eye feature that is circular in shape and that can be detected easily. For this research there was a need to perform calculations using the limbus, pupil and the iris hence the paper has been named as such.

II. EYE TRACKING ALGORITHM

The goal of the eye-tracking algorithm is first to locate the eyes of the user from an image and then use the location information to perform certain functions. Static images are retrieved from an image library and are used to initiate the system. In the first stage, an efficient image enhancement sharpening filter is employed. This is followed by a simple method to segment the eyes. Following this, an iris detection method is used to find the direction of the user's gaze and finally the computed direction information of eye movements is used to drive the computer interface. Each step will be explained in detail in the following sections.

A. Image Enhancement

The first step after retrieving the input image is to enhance it. This increases the image definition by improving contrast. In the presence of noise, the sharpening and smoothing of the image are important pre-processing steps. These are usually the precursors in many operations such as object recognition, edge detection, feature extraction and pattern recognition (Liu et. al., 2002). Smoothing removes noise but typically also blurs edges. To facilitate edge detection and other similar processes, deblurring (sharpening) of the image is required. After several experimental enhancement schemes, it was found that the unsharp filter provided results that were closest to the ones desired. The unsharp filter is created from the negative of the Laplacian filter. Certain parameters are tuned to provide improved results.

$$\frac{1}{(\alpha+1)}\begin{bmatrix} -\alpha & \alpha-1 & -\alpha \\ \alpha-1 & \alpha+5 & \alpha-1 \\ -\alpha & \alpha-1 & -\alpha \end{bmatrix}$$

Fig. 2. Algorithm of unsharp filter

Here alpha controls the shape of the Laplacian and must be in the range 0.0 to 1.0.

B. Image Segmentation

In a general sense, image segmentation is defined as one process that partitions a digital image into disjoint (non overlapping) regions (Liu et. al., 2002). It is the first stage of extracting the eyes and is carried out by color image segmentation in the L*a*b* (also known as CIELAB or CIE L*a*b*) color space. Conversion of the input RGB image to the L*a*b color space prior to segmentation and clustering is necessary. The L*a*b* space consists of a luminosity 'L*' or brightness layer, chromaticity layer 'a*' indicating where color falls along the red-green axis, and chromaticity layer 'b*' indicating where the color falls along the blue-yellow axis. Within this color space, the quantification of visual differences is possible thus enabling the distinction of the different colors in the image. The approach used is to choose a small sample region for each color and to calculate each sample region's average color in 'a*b*' space.

Each color marker now has an 'a*' and a 'b*' value. Each pixel in the eye image is classified by calculating the Euclidean distance between that pixel and each color marker. The smaller the distance, the more closely that pixel matches the color marker. For example, if the distance between a pixel from the input image and the red color marker is the smallest, then the pixel would be labeled as a red pixel. The label matrix contains a color label for each connected pixel and can thus be used to separate objects in the original image.

C. Binarization

The first step to localize the eyes is through binarization. The segmented RGB image was first converted to grayscale by eliminating the hue and saturation information while retaining the luminance. The grayscale image is binarized using a global threshold value that is automatically calculated.

D. Morphological Opening

Morphological refinements are applied to the binary output in order to reduce some of the effects of noisy pixels. Since these spurious errors are generally much smaller, morphological techniques are used after the color based image segmentation to clean the binary image prior to extracting the skinned region. Morphological opening removes small objects from an image while preserving the shape and size of larger objects in the image. The connected components that have fewer than *P* pixels are also removed producing the final, cleaned image.

E. Boundary Tracing

Tracing the boundaries of the eyes is important as finding the outline of the eyes makes it easier (computationally) to localize the position of the irises. The eye boundaries in the binary image were found by tracing the exterior boundaries of objects, as well as boundaries of holes inside these objects. The boundaries of the outermost objects (parents) are traced along with their children (objects completely enclosed by the parents).

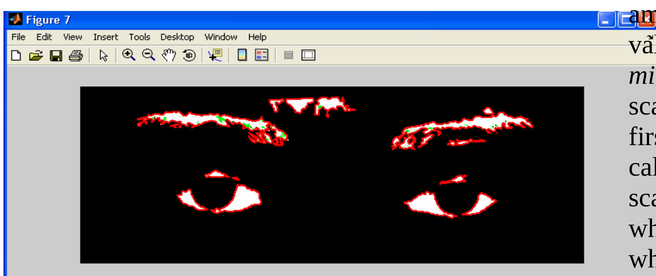

Fig. 3. Boundary traced image

F. Image Cropping

The image was cropped in order to find the correct iris position in both eyes hence. The original image was cropped into two images-one of left eye and one of right eye. Calculations are then performed separately on both of the images and results are matched for both.

G. Iris Position Detection

Several calculations were performed on both cropped images in order to detect the actual position of the iris. This in turn indicates which direction the user is looking in. There were 8 parameters calculated, namely: *(min_x, y_min_x, max_x, y_max_x, min_y, x_min_y, max_y, x_max_y).*

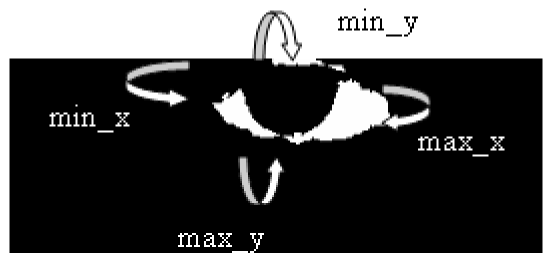

Fig. 4. Eye parameters

min_x is the value of the x-coordinate calculated on the left corner point of both eyes and *y_min_x* is the value of the y-coordinate at the same point. *min_x* was calculated by performing a raster scan and finding a white pixel whose x-coordinate value is minimum among all white pixels. Similarly at this point the value of y was also calculated. *max_x* is the value of x calculated on the right corner point of both eyes and *y_max_x* is the value of y at the same point. *max_x* was calculated by performing a raster scan and finding a white pixel whose x value is maximum among all white pixels. Similarly at this point value of y was also calculated. The value of *min_y* was initially set to 1. It was changed after scanning the image left to right and finding the first white pixel. At this point value of x was also calculated. *max_y* was also calculated after scanning the image from left to right until a white pixel whose y value is greater than all white pixels is encountered. At this point value of x was also calculated.

After calculating the coordinates of the eye, a variable by the name of *lok* was declared to indicate the direction the user was looking. The *lok* variable could take on one of three values at a single time- 1) Iris is left 2) Iris is right 3) Iris is center. For this to be accomplished some calculations were performed in the second stage of segmentation which included no of black and white pixels, left and right count.

The number of black and white pixels were counted by counting the number of 0's (black) and 1's (white) in the region enclosing the area from *min_y* to *max_y* When the user is looking straight i.e. the iris is centrally placed, the number of black pixels must be greater than the number of white; *lok* is therefore set to 3 in this situation. If the number of black pixels is less than the number of white pixels then this indicates that the user is looking either to the right or to the left. It is therefore necessary to count pixels in both regions. For the left, the number of white pixels were counted within the region enclosed by *max_y* to *min_x*. similarly, the number of white pixels was counted in the region enclosed by *max_y* and *max_x*. When the left count is greater than the right count, then the user is found to be looking towards the right and vice versa.

H. Driving the Interface

Mouse events were triggered based on the calculated values of the *lok* variable. All mouse events were generated in Visual C. When the value of *lok* is calculated as 1, the cursor moves to the left. Similarly, when value of *lok* is 2, the cursor moves to the right. When the user is looking straight i.e. the iris is in the center and *lok* is 3, then the mouse click is generated at the current position of the mouse.

III. EXPERIMENTAL RESULTS

The system has been tested for its accuracy as well as its usability. It was noticed that successful gaze detection requires extremely accurate segmentation of the eyes. MATLAB 7 was used as the testing platform for design & implementation. This is because MATLAB handles a range of computing tasks in engineering and science, from data acquisition and analysis to application development. The MATLAB environment integrates mathematical computing, visualization and a powerful technical language (Mathworks, 1999 and Knight, 2000). Testing was done on eye images in which the user is looking towards right, left and centre.

Color based segmentation using L*a*b color space was applied on the images shown in Figure 5 to get the segmented images shown in Figure 6.

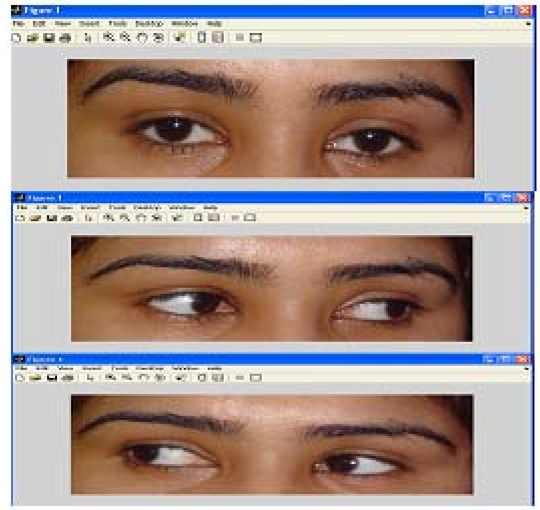

Fig. 5. User looking straight left and right respectively

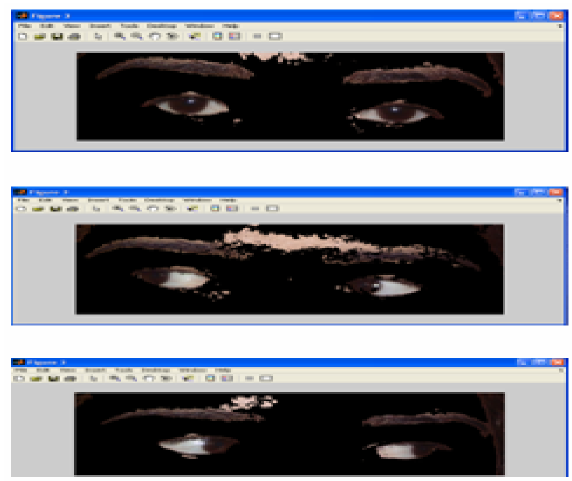

Fig. 6. Segmented Images

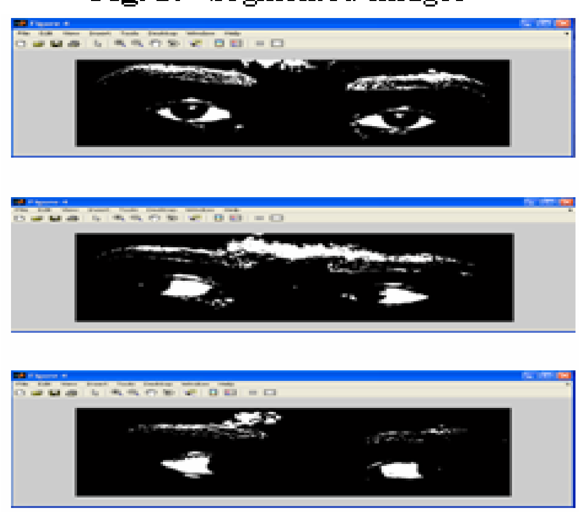

Fig. 7. Binary Images

The segmented images shown in Figure 6 were subsequently thresholded to get the binary images shown in figure 7.

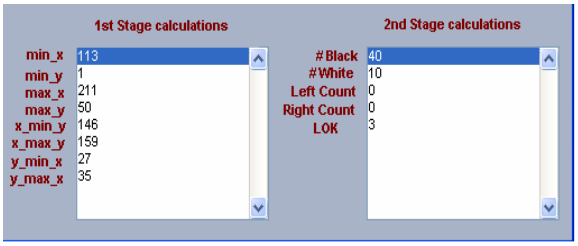

Fig. 8(a) Calculations when looking straight

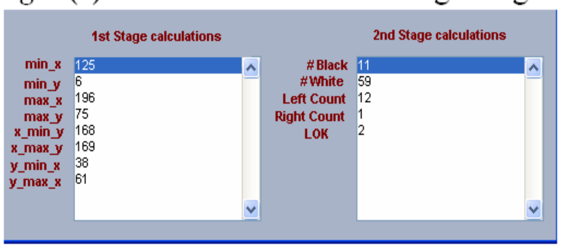

Fig. 8(b). Calculations when looking to the left

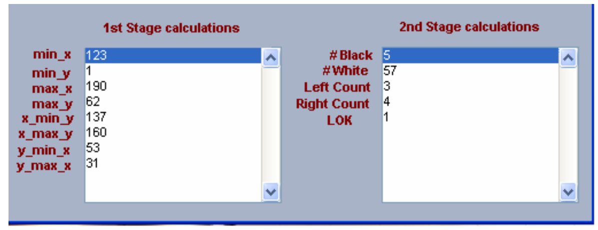

Fig. 8(c). Calculations when looking right

1st and 2nd stage parameters were calculated. Given in Figure 8(a) is the list of calculations showing the value of *lok* for the user looking straight and in figure 8(b,c) when looking to the left and right respectively.

Results show that we can locate and detect the eyes in images with different iris positions. As the research has been conducted on static images, a higher degree of accuracy is achieved. Results vary in certain cases such as when the user looking up, down or has closed his or her eyes. Correct classification of the user gaze direction is 80%. This value can be increased by including further directions such as up and down.

A. Results on different eye images

More results on different eye images are presented in figure 9.

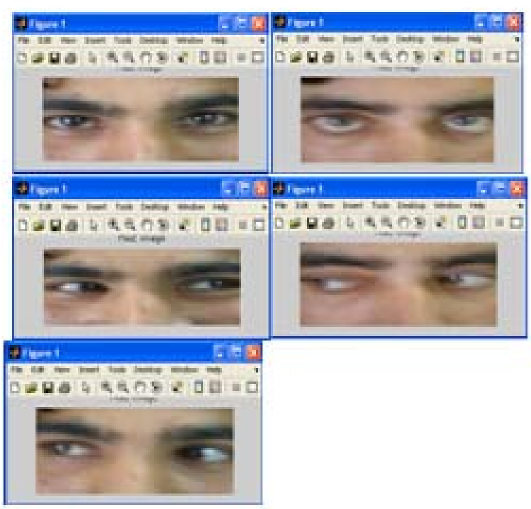

Fig. 9(a). original eye images

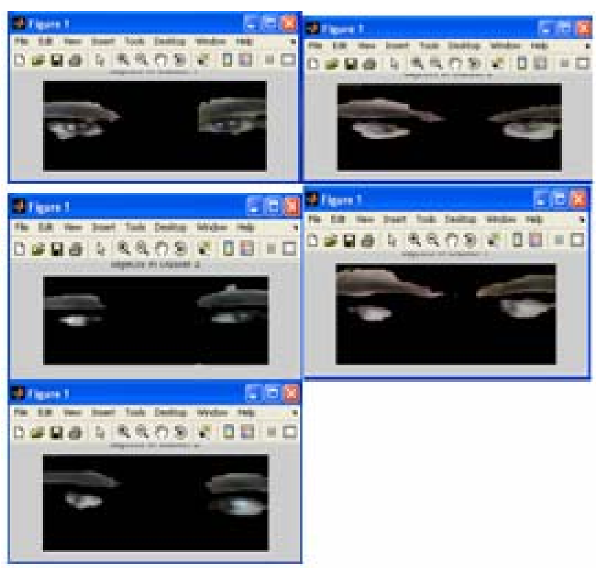

Fig. 9(b). Segmentation results

IV. CONCLUSION AND FUTURE WORK

The algorithm and software presented in this research provides an alternative communication method for people with severe disabilities. Although some aspects of the system are still under experimental development, the project has been an overall success in achieving the goal of reliable detection of human eye gaze direction. Under static mode where detection of three directions is performed on single scaled images, a detection accuracy of approximately 80% is achieved. The system can be expanded to work in real time such that the algorithm is applied on a live video stream allowing the user to drive the interface. The detection of more gaze directions and also for blinks and closed eyes can also be included. This will lead to a more functional and robust system. The system will be more functional as for each new gaze there will be a new command. It will be more robust as each of the different gazes will be handled appropriately. Such a system can then be marketed for commercial use. Imperative to any extension on this project would be speed improvements. Specifically, a Direct Show implementation written in C would provide increased speed.

REFERENCES:

[1]. Basu, S., I. Essa and A. Pentland. (1996). *Motion regularization for Model Based Head Tracking*. In 13th International conf. on pattern recognition (ICPR'96), vol 3, Vienna, Austria, p. 611.

[2]. Grauman, K., M. Betke, J. Lombardi, J. Gips and G. Bradski. (2003). Communication via Eye Blinks and Eyebrow Raises: Video-Based Human-Computer Interfaces. Universal Access in the Information Society, 2(4), 359-373, November 2003.

[3]. Kaufman, A.E., A. Bandopadhay and B.D. Shaviv. (1993). An Eye Tracking Computer User Interface. Research frontiers in virtual reality workshop proceedings, IEEE Computer Society Press, October 1993, pp 120-121.

[4]. Kawato, S. and N. Tetsutani. (2002). *Detection and Tracking for Gaze-Camera control*. Proc. VI, ATR Media Information Science Laboratries, Japan.

[5]. http://citeseer.ist.psu.edu/kawato02detection.html

[6]. Knight, A. (2000). *Basics of MATLAB and Beyond*. Chapman and Hall/CRC Press, Florida, USA.

[7]. Kyung-Nam, K. and R. S. Ramakrishna. (1999). *Vision Based Eye-Gaze tracking for Human Computer Interface.* IEEE International Conf. on Systems, Man, and Cybernetics, Tokyo, Japan, Oct. 1999.

[8]. Magee, J.J., M.R. Scott, B.N. Waber and M. Betke. (2004). EyeKeys: A Real-time Vision Interface Based on Gaze Detection from a Low-grade Video Camera." Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04), Volume 10: Workshop on Real-Time Vision for Human-Computer Interaction (RTV4HCI), Washington, D.C., pp. 159-166, Washington, D.C., June 2004.

[9]. Magee, J.J., M.R. Scott, B.N. Waber and M. Betke. (2005). "A Real-Time Vision Interface Based on Gaze Detection -- EyeKeys" in "Real-Time Vision for Human-Computer Interaction," Kisacanin, Branislav; Pavlovic, Vladimir; Huang, Thomas S. (Eds.) 2005, pages 141-157, Springer-Verlag.

[10]. MathWorks. (1999). *Getting Started with Matlab 5*, Oxford: Oxford University Press.

[11]. Morimoto, C.H., D. Koons, A. Amir and M. Flickner. (2000). Pupil Detection and tracking with multiple light sources, Image and Vision Computing, vol 18, pp 331-335, 2000, Haro, Antorio, Myron.

[12]. Point Grey Research (2003). *Point Grey Research Web Page*.

a. http://www.ptgrey.com

[13]. Shaviv, B. (2002). The Design and Improvement of an eye controlled interface. Computer Science Department , State University of New York at Stony Brook, NY 11794-4400, USA.

[14]. http://www.cs.sunysb.edu/~vislab/projects/eye/Reports/report/

[15]. Liu, X., F. Xu and K. Fujimura. (2002). *Real*-Time eye detection and tracking for driver observation under various light conditions. In IEEE Intelligent Vehicle Symposium, Vol. 2, pp 344-351.