Jurnal Teknologi, 50(D) Jun 2009: 43–57

REAL-TIME EYE TRACKING AND IRIS LOCALIZATION

HUSNIZA RAZALLI[1*], RAHMITA WIRZA O. K. RAHMAT[2] & RAMLAN MAHMUD[3]

Abstract. Robust, non-intrusive human eye detection problem has been a fundamental and challenging problem for computer vision area. Not only it is a problem of its own, it can be used to ease the problem of finding the locations of other facial features for recognition tasks and human-computer interaction purposes as well. Many previous works have the capability of determining the locations of the human eyes but the main task in this paper is not only a vision system with eye detection capability. Our aim is to design a real-time face tracker system and iris localization using edge point detection method indicates from image processing and circle fitting technique. As a result, our eye tracker system was successfully implemented using non-intrusive webcam with less error.

Keywords: Real-time face tracking; iris localization; image processing; edge detection; circle fitting

Abstrak. Masalah sistem pengesanan mata yang tegar tanpa sebarang gangguan adalah satu isu yang penting dan mencabar di dalam bidang visi komputer. Masalah ini bukan hanya mengurangkan masalah dalam carian ciri-ciri paras rupa untuk proses pengecaman tetapi juga boleh digunakan untuk memudahkan tugas pengenalpastian dan interaksi antara manusia dan sistem komputer. Walaupun kebanyakan hasil kerja terdahulu telah pun mempunyai keupayaan menentukan lokasi mata manusia tetapi objektif utama rencana ini bukan tertumpu kepada pengesanan mata sahaja. Objektif kajian adalah untuk merekabentuk sebuah sistem masa nyata dan terperinci, iaitu sistem pengesanan muka berskala dengan ciri-ciri petunjuk pergerakan mata berdasarkan pergerakan anak mata (iris) dengan mengunakan teknik penempatan yang terhasil daripada teknik pemprosesan imej dan teknik muatan bulatan. Hasil daripada kajian ini telah pun berjaya diimplimentasikan menggunakan kamera web dengan ralat yang minimum.

Kata kunci: Pengesanan mata masa nyata; penempatan anak mata; pemprosesan imej; pengesanan bucu; muatan bulatan

1.0 INTRODUCTION

Eye tracking and eye movement-based interaction using computer vision techniques have the potential to become an important component in future perceptual user interfaces. So by this motivation designing a real-time eye tracking software compatible with a standard PC environment is the main aim of this paper.

[1,2&3] Department of Multimedia, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400. Serdang, Selangor, Malaysia

* Corresponding author: Tel: 017-6605303/03-888839595-318. Email: [1]husniza@tmrnd.com.my. [2]rahmita@fsktm.upm.edu.my, [3]ramlan@mimos.com

In general, the term "eye detection" is widely used when static face images are of concern and the main aim is to find the face region which contains both eyes, and eye tracking term is used referring to the process of continuously detecting eyes in video sequences which contains only face images [1]. In this paper the term eye tracking means real-time, continuously detection of human eyes individually and extraction of eye features with scale invariance property and without making the assumption that the image sequences contain only face images.

The most accurate, but least user-friendly technology uses physical attachment to the front of the eye. A non-slipping contact lens is grounded to fit precisely over the corneal bulge. Another popular common technology is based on non-contacting, special equipment aided vision techniques such as illuminating the eye with a barely-visible infrared light source. These methods are obviously practical only for laboratory studies, as they are very awkward, uncomfortable for practical approaches. In this paper a more practical real-time approach for simultaneously tracking and feature extraction of individual eyes is implemented using a web camera based vision technique without using any special equipment given above. The eye tracker sits at a several meters range to the camera and head motion is restricted only to the extent necessary to keep the face, eye region and pupil of eye within view of the camera. The eye tracker provides data about the location of the face and iris of the eye. The x and y coordinates data of both eyeball areas, outline of the eye area are detected using image processing techniques and also located the position of the iris using circle fitting method.

The developed technique is aimed to be a fast and easy to operate real-time method, thus, it is suitable for ordinary user settings outside the laboratory environment although it is not consider being as accurate as equipment based techniques given above.

2.0 PROPOSED METHOD

In this research, several approaches are combined for developing an eye tracker system. As our approach does not make the assumption that the image sequences contain only face images; the method has to start dealing with the problem of face detection in real-time application. With this intention, an in depth study of the RGB and HSV concepts were made and the correlation between these two models was drawn before proceeding to the face detection. After these pre-processing steps, search areas for left and right eyes are automatically determined using the geometrical properties of the human face and in order to locate each eye individually, the training images are divided into row and column information supplied by the cropping function. Search regions for left and right eyes are individually passed to the eye detection algorithm to determine the exact locations of each eye. After the detection of eyes, eye areas are individually passed to the iris position extraction process which are based on the image processing and edge detection function to indicate the iris from the located eye. Then proceed to the localization of the eye by mapping edge point data with circle fitting method. By combining all the process into a system, finally an application is developed using Matlab program. We tested the system several times using sequences of images with random face and eye motion before applying for real-time program. Overall process flows of the proposed system are outlined as in Figure 1.

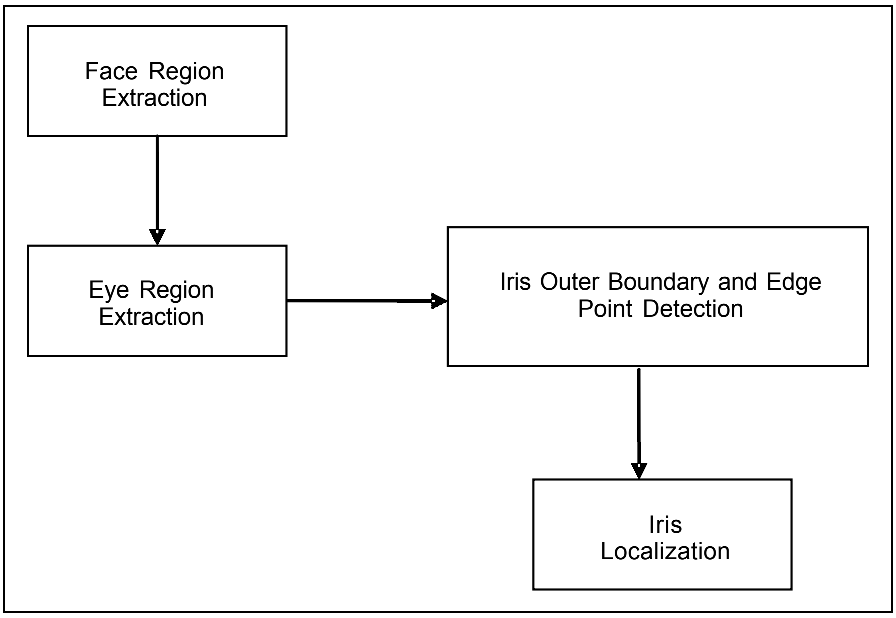

Figure 1 Overall process flow of the proposed system

3.0 FACE REGION EXTRACTION

The proposed face detection technique is to find the potential face regions in the image by using skin colour information. We apply a fast algorithm for detecting human faces in colour images. The algorithm uses colour histogram for skin (in the HSV space) in conjunction with edge information to quickly locate faces in a given image. The proposed algorithm has been tested on various real images and its performance is found to be quite satisfactory.

Our face detection system consists of several steps. The first step is to classify each pixel in the given image as a skin pixel or a non-skin pixel. The initial step in designing a skin colour filter; first we need a sample data which is image of target skin(s). In order that, the sample representing the wide possibilities of face skin colours collected from 56 persons with different skin types appearing from the population a collage of faces. The example of collected samples is given in Figure 2. The next step is changing the image from RGB to HSV, than extract the Hue and Saturation dimensions into separate new variables (H & S) where this reduces the effect of illumination as shown in Figure 3.

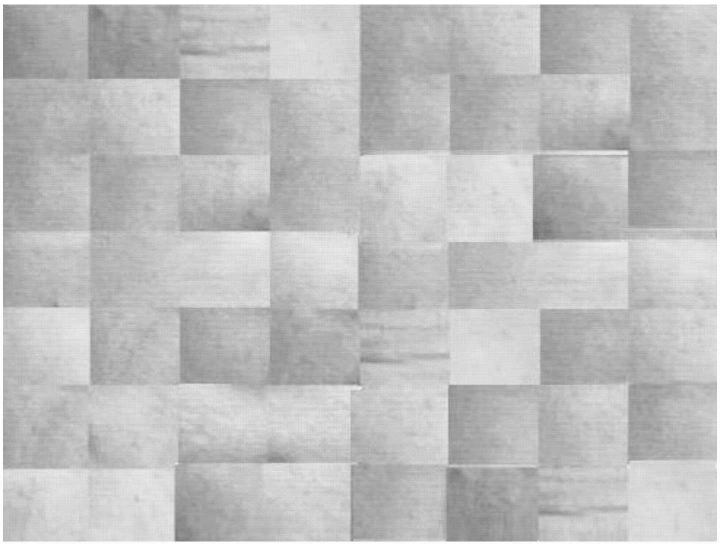

Figure 2 Face skin colours collected from 56 persons

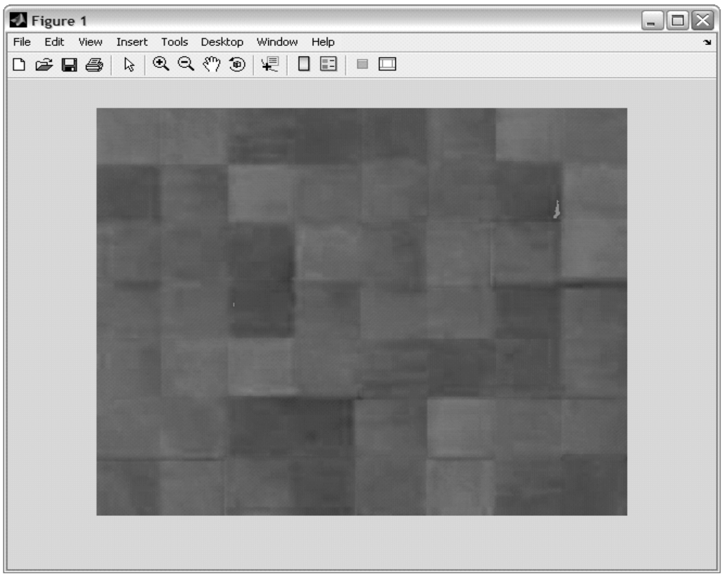

Figure 3 HSV colour space converted image

Then we plot both a 3D and 2D plot of the target skin collage pixel values Hue by Saturation. It is important to notice that the values within the V dimension are never used (i.e., V/intensity values are too unstable). The plotted figures allow us to determine the skins spatial region of interest (see Figure 4 and Figure 5).

From our experiments, we found skin colour has its range on H (hue) and S (saturation) which is H value should be less than 0.1 or bigger than 0.9 where minimum

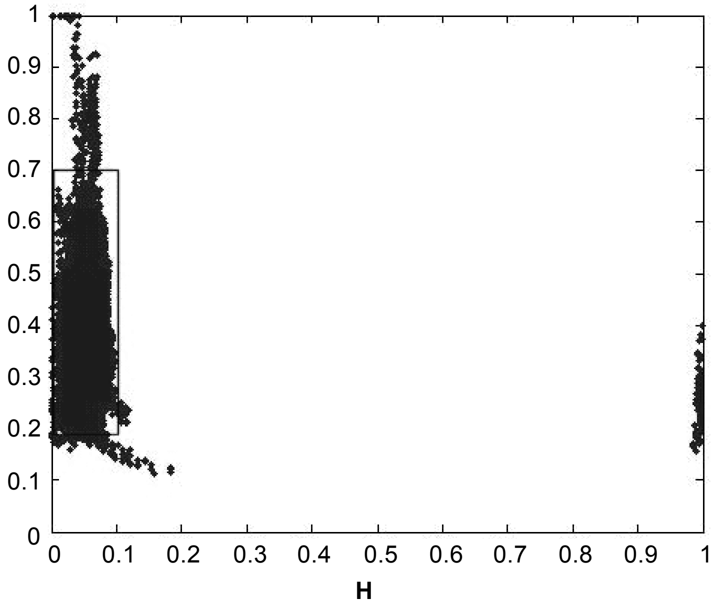

Figure 4 Face skin colour projected in 2D HS color space

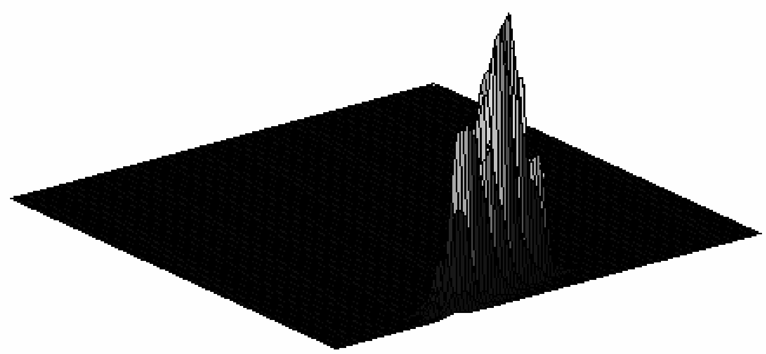

Figure 5 Face skin colour projected in 3D HS color space

range is 0 and maximum range is 1 and S value should be bigger than 0.2. We give the range as below;

$$\text{Skin colour} = (H<0.1 \mid H>0.9)\ \&\ S>0.2. \tag{1}$$

Based on equation (1), we construct a skin colour classification algorithm as below;

Step 1 Input a colour image, transform this image from RGB colour space to HSV colour space.

Step 2 Classify a pixel as a skin colour pixel, if H falls in the range [H<0.1 or| H>0.9] and S falls in the range [0.2, 0.6]; otherwise, the pixel is set as a non-skin colour pixel.

Step 3 The gray levels of all skin colour pixels are set to 255, and those of other pixels are set to 0. The equation is given as below;

Pixel (x, y) = 255 (white) II skin pixel (2)

Pixel (x, y) = 0 (black) II nonskin pixel (3)

From the above algorithm we construct a function to detect skin region and do the experiment to several samples. The function is generated using Matlab programs with several image processing tools. The samples consist of single face image with and without background, multiple faces image and multiple person images. The samples are taken from internet and face databases as shown in Figure 6, Figure 8, Figure 10 and Figure 12. Skin segmentation results taken from the samples are demonstrated as Figure 7, Figure 9, Figure 11 and Figure 13. As mentioned earlier, the white pixel stated in these figures are a skin region.

People of same race have similar skin colour, but still have difference place. Transforming colour into HS colour space, we can exclude lighting factor and converge skin colour to a range of hue and saturation. Except using colour information, we also assist image procession skills in post-processing. After skin colour classification, we can obtain one or more connected skin colour regions like the result shown in Figure 4, but it's not good enough for next step because it filled with fracture holes and regions. We hope to converged connected part more complete, and remove small fracture regions. Hence morphological operations are utilized to refine the segmented results. After those operations, we assume each connected region is a face, then apply face verification algorithm on each face region to verify if it is a face.

Figure 6 A single faced images

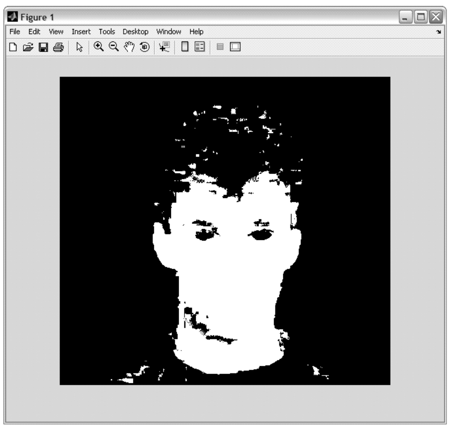

Figure 7 The segmented single-faced image

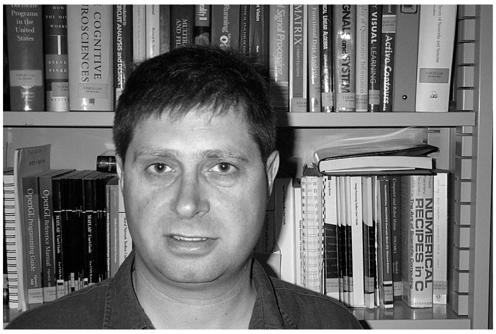

Figure 8 A single faced images with cluttered background

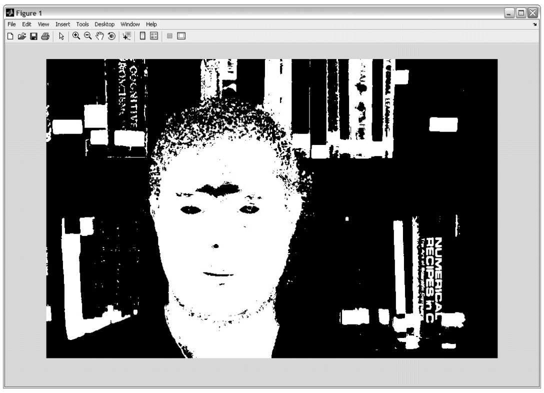

Figure 9 The segmented single faced image with cluttered background

Figure 10 A multiple person image as input image

Figure 11 A segmented multiple person image as input image

Figure 12 A multiple faced image as input image

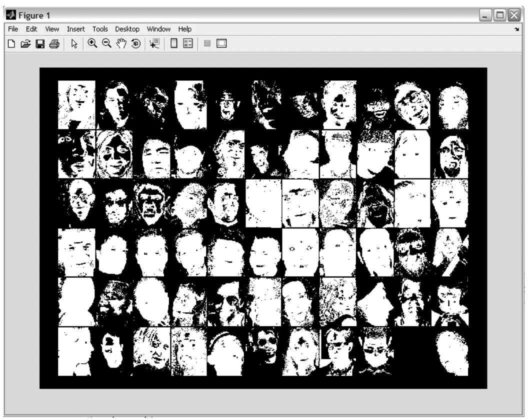

Figure 13 The segmented multiple faced image

The second step is to identify different skin regions in the skin detected image by using connectivity analysis. We group the skin pixels in the image based on an 8-connected neighbourhood i.e. if a skin pixel has got another skin pixel in any of its 8 neighbouring places, then both the pixels belong to the same region. At this stage, we have different regions and we have to classify each of these regions as a human face or not. This is done by finding the centroid, height and width of the region as well as the percentage of skin in the rectangular area defined by the above parameters. The centroid is found by the average of the coordinates of all the pixels in that region. For finding height, the y-coordinate of the centroid is subtracted from the y-coordinates of all pixels in the region. Find the average of all the positive y-coordinates and negative y-coordinates separately. Add the absolute values of both the averages and multiply by 2. This gives the average height of the region.

The last step is to remove the noise each of the skin regions. This is done by applying morphological an opening-closing filter operation to the binary image and than finally labelling the potential areas of face region. The whole process of face region extraction is described by the following images in Figure 14 below;

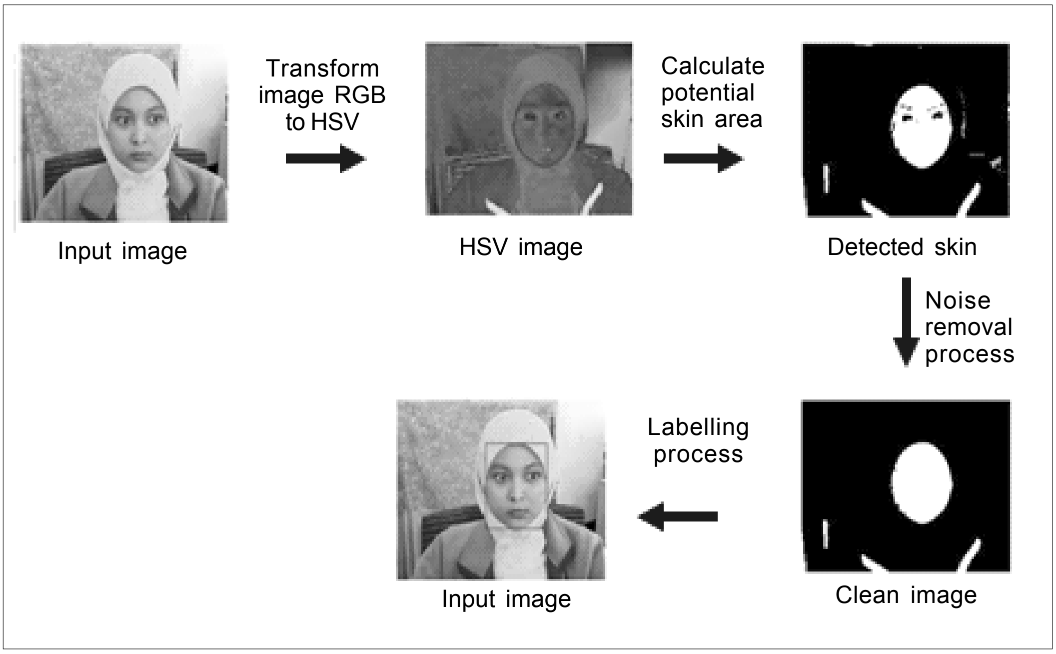

Figure 14 Face region extractions process

Upon the successfully of face region detection, there is some limitation that we need to consider in order to obtained a better detection result. We assume that the persons that use as a sample wearing a different colour hair scarf with their skin and the samples consists of the person who did not wearing spectacles.

4.0 EYE REGION EXTRACTION

Since this part is out of our project, we just adopted the eye region extraction method that produces by Zafer Savas. Given an arbitrary face image, the goal of eye detection is to determine the location of the eyes. In eye detection, the both left and right eyes areas were located and extracted individually as shown in figure below.

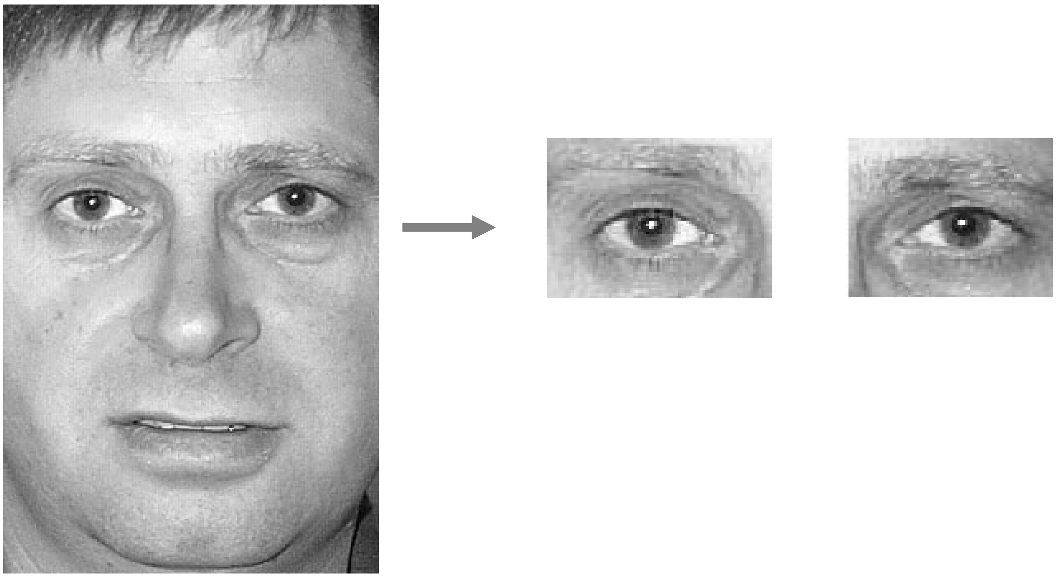

Figure 15 Left and right eyes region from devoted face image

The locations eyes are determined by the following two constraints:

(1) Both eyes should appear in the image and eyes are located below the eyebrows.
(2) Eye regions should be larger than an appropriate size.

As a result of the process, we come out with a model that used as a target to recognize the exact location of the eyes in detected face region as in Figure 16.

Usually eye areas are indicated by a rectangle. Search areas for left end right eyes are defined as in Figure 17.

5.0 IRIS OUTER BOUNDARY AND EDGE POINT DETECTION

The goal of this section is to give detailed information such as the contour of the visible eye region, circular area formed by iris, location of iris in the visible eye area, state of the eye (e.g., blink/not blink). This type of work is more difficult in computer vision area as detection or real-time tracking of small details are highly effected from

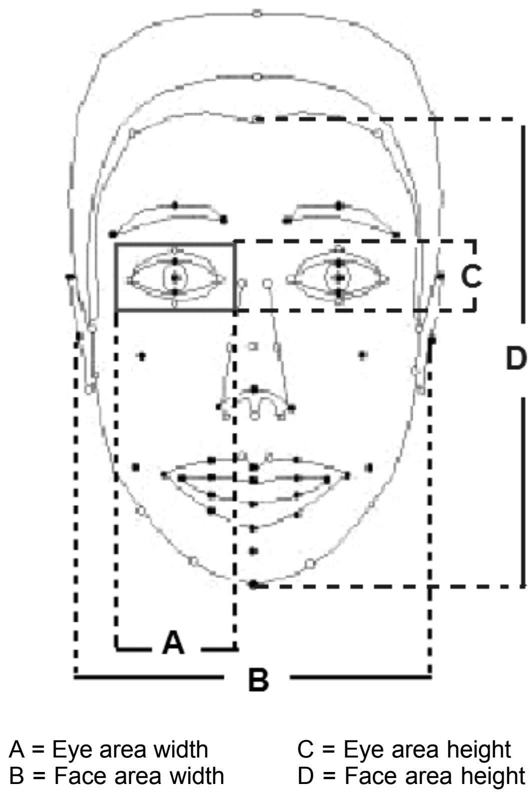

Figure 16 The face model used in eye detection

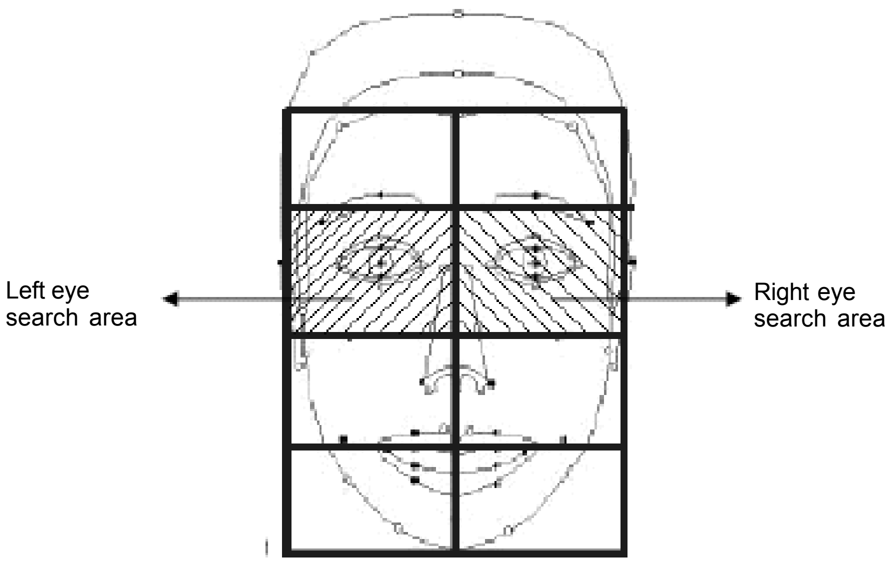

Figure 17 Search areas for left and right eyes

varying ambient conditions and result may easily fail [1]. Finally we proposed a new method to extract edge point based on the image processing method. By using Hough transform algorithm we calculate the probability distribution of every pixel during the conversion process of edge data to get appropriate edge point for further processing. The process of the irises outer boundary and edge point detection is depicted in Figure 18.

A global threshold is selected based on the histogram of the eye region. Global threshold converts a gray scale image into a binary image, which is a special type of gray scale having only two pixels values; black and white, using threshold value(s) for the hold image. In this paper we assume that the eye properties are in the black pixel and then the iris is assumed as the maximum area of the black pixel. Then the pixel is segmented using morphological "dilate" operation with 8×8 structuring connected element which is applied to separate the iris from the other eye properties such as eyelids and eyelashes. After that, remove unwanted pixel which is declared as a noise that will give inappropriate result during the edge detection process. Then continue with edge detection process which involved two steps of detections. The process is to give apparent result for continuing to edge point detection. Based on [4], the process of detecting the edge done with direction 90 degrees which is only in vertical contour

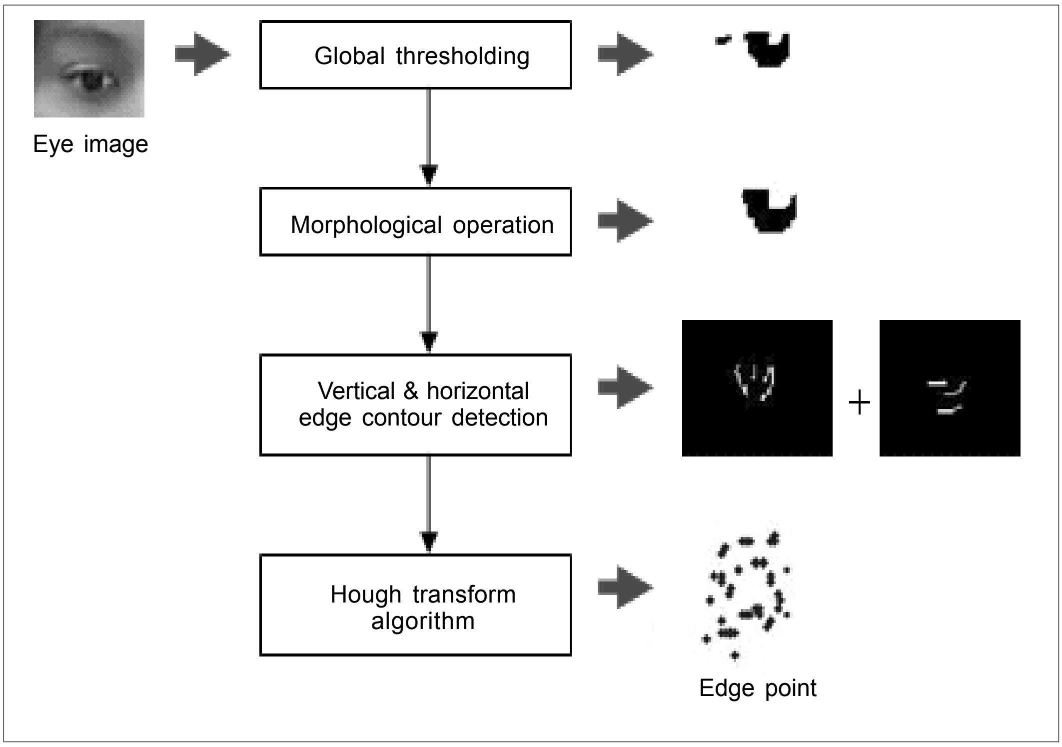

Figure 18 Process of iris outer boundary and edge point detection

of eye and the result shows that it would be difficult to extract the vertical edge of the eye if subjects move their eyeball to left or right position. The edge detection process is continued to the horizontal contour of the eye image to give a clearer result for further processing. Finally we obtained the edge counter pixel result for both the vertical and horizontal boundary. Through Hough transform algorithm we translate the pixel into coordinate point which involved feature extraction technique used in image analysis, computer vision, and digital image processing. By mean of the algorithm we convert edge pixel into a point for circle fitting process different from [4], which used "edge following" technique to extract the edge point.

6.0 IRIS LOCALIZATION

When iris edge points were obtained, we fit a circle to translated data based on an efficient and accurate circle fitting algorithm as drafted in Figure 19. The circle fitting algorithm assumes that the input points rest on a circle and these points are used to estimate the parameters of the circle by solving a least-square fitting problem.

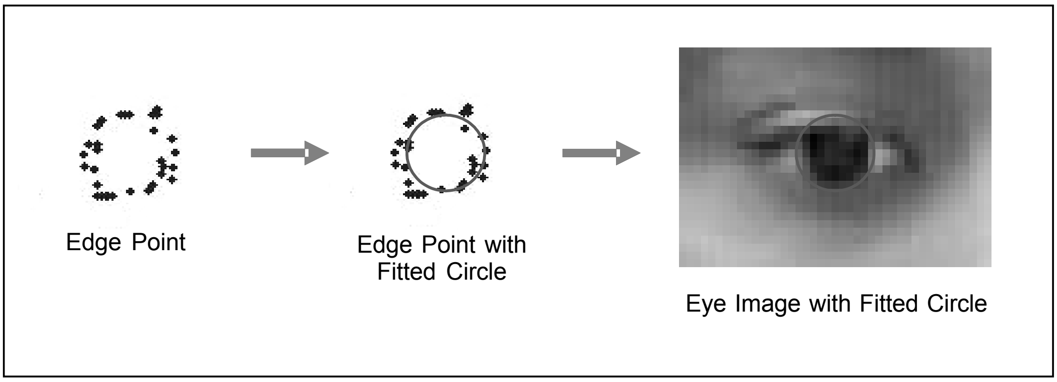

Figure 19 Process of circle fitting

7.0 EXPERIMENTAL RESULT

The system was primarily developed and tested on a Window XP PC with and Intel Core 2 Quad, CPU Q6600 @ 2.40 GHz, and 2.39 GHz, 3.25 GB of RAM. The video was captured with a Logitech QuickCam Pro 5000 webcam at 30 frames per second and processed as RGB images of 320×240 pixels using various utilities from Matlab Image processing program and the system configuration run in an open office environment with a normal illumination conditions.

Reviewing the work done by [8, 9], it is apparent that similar results were obtained with experiments based on testing the accuracy of the system and experiments based on testing the usability of the system as a switch input device. Intuitively, this makes sense as good detector accuracy should yield correspondingly high accuracy results for the usability tests, subject to the user's understanding and capabilities in carrying out the given tasks, such as simple reaction time and matching games, as described by [8, 9]. Therefore, the experiments conducted for this system were more focused on detector accuracy, since this is a more standard measure of overall accuracy of the system across a broad range of users. In order to measure the detection accuracy, test subjects were seated in front of the computer, approximately 2 feet away from the camera. Subjects were instructed to move their eye randomly, but were asked not to turn their head or move too abruptly, since this could potentially lead to repeated initialization of the system, making it difficult to test the accuracy [3].

Based on the above conditions we tested our subjects with the proposed method and, compared the detection accuracy with the eye detection method by [1] which is based on adaptive eigeneyes. Based on Table 1, the result shows that our method detected iris with less error which is 81% of captured frames the result obtained from calculation of frame error rate acquired during recording 2 minutes of real-time video with the detection process. Refer Table 1 and 2 and Figure 20 shows the sample of detection process for both method and based on the detection rate we can see that the accuracy of our method better than adaptive eigeneyes [1].

From Table 1, the results shows the total frames that we used to analyse data from both method is 1800 frames which is contained a front view face images that was capture with random motion and we assume that the motion is only in a small angle

Table 1 Frame error rate result

Method	Total Frame	Detect Face (frame)	Detect 1 Eye (frame)	Detect Both Eye (frame)	Error Frame frame)
Adaptive Eigeneyes	1800	1630	316	254	1400
Proposed Method	1800	1719	150	1405	326

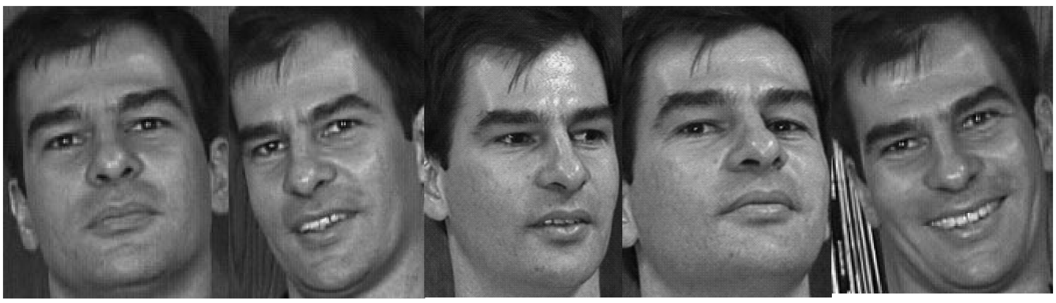

Figure 20 Facial images

as long as the camera can capture the full view of face as a figure below. The images are captured for 25 different persons. From the results at Table 1, we calculate the average of frames that was capture from all samples to analyze the accuracy of detection and to find out the error for both methods. The results represents frames rate of face detection and eye detection. The eye detection results also mention not only about the successfully detection of both eye, there also point out the frames rate of detection either left or right eye. The error frames means the method unable to detect face or eye inside the frames. From the above result we calculate the percentage of detection rate, detection error and computing time from all of the frames from both methods as represent in Table 2. The final results of real-time face detection and iris localization system are shows in Figure 21.

Table 2 Eye tracking result

	Detection Rate (%)	**Detection Error (%)**	**Computing Time (sec)**
Adaptive Eigeneyes	90.6	9.4	20
Proposed Method	97.2	2.8	25

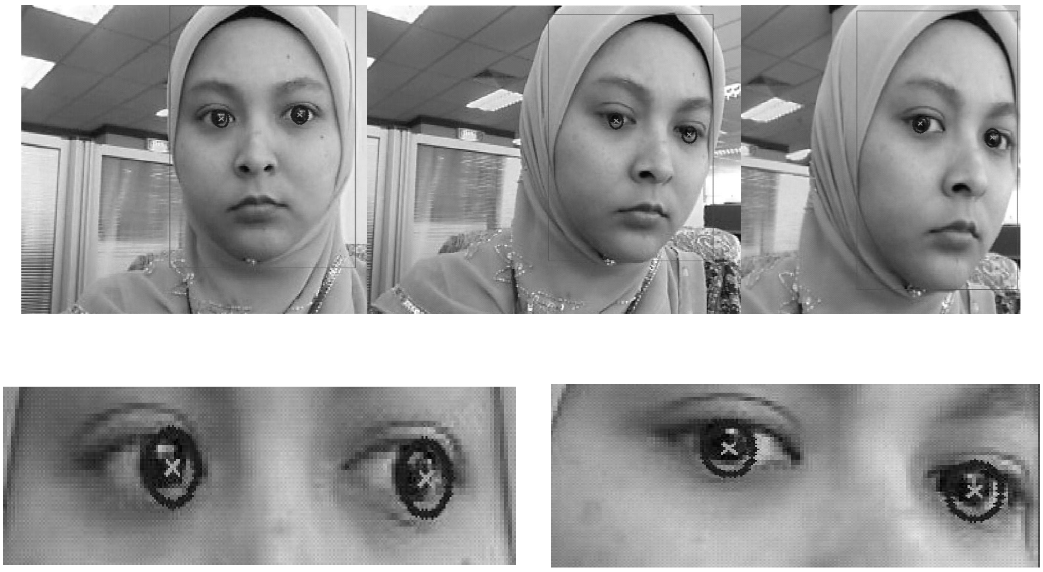

Figure 21 Results of real-time face detection and iris localized

8.0 CONCLUSION

We presented a robust and simple method for detecting and tracking human circular irises. The method employs the common approach technique with different technique which is based on image processing with edge point detection. It produces information on the outer boundary of iris with appropriate radius of the circle as well as additional information on the characteristics of the gradient at the iris boundary. The features were successfully run on real-time tracking of saccadic motion of the eye. The computations time show that it's applicable in a real-time environment.

The system works well with single camera and does not require camera with lens calibrations and able to handle different colour of iris and face. Although the system performance is highly dependent on the computer operating system and face region extraction process, it is easy to make these adjustments.

Future work will concentrate on optimum the detection process and continue with detecting the eyelids of the eye and it will be usable for 3D application. We will also develop a user friendly interface to make the system functional for further investigation in tracking application.

ACKNOWLEDGEMENT

I would like to gratefully thank my supervisor for her advice and suggestions. I am would also like to thank to my Co-supervisor for his invaluable advice, criticism and correction. Not to forget my superior Prof. Dr. Hamid Abdul Rashid for his guidance and caring. I also have to thank my family for their unconditional loving and support. It has been difficult for them to understand the details of what I have been doing and why I have been away from home for so long. Last, but not least, I would like to thank all the staff of the Faculty of Computer Science & IT of Universiti Putra Malaysia for their services and making my time pleasant one. With that, best wishes and good health to all, sincerely from me, and thank you once again.

REFERENCES

[1] Zefer Savas. 2005. Real-Time Detection and Tracking of Human Eye Detecting in Video Sequences. Unpublished Master Thesis of The Graduate School of Natural and Applied Sciences of Middle East Technical University.

[2] Yang, M. H. *et al.* 2004. Detecting Faces in Images: A Survey. *IEEE trans. on PAMI.* 24(1): 34–59.

[3] Hsu, R. L., M. A. Mottaleb and A. K. Jain. 2002. Face Detection in Color Images. *IEEE Trans. Pattern Analysis and Machine Intell.* 24: 696–706,

[4] Wang, J. G., E. Sung and R.Venkateswarlu. 2004. Estimating the Eye Gaze from One Eye. *Computer Vision and Image Understanding*. 98: 83–103.

[5] Fitzgibbon, M. Pilu and R. B. Fisher. 2000. Direct Least Square Fitting of Ellipses. *IEEE trans. on PAMI.* 21(5): 476–480.

[6] Lam, K. and H. Yan. 1996. Locating and Extracting the Eye in Human Face Images. *Pattern Recognition*. 29(5): 771–779.

[7] Zhou, Z. H. and X. Geng 2004. Projection Functions for Eye Detection. *Pattern Recognition.* 37(5): 1049–1056.

[8] Grauman, K., M. Betke, J. Gips and G. Bradski. 2001. Communication via Eye Blinks - Detection and Duration Analysis in Real Time. *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference(CVPR 2001),* Kauai, Hawaii. 2: 1010–1017.

[9] 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.