J. Basic. Appl. Sci. Res., 4(2)222-228, 2014

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Eye Features Detection for Eye Gaze Tracking

[1]Muhammad Shahid and [2]Tabassam Nawaz

[1]Computer Engineering Department, University of Engineering and Technology, Taxila, Pakistan
[2]Software Engineering Department, University of Engineering and Technology, Taxila, Pakistan

Received: August 26 2013
Accepted: October 2 2013

ABSTRACT

Pupil and iris are among the most important eye features for gaze estimation. Success of any eye gaze tracking system is highly dependent on the precise detection of these features. This paper presented a new method for pupil and iris detection. The proposed method took into consideration the structural appearance and intensity distribution of eye region. First, pupil prospect point was detected within the dark pupil region and then horizontal projections and distance parameters were employed to detect pupil and the iris-sclera boundaries. Simple geometric properties were used to measure the pupil and iris centers. Unlike other methods, the proposed method was computationally quite efficient as the processing involved was linear in nature. The method was tested on Chinese Academy of Sciences – Institute of Automation eye database version 1.0 and accuracy of 98.6% and 96.7% for pupil and iris detection respectively had been achieved.

KEYWORDS: Pupil detection, iris detection, eye tracking, gaze estimation, human-computer interaction

1. INTRODUCTION

During the last decade eye gaze tracking has emerged as one of the most important research discipline due to having vast application domains such as human computer interaction, neurological and medical research, marketing research, assistive technology for disabled persons as an alternate mode of interaction as well in automobile industry for vigilance detection and many more. Pupil and iris boundaries detection and center estimation is considered the most essential part for the success of any eye gaze tracking application. Accuracy and computational efficiency are key challenges for their detection. In this paper, we presented a new method for robust and fast pupil and iris boundary detection which works by taking into consideration the physical structure of the pupil and iris edges as well as their photometric appearances.

The rest of the paper is organized as follows; literature review and proposed method are described next which is followed by results and discussion. The paper is finalized with the concluding remarks.

2. LITERATURE REVIEW

Literature offer very rich information on techniques explored for pupil and iris detection. In this section, we will give brief and high level information of the already proposed technique.

The pioneer promising method for pupil and iris detection was proposed by Daugman [1]. The method was based on Integro-differential operator. The operator assumes pupil and limbus as circular contours and consequently performs circular edge detection.

$$max_{(r,x_o,y_o)} \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r,x_o,y_o} \frac{I(x,y)}{2\pi r} ds \right| \quad (1)$$

where $I(x,y)$ represents the source image, $G_\sigma(r)$ is a Gaussian smoothing function. s is the contour of the circle defined by parameters r, x_o, y_o. By varying radius and center coordinate of the circular contour, the operator searches for the circular curve with maximum change in pixel values. Wildes [2] proposed a technique in which detection was based on Hough transform and used circular edge detection technique. Boles et al.[3] utilized gray levels of images for detection. He obtained zero crossings using wavelet transforms for iris representation. Qi-Chaun et al.[4] proposed algorithm which makes use of black pupil to locate center and self-adaptive window for iris detection. This method was quite similar to one used by Wildes. Tisse et al. [5] proposed algorithm which combines the traits of Hough transform and the integro-differential operator to improve the performance of iris detection. Wang et. al.[6] based their study on iris detection for gaze estimation. They applied metrological operator on the

*Corresponding Author: Muhammad Shahid, Computer Engineering Department, University of Engineering and Technology, Taxila, Pakistan.

threshold eye image. They analyzed the vertical edges to select two of the longest edge segments and then fit curves for iris detection. Goni et. al.[7] used adaptive threshold and morphological operations. They then applied blob detection and the most compact blob selected as the pupil. Keil et. al.[8] proposed method revolve around the glints detection within image edge map. Pupil is then searched in the vicinity of the detected glints. Agustin et. al.[9] initially threshold the image, followed by points detection within the region between the pupil and the iris. They later applied RANSAC[10] to fit curves to the detected points. Starburst [11] adapted an iterative approach based on radial features detection. The method cast rays from an initial selected points with 20° radial steps with the assumption that the initial point lies within the pupil boundary. The rays seizes at points where finding high intensity transition expecting the pupil boundary. The method then adapts RANSAC[10] for ellipse fitting.

3. PROPOSED METHOD

Figure1 shows the design flow chart of the proposed method. the input image is first preprocessed. This includes image conversion from RGB color to gray space. Image is resized to reduce computational complexity and improve performance. Pupil prospect point based on photometric appearance of the eye region as characterized by different gray level intensities is determined. Projections initiated from the candidate pupil prospect point are used to detect pupil and iris boundaries. The rest of this section will discuss the method in detail.

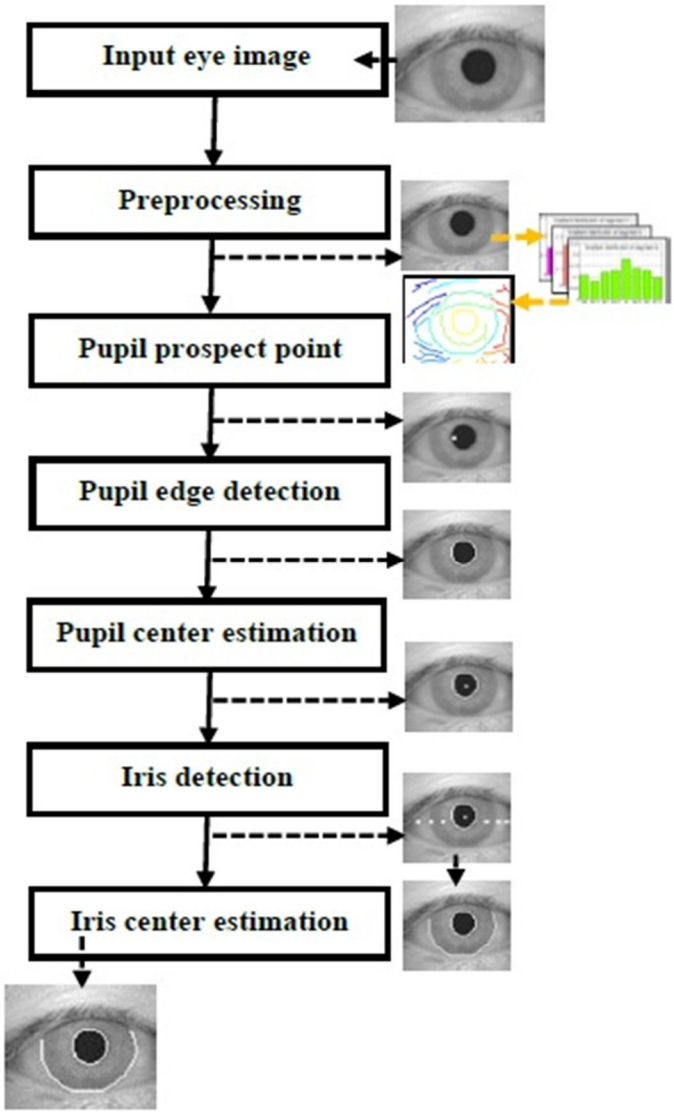


Figure1.Flow chart of the proposed method

A. Image preprocessing

In spite of phenomenon growth in processing capabilities of new machines, processing of multi-dimensional signal such as images for extraction of any meaning information is still computationally a complex task. In order to achieve performance equivalent to typical frame rate of the imaging devices, the images need to be preprocessed such that minimum processor utilization must be required. Based on such consideration, eye images are first preprocessed to convert from RGB color space to gray space and downsize the resultant gray scale images as defined by:

$$G(x,y) = resize[I(x,y) * [M/4, N/4]] \quad (2)$$

where $I(x,y)$ is the input image and $G(x,y)$ is the resized image. M and N are the image dimensions where $M = 320$ and $N = 280$. Canny edge detector is applied on $G(x,y)$ with threshold value of 0.1 to derive edge map, $EdgeMap(x,y)$.

$$f\nabla = F\left(\frac{\partial^2 (G * I)}{\partial n^2}\right) \quad (3)$$

The $EdgeMap(x,y)$ consists of dense edges of varying size and orientation. Among these edges, pupil and iris-sclera edges (hereafter referred as iris edge or boundary) are composed of relatively large connected components. Threshold is defined for edge magnitude. Edges are equated against the defined threshold and those having connected components above the threshold value are considered qualified for pupil and iris edges. Non-qualified edges are prune from the edge map.

$$EdgeMap(x,y) = \begin{Bmatrix} 1 \rightarrow EdgeMap_{EdgeArea} > 10 \\ 0 \rightarrow otherwise \end{Bmatrix} \quad (4)$$

The edges in the resultant $EdgeMap(x,y)$ are labeled to yield $LblEdgeMap(x,y)$.

B. Pupil prospect point detection

Pupil consists of lowest intensity value pixels within the eye region. Eye brows and eye lashes are other regions that contain pixels of similar intensities. The initial step for pupil detection is to find pixel of lowest most intensity value. Image histogram is derived from the image which provides distribution of pixels intensity on 256 gray levels as defined by the following equation.

$$h(r_k) = n_k \quad for\ k = 0,1,2,\ldots,255 \quad (5)$$

where r_k is the k[th] gray level and n_k is the number of pixels in the image having gray level r_k.

Pupil pixels have lowest intensity values and are mostly concentrated within lowest histogram bins. Initial most bins are considered to determine pupil prospect point. Spatial position of pixels having intensity values of initial most bins are examined within the gray image. Considering the fact that pupil region constitutes nearly uniform intensity level pixels, vicinity of each candidate pixel is examined for pupil prospect point. Average neighborhood intensity values of all pixels contained in the initial bins is obtain and cross compared to determine lowest neighborhood intensity value. A neighborhood size of 7 x 7 is used for the purpose. The pixel having the lowest most averaged neighborhood intensity value is considered candidate for pupil prospect point (PPP).

$$Avg(i) = \frac{1}{9}\left[\sum_{x=-1}^{x=1}\sum_{y=-1}^{y=1} G(x,y)\right] \quad (6)$$

$$ProspectPnt = \min(avg(i) \rightarrow i = 1,2,\ldots.) \quad (7)$$

C. Pupil detection

Outward horizontal projections initiated from PPP are drawn within the labeled edge map as shown in figure 2. The projections striking the same labeled edge is considered prospective pupil edge. Based on general observation that pupil is continues near circular boundary. If PPP lies within the pupil region, the projections most likely strikes the same labeled edge. In case the projections fail to detect same edge, the PPP is rejected as prospective candidate. Next lowest averaged pixels are considered in turn as candidate for PPP and assessed on same criteria until the projections strike single edge.

$$PupilPnt1 = match(EdgeMap(x,j),1)\ for\ j = y_{pp}\ to\ n \quad (8)$$

$$PupilPnt2 = match(EdgeMap(x,j),1)\ for\ j = y_{pp}\ to\ 1 \quad (9)$$

Where $EdgeMap$ represent binary image of the subject. Y_{pp} defines y-component of the spatial position of candidate pupil prospect point and $\boldsymbol{n}$ is the total number of pixels in each row.

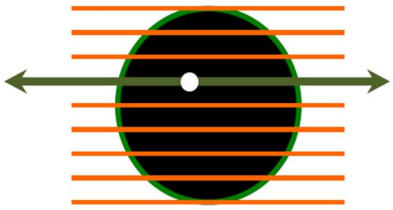


Figure 2.outward horizontal projections from PPP.

The corresponding pixels encompassing pupil boundary within *EdgeMap* are marked in the original image $G(x, y)$. It will result in a closed, continuous and smooth curve of connected components of pupil boundary.

$$G(x,y) = \begin{Bmatrix} 255 \rightarrow LblEdgeMap(x,y) = Label1; \\ G(x,y) \rightarrow otherwise \end{Bmatrix} \quad (10)$$

Where $Label$ is the edge number of the pupil edge found within the edge map.

D. Pupil center estimation

The approach adopted is illustrated in figure 3 and is as under:

- Horizontal distances of the pupil edge pixels are calculated on each pixel row. Row having maximum pixel distance is obtained. Dividing the pixel distance by two will yield the x-component of the center co-ordinate.

$$C_x = \left(Pupil_{LowestCol} + Pupil_{HighestCol}\right)/2 \quad (11)$$

- Vertical distances of the pupil edge pixels are calculated on each pixel column. Column having maximum pixel distance is obtained. Dividing the pixel distance by two will yield the y-component of the center co-ordinate.

$$C_y = \left(Pupil_{LowestRow} + Pupil_{HighestRow}\right)/2 \quad (12)$$

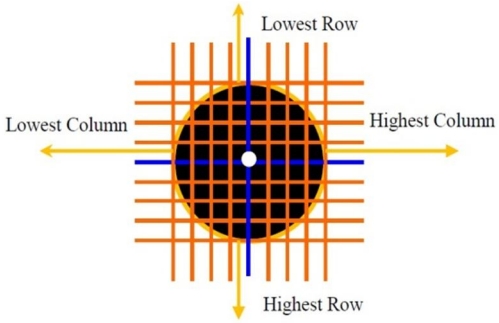


Figure3.illustrates pupil center detection

E. Iris detection

Iris boundary is not as distinct as pupil. A lot of varying size edges appears between the pupil and the iris boundary due to texture of iris. Iris boundary detection needs to be robust enough to distinguish between ordinary edge and the iris edge.

Outward horizontal projections initiated from the pupil edge are drawn in the binary *EdgeMap* . The projections traverse several edges. Edge pixels on the projections are characterized by value 1. Spatial positions of every edge pixel on the projections are accumulated in arrays.

$$IrisPnts1(i) = match(EdgeMap(x,j),1)\ for\ j = PupilPnt1\ to\ n \quad (13)$$
$$IrisPnts2(i) = match(EdgeMap(x,j),1)\ for\ j = PupilPnt2\ to\ 0 \quad (14)$$

Where $IrisPnts1$ and $IrisPnts2$ are the two arrays used to hold y-component of the spatial positions of all edge pixels determined on the left and right projections respectively. $PupilPnt1$ and $PupilPnt2$ are the pupil edge pixels from where the left and right projections are initiated. Spatial distance between every edge pixel and the corresponding pupil edge pixel is determined and recorded in separate arrays as defined by the following equations.

$$Dist1(i) = PupilPnt1_{y-comp} - IrisPnt1(i)_{y-comp} \quad (15)$$

$$Dist2(i) = PupilPnt2_{y-comp} - IrisPnt2(i)_{y-comp} \quad (16)$$

Under neutral forward looking eye postures the iris edge is equidistance from either side of the pupil edge. None of the other edges could resemble this property. Although there might be many edges in the region between the pupil and iris boundaries but there may be hardly any two edges that maintain such geometrical configuration. Besides this, iris boundary has another feature that in most of the cases it can be one complete curve shaped edge having circular contour around pupil and both the outward projections striking the same edge twice – once during each projection. There may be many other edges but they will be of smaller connected components and probably none of them maintain circular contour around the pupil boundary. So even such edges might fulfill the equidistance factor in a rear cases but may not represent complete edge. In addition there is large gradient shift on the iris-sclera boundary due to dark iris and bright sclera. Such geometric and photometric properties are analyzed and adaptive thresholds are employed to determine iris edge.

The first factor to determine iris is to check whether the edge pixels found on both sides belong to the same edge and whether they maintain any geometric relationship between them. The distance maintained by the edge pixels found on left and right-side of pupil boundary is cross compared. Edge pixels on the two projections equidistance from pupil edge are considered as candidate iris edge. An adaptive threshold is used to cater change in the eye position due to shift in gaze direction. To enhance application specific performance of algorithm, distance thresholds are considered for the purpose of comparison. During thorough testing of the algorithm on CASIA database [12], it is observed that the distance between pupil and iris edge is relatively larger than the pupil radius. Unitizing this assumption help reduce complexity involved in the assessment of many edge pixels.

Based on the fact that pupil and iris are not always co-centric, the next step was to find the iris center co-ordinates. The mean location of the two iris edge pixels is computed, as shown below:

$$C_{iris} = \left(IrisEdgePt_{Left} + IrisEdgePt_{Right}\right)/2 \quad (17)$$

The set of pixels on the iris edge seldom characterizes a complete edge. Non-uniform illumination and other affects introduce spurious intensity discontinuities. The algorithm needs procedures to assemble edge pixels into meaningful iris boundary. Since we have obtained the center co-ordinates of the iris boundary as well as boundary edge pixels, a simple method that is employed using this information is drawing circle which works by making use ofiris center co-ordinate and radius. The circle is correlated and adapted to the detected iris boundary pixels for exact positioning.

Finally, the image is to be resized to its original dimension (i.e. 280 x 320 pixels).

$$G(x,y) = resize\left[G(x,y) * [M*4, N*4]\right] \quad (18)$$

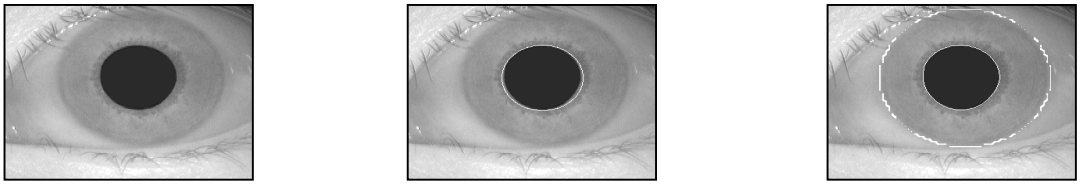
The process of reducing the image size for performance enhancement and then regenerating to original image size results in degradation and loss of pixel characteristics in term of gray level information. To preserve the original information, the spatial position of marked pupil and iris boundaries in image $G(x,y)$ are processed to determine and mark corresponding pixels in the original image $f(x,y)$.

$$f(x,y) = \begin{Bmatrix} 255 \rightarrow G(x,y) = 255; \\ f(x,y) \rightarrow otherwise \end{Bmatrix} \quad (19)$$

4. RESULTS AND DISCUSSION

This research work has focused on accuracy and computational complexity of pupil and iris detection and has proposed a new method for their detection to increase performance and accuracy of eye gaze tracking system. The success of eye gaze tracking system is highly dependent on precise pupil or iris detection. A poor detection can lead to inaccuracy and ultimate failure of the system. Precisely detection of such boundaries, especially, while considering the irregular appearance of pupil boundary due to eye movement was one of the considerations of research to develop a new algorithm. The proposed algorithm takes into consideration that actual pupil boundary is not a perfectly circular in shape. In addition, the appearance varies from near circular shape to elliptical shape with the shifts in eye gaze movement from neutral forward looking direction to left or right direction. The proposed algorithm has successfully detected nearly all the pupil boundaries while detection of iris boundary too has attained very good success results. The second motivation behind the work was to reduce the computation complexity of the algorithm. Most of the presented approaches are computationally too intensive to be used with the standard frame

rate of imaging devices during real-time applications scenarios. The proposed scheme involves linear processing. It takes into account the darkest region strategy, looks for pixels in the pupil boundary and then draws outward projections in image edge map to mark both the boundaries. To verify actual performance of the proposed method, it was experimented on CASIA database [12] version 1.0 and attained detection precision of 98.6% and 96.7% in pupil and iris boundaries. Figure 4 shows some results of our proposed method.



(a) (b) (c)

Figure4. pupil and iris detection experimental results(a) original image (b) pupil detection(c) iris detection

5. CONCLUSION

Precise and computationally efficient pupil and iris detection has been a major challenge in the success of eye gaze tracking and iris recognition system. To improve the quality of such system a novel algorithm has been proposed which address issues related to irregular and variable appearing pupil shapes especially during shift in eye gaze direction as well as computation intensive nature of the prevailing algorithms. Contrary to all the presented algorithms, this approach is based on detection of pupil prospect point within the pupil region and utilizes bi-directional horizontal projections and distance parameters to detect the pupil as well as iris boundaries. The processing involved is linear in nature. Simulation of the proposed algorithm is done in Matlab.

REFERENCES

[1] Daugman, J.,2004. How Iris Recognition Works. IEEE Transactions on Circuits and Systems for Video Technology,14(1): 21-30.

[2] Wildes, R.P., 1997.Iris Recognition: An Emerging Biometric Technology. In the Proceedings of the IEEE, 85(9): 1348-1363.

[3] Boles, W. and B. Boshash,1998. A Human Identification Technique Using Images of the Iris and Wavelet Transform. IEEE Transactions on Signal Processing, 46(4):1185-1188.

[4] Tian,Q.C., Q. Pan, Y.M. Cheng and Q.X. Gao, 2004. Fast Algorithm and Application of Hough Transform in Iris Segmentation. in the Proceedings of 3rd International Conference on Machine Learning and Cybernetics, pp:3977-3980.

[5] Tisse,C.,L. Martin, L. Torres and M. Robert, 2002. Person Identification Technique Using Human Iris Recognition. In the Proceedings of *15th International Conference on Vision Interface*,pp:294-299.

[6] Wang, J.G. and E. Sung, 2002.Study on Eye Gaze Estimation. IEEE Transaction on Systems, Man and Cybernetics, Part B, 32(3):332-350.

[7] Goni, S., J. Echeto, A. Villanueva and R. Cabeza, 2004. Robust Algorithm for Pupil Glint Vector Detection in a Video-Occulography Eye Tracking System. In the Proceedings of International Conference on Pattern Recognition, pp:941–944.

[8] Keil, A., M. Andreas, K. Berger and M.A. Magnor, 2010. Real-Time Gaze Tracking with a Consumer-Grade Video Camera. In the Proceedings of18th International Conference on Computer Graphics, Visualization and Computer Vision, pp: 129-134.

[9] Agustin,J.S.,H. Skovsgaard, E. Mollenbach, M. Barret, M. Tall, D.W. Hansen and J.P. Hansen, 2010. Evaluation of a Low-Cost Open-Source Gaze Tracker. In the Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications, pp: 77-80.

[10] Fischler, M. and R. Bolles, 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM, 24(6): 381-395.

[11] Li, D., D. Winfield and D.J. Parkhurst,2005.Starburst: A Hybrid Algorithm for Video-based Eye Tracking Combining Feature-based and Model-based Approaches. In the Proceedings of IEEE Vision for Human-Computer Interaction Workshop at CVPR, pp:79-87.

[12] Chinese Academy of Sciences–Institute of Automation. Eye Images Database version 1.0.www.sinobiometrics.com