Real-Time Iris Detection

[1]Mohamed Rizon, [2]Chai Tong Yuen, [1]Ali AlMejrad and [3]Naif Alajlan

[1]Department of Biomedical Technology, King Saud University
P.O. Box 10219, Riyadh 11433, Kingdom of Saudi Arabia
[2]Department of Mechatronic & Biomedical Engineering
Universiti Tunku Abdul Rahman (UTAR),
Jalan Genting Kelang, Setapak 53300, Kuala Lumpur, Malaysia
[3]Advanced Lab for Intelligent Systems Research
Department of Computer Engineering
King Saud University, P.O.Box 51178, Riyadh 11543, Kingdom of Saudi Arabia
(mjuhari@ksu.edu.sa)

***Abstract**:* A real-time algorithm to automatically detect human face and irises from color images has been developed. Haar cascade-based algorithm has been applied for simple and fast face detection. The face image is then converted into grayscale image. Three types of image processing techniques have been tested respectively to study its effect on the performance of iris detection algorithm. Then, iris candidates are extracted from the valley of the face region. The iris candidates are paired up and the cost of each possible pairing is computed by a combination of mathematical models. The pairing with the lowest cost is considered as iris. The algorithm has been tested by quality images from Logitech camera and noisy images from Voxx CCD camera. The proposed algorithm has achieved a success rate of 83.60% for iris detection in an open office environment.

***Keywords**:* - Face detection, Iris detection, Illumination normalization, Face recognition

I. INTRODUCTION

In past ten years, face recognition has attracted substantial attention from various disciplines and experienced a tremendous growth in researches. Nowadays, face recognition has been widely applied in security system, surveillance, identity verification and other purposes. Automatic face and facial features detections are vital for face recognition development. Humans are good at face identification but it is never an easy task for a computer to recognize human faces automatically. Automatic real-time face detection systems continue to gain substantial attention in the field of face recognition. Two of such systems have been developed based on still-image [1, 28, 29] and video sequences [30] achieving detection rate of 90% and 99% respectively. These systems used YCbCr or RGB color space in the effort to detect face region. No significant efforts have been reported in dealing with variabilities such as lighting, facial expressions and head tilts.

Generally, face recognition is categorized into holistic method and feature-based method. The holistic approach [3, 4] treats a face as a two dimensional pattern of intensity variation. The feature-based approach [2, 13, 14] recognizes a face using the geometrical measurements [6, 8] taken among facial features. Many methods have been proposed to locate eyes [8, 32], mouth [4, 5], and face region [3, 7] in an image. Popularity of using template matching [2, 3] features detection has been shown at the earlier stage of feature-based research. However, template matching and eigenspace methods require large number of templates for varying poses and face normalization for variation of size and orientation.

Vertical and horizontal projections have been combined with template matching [2, 3], gabor transformation [22], genetic algorithm [7] and the Smallest Univalue Segment Assimilating Nucleus (SUSAN) [14]. SUSAN algorithm has emerged as a simple and accurate method for corner detection either in grayscale image [15, 16] or color image [17, 18, 19]. Anyhow, integral projection might not robust enough to tackle the illumination changes.

In the most recent researches, Adaboost [23] is used to search through the entire face region to get all possible combinations of features based on maximum likelihood. A new facial features localization system has been developed based on Gaussian Mixture Model (GMM) to locate the regions of eyes [24] but it is not fully automated. Most of the studies have been done under controlled environment. Conditions such as beard, moustache, head orientation, hairstyle and facial expressions have been excluded from the face databases used in researches [6, 8, 13, 14, 17, 18, 19].

The aim of this research is to improve the performance of face and iris detection in real-time environment. The proposed algorithm will first locate the face region by using Haar Cascade face detector. Then, illumination normalization is applied to reduce the effect of lighting variation. After that, iris candidates are extracted from the valleys of the face region using feature template and separability filtering. Iris candidates are determined by the total costs computed from Hough Transform, seperability and intensity information. Finally, the pair of irises will be determined by the total costs and matching correlation in a mathematical approach.

II. FACE LOCALIZATION

There are two databases with images captured by using Voxx CCD box camera (Database A) and Logitech Quickcam Pro9000 (Database B) in office environment. Logitech camera has 2.0 megapixels of resolution while Voxx camera has high resolution

480TVL. Both of the cameras have different build qualities and specifications.

In this experiment, OpenCV Haar Cascade based face detector [25] has been used for real-time detection instead of the skin-color based face detector [29]. This is because the RGB color based face detection tends to be affected easily by the lighting variation and complex background in the office. The images of each employee are captured during the face verification process to open their lockers. The images are collected during office hours and no constraint has been set besides facing the camera for verification purpose. Eq. (1) is the main function being used in face detection.

$$cvHaarDetectObjects(a, b, c, d, e, f, g) \quad (1)$$

where (a=image; b=cascade; c=memory storage; d=scale factor; e=min. neighbors; f=flags; g=min. size). The parameters d, e, f, g have been set as 1.1, 3, CV_HAAR_FIND _BIGGEST_OBJECT and 65×65 in this experiment.

III. ILLUMINATION NORMALIZATION

This section describes the chosen image processing technique which is similar to the method proposed by Xiaoyang Tan and Bill Triggs [27] before implementing the iris detection algorithm. It incorporates a series of steps to counter the effects of illumination variations, local shadowing and highlights while preserving the originality of the visual appearance. The first step of this algorithm is gamma correction which is a nonlinear gray-level transformation that replaces gray-level I with I^{γ}, where $\gamma \in [0,1]$ is a user-defined parameter. This can enhance the local dynamic range of the image in dark region while compressing it in bright regions.

. In order to recover object-level information independent of illumination, a power law with the exponent γ in the range of [0, 0.5] is a good compromise to this problem. Difference of Gaussian (DoG) Filtering can be used to reduce the aliasing and noise. The inner Gaussian is typically set to $\sigma_0 \leq 1$ pixel while the outer one might have σ_1 of 2-4 pixels, depending on the spatial frequency at which low frequency information becomes misleading rather than informative. $\sigma_0 = 1.0$ and $\sigma_2 = 2.0$ are used as default setting. Then, two stages of processes as shown in Eq. (2) have been proposed:

$$I(x, y) \leftarrow \frac{I(x, y)}{\left(mean\left(\left|I(x', y')\right|^{a}\right)\right)^{1/a}}$$

$$I(x, y) \leftarrow \frac{I(x, y)}{\left(mean\left(\min\left(\tau, \left|I(x', y')\right|\right)^{a}\right)\right)^{1/a}} \quad (2)$$

Here, α is a strongly compressive exponent that reduces influence of large values, τ is a threshold used to truncate large values after the first phase of normalization and the mean is over the unmasked part of the image. $\alpha = 0.1$ and $\tau = 10$ are used in this experiment. Finally, a nonlinear function is being used to compress over-large values to reduce the influence of those extreme values on subsequent stages of processing. The hyperbolic tangent has been used $I(x, y) \leftarrow \tau \tanh(I(x, y)/\tau)$, thus limiting I to the range $(-\tau, \tau)$.

This algorithm has been tested on its robustness under various lighting conditions. From Fig. 1, the normalized images have shown similarity in terms of illumination distribution even though the input images are taken under different lighting conditions. Besides, LogAbout [26] and histogram equalization have been compared with the proposed normalization method to evaluate the performance of iris detection based on the processed images.

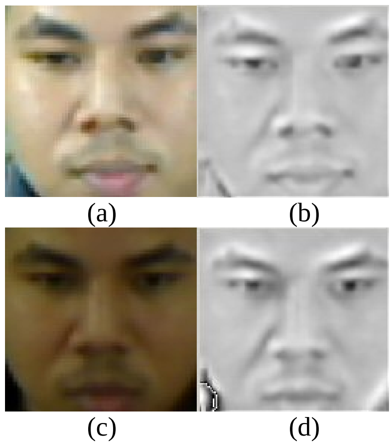

Fig. 1. (a) and (c) are input images; (b) and (d) are the respective output images after illumination normalization

IV. IRIS DETECTION ALGORITHM

The proposed algorithm will first apply grayscale conversion on the input image followed by grayscale closing [12]. Then, illumination normalization and light spot deletion are applied to remove the reflection of lighting on the irises. This can be done by replacing the center pixel of a mask by the smallest intensity value of the pixels within the mask. Histogram equalization [29], LogAbout and illumination normalization have been tested and compared in this experiment to reduce the effect of lighting variation.

1. Valley extraction

After the illumination normalization, valley extraction as shown in Eq. (3) has been carried out. Each pixel (x, y) in the face region:

$$V(x, y) = G(x, y) - I(x, y) \quad (3)$$

where $G(x, y)$ and $I(x, y)$ denote the value obtained from grayscale closing and intensity value. Region which consists of pixels (x, y) such that $V(x, y)$ is greater than or equal to a threshold value are determined to be valleys.

2. Selection of iris candidates

In irises selection, this algorithm performs similar method as proposed in [29]. First, it computes the costs $C(x, y)$ for all pixels in the valleys and selects m pixels according to non-increasing order that give the local maxima of $C(x, y)$ as the iris candidate locations.

$$C(x,y) = C_1(x,y) + C_2(x,y) \quad (4)$$

Where $C_1(x, y)$ is the mean crossing of the row and column pixels and $C_2(x, y)$ is the intensity difference between the center part and the boundary part of a square region. An eye template as shown in Fig. 2 is placed at each candidate location and measures the separability [10] between the two regions R_1 and R_2 given by:

$$\eta = B/A \quad (5)$$

Where $A = \sum_{i=1}^{N}(I(x_i, y_i) - \overline{P}_m)^2$,

$B = n_1(\overline{P}_1 - \overline{P}_m)^2 + n_2(\overline{P}_2 - \overline{P}_m)^2$,

$n_k(k = 1,2)$: number of pixels in R_k,

$N = n_1 + n_2$,

$\overline{P_k}$ $(k = 1,2)$: average intensity in R_k,

$\overline{P_m}$: average intensity in the union of R_1 and R_2,

$I(x_i, y_i)$: the intensity values of pixels (x_i, y_i) in the union of R_1 and R_2.

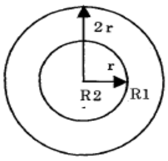

Fig. 2. An eye template to detect blob

Next, it applies Canny edge detector [11] to the face region and measures the fitness of those iris candidates to the edge image using Hough Transform [21]. The vote calculated from Hough Transform is defined as $C_1(i)$. Given an iris candidate $B_i = (x_i, y_i, r_i)$, measures the fitness of iris candidates to the intensity image by placing two templates similar as Fig. 2. Compute the separabilities $\eta_{23}(i)$, $\eta_{24}(i)$, $\eta_{25}(i)$ and $\eta_{26}(i)$ using Eq. (5) for $C_2(i)$ and $C_3(i)$. $C_4(i)$ is the ratio of average intensity of each iris candidate over average intensity of all the candidates. Then the cost for each iris candidate is calculated as below:

$$C(i) = C_1(i) + C_2(i) + C_3(i) + C_4(i) \quad (6)$$

Where $C_1(i) = \frac{V_{max}}{V(i)}$

$C_2(i) = \frac{|\eta_{23}(i) - \eta_{24}(i)|}{\eta_{23}(i) + \eta_{24}(i)}$

$C_3(i) = \frac{|\eta_{25}(i) - \eta_{26}(i)|}{\eta_{25}(i) + \eta_{26}(i)}$

$C_4(i) = \frac{U(i)}{U_{av}}$

where $V(i)$ is the vote for B_i given by Hough transform; V_{max}, the maximum of $V(i)$ over all iris candidates; $U(i)$ is the average intensity inside B_i and U_{av} is the average of $U(i)$ over all iris candidates. Finally, computes the cost for each pair of iris candidates B_i and B_j:

$$F(i,j) = t\{C(i) + C(j)\} + (1-t)/R(i,j) \quad (7)$$

Where $C(i)$ and $C(j)$ are costs computed by Eq. (6). $R(i, j)$ is the normalized cross-correlation value computed between eye template and the input image. The eye template used in this experiment is shown in Fig. 3 t is the weight to adjust two terms of the cost. The lowest cost $F(i, j)$ is selected as the pair of irises as shown in Fig. 5.

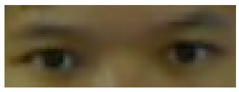

Fig. 3. The eye template used for computation of $R(i, j)$

V. RESULT AND DISCUSSION

Experiments have been done to evaluate the real time performance of the proposed iris detection algorithm with histogram equalization, LogAbout and illumination Normalization. A total of 109 and 122 images captured by Voxx and Logitech cameras are collected for this experiment under office environment. One image from Voxx camera and three images from Logitech camera have been eliminated from the databases since one of the iris is not appeared in the images. This research intends to make the system user-friendly whereby users have not been limited in wearing spectacles, type of clothes, hairstyles, facial expressions, multiple faces and background. Here, Database A represents the images captured by Voxx camera while Database B represents the images captured by Logitech camera.

From Table 1, histogram equalization, LogAbout and illumination normalization scored 38.53%, 54.13% and 68.81% respectively for database A for success detection. The success rates of iris detection for database B are 62.30% (histogram equalization), 71.31% (LogAbout) and 83.60% (illumination normalization). Thus, Illumination normalization has achieved the highest iris detection rate in both databases at real-time followed by LogAbout and histogram equalization. This means the images processed by illumination normalization have better contrast and stability when dealing with lighting variation. However, all of the processing techniques do not show outstanding results when dealing with database A which are noisier and blur images.

Table 1. Iris detection rates for different image processing techniques for Database A and Database B

Processing Technique	Database A (%)	Database B (%)
Histogram Equalization	38.53	62.30
LogAbout	54.13	71.31
Illumination Normalization	68.81	83.60

The proposed processing technique has improved the performance of iris detection more than 30% (Database A) and 21% (Database B) compared to the histogram equalization which was proposed in previous research. Although the improvement rate for Database A is greater but the overall success rate is still considered low for all the tested processing methods with the highest detection rate of 68.81%. Anyhow, illumination normalization outperforms LogAbout to nearly 15% when processing the low quality and noisy images from Database A.

Successful iris detections from both databases are shown in Fig. 4. This has covered a variety of lighting conditions, genders, backgrounds, spectacles, expressions and hairstyles. The improved iris detection algorithm has shown its robustness through this experiment especially when facing issues such as complex backgrounds and multiple faces. However, there are also some failures reported. The examples of the fail detection are shown in Fig. 5. Most of the subjects in those fail detections are wearing spectacles. The proposed algorithm has not been successful when dealing with certain types of spectacles which have some levels of lens reflectance when expose to lighting. This explains why there are also successful iris detections for the same subjects under different lighting conditions.

By studying the fail cases, frame of a spectacle can cause the iris detection algorithm to fail in some situations. This tends to happen when the subject wears the spectacle slightly lower than its normal position. The upper frame will block or appear at the same line as the irises horizontally. This has confused the proposed algorithm in making the correct decision especially when the color of the frame is black which has similar intensity values with the iris. Besides, spectacle with low transparency affects the visibility of iris. This becomes worse in darker lighting condition, low quality image and the subject has some distances from the camera. Another possibility that can cause fail detection is the appearance of irises.

Some facial expressions of the subjects have caused the eyes to be nearly closed and indirectly causing the irises not visible enough for detection. One of the failure case in Fig. 5 shows that the thick moustache of a subject under such lighting condition might confuse the detection algorithm. Solutions in terms of mathematical model or rules need to be set to differentiate the features which have similar intensity value as irises. The integration of illumination normalization into the iris detection algorithm replacing histogram equalization has improved the performance of the proposed algorithm in real-time. Besides, the ability of this algorithm to deal with the appearance of spectacles has been increased.

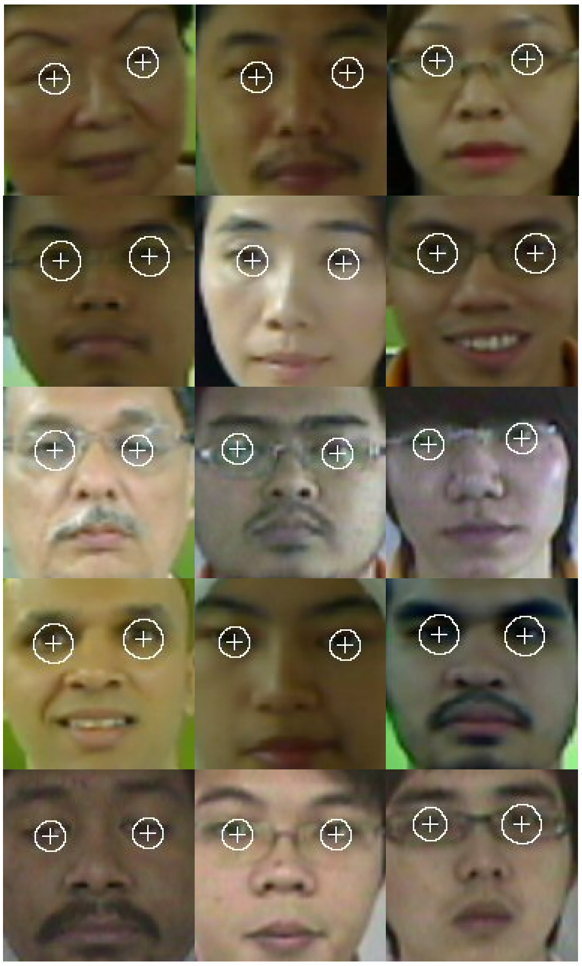

Fig. 4. Successful iris detections

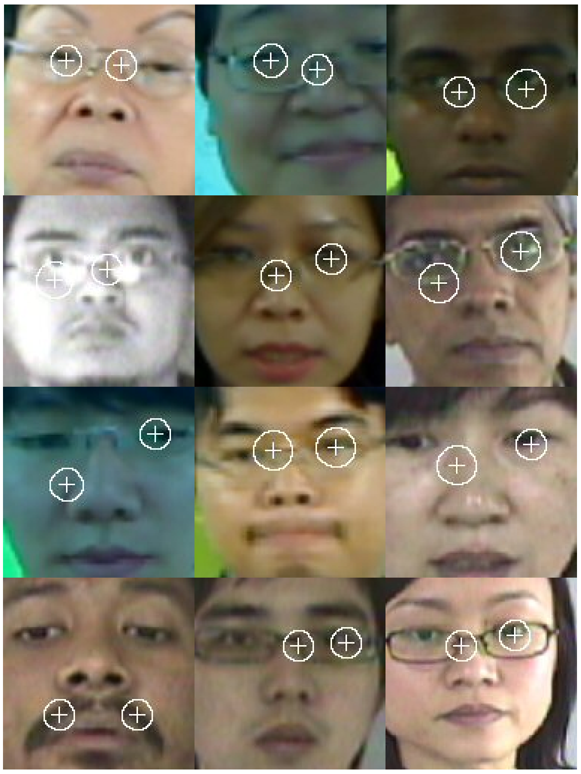

Fig. 5. Fail iris detections

VI. CONCLUSION

The experimental result has proven that the performance of the iris detection algorithm can be improved more tan 30% by applying a better image processing technique. Illumination normalization has proven its ability to cope with lighting variation under office environment. The proposed iris detection algorithm has achieved a success rate of 83.60% in a user-friendly real-time environment. Through this experiment, the algorithm has shown its robustness when tested with subjects wearing different kind of spectacles. In future, the iris detection algorithm needs to be enhanced in order to perform better under conditions like lens reflectance, facial expressions, disruption of moustache and excessive lighting. Finally, official online databases will be used to benchmark and evaluate the performance of the proposed algorithm in near future.

REFERENCES

[1] Tsuyoshi K and Mohamed R (2003), Iris detection using intensity and edge information, Pattern Recognition 36:549-562.

[2] Brunelli R and Poggio T (1993), Face recognition: features versus templates, IEEE Trans. Pattern Anal. Mach. Intell. 15 (10):1042-1052.

[3] Beymer DJ (1994), Face recognition under varying pose, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, Washington:756-761.

[4] Li CM, Li YS, Zhuang QD and Xiao ZZ (2004), The face localization and regional features extraction, Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai: 26-29.

[5] Karin S and Ioannis P (1998), A novel method for automatic face segmentation, facial feature extraction and tracking, Signal Processing: Image Communication 12:263-281.

[6] Selin B, Bulut MM and Volkan A (2002), Projection based method for segmentation of human face and its evaluation, Pattern Recognition Letters 23:1623-1629.

[7] Wong KW, Lam KM and Siu WC (2001), An efficient algorithm for human face detection and facial feature extraction under different conditions, Pattern Recognition 34.

[8] Ryu YS and Oh SY (2001), Automatic extraction of eye and mouth fields from a face image using eigenfeatures and multilayer perceptrons, Pattern Recognition 34:2459-2466.

[9] Lin CH and Wu JL (1999), Automatic facial feature extraction by genetic algorithms, IEEE Trans. Image Process. 8 (6):834-845.

[10] Fukui K and Yamaguchi O (1997), Facial feature point extraction method based on combination of shape extraction and pattern matching, Trans. IEICE Japan J80-D-II (8):2170-2177.

[11] Canny J (1986), A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6):679-698.

[12] Sternberg SR (1986), Grayscale morphology, Computer Vision Graphics Image Process. 35:333-355.

[13] Hua G, Guangda S and Cheng D (2003), Feature points extraction from faces, Image and Vision Computing NZ, Palmerston North.

[14] Yu S, Kun H, Zhou J, Liu Z and Kui L (2004), Multi-resolution feature extraction in human face, Proceedings of International Conference on Information Acquisition.

[15] Stephen MS and Michael BJ (1997), SUSAN – A new approach to low level image processin, International Journal of Computer Vision 23(1):45-78.

[16] Mauricio H and Geovanni M, Facial feature extraction based on the Smallest Univalue Segment Assimilating Nucleus (SUSAN) algorithm, Image Processing and Computer Vision Research Lab (IPCV-LAB), Escuela de Ingenieria Electrica, Universidad de Costa Rica.

[17] Yulu Q, Goenawan B and Nuanwan S (1998), Finding the estimated position of facial features on the human face using intensity computation, IEEE Asia-Pacific Conference (APCCAS):579-582.

[18] Vladimir V (2002), Face and facial feature tracking for natural human-computer interface, International Conference Graphicon, Nizhny Novgorod, Russia.

[19] Jie Y and Alex W (1996), A real-time face tracker, Application of Computer Vision, WACV'96:142-147.

[20] Martinez AM and Benaventa R (1998), The AR face database, CVC Technical Report, No.24.

[21] Tsuyoshi K, Daisuke H, Mohamed R, Detection of eyes from human faces by hough transform and separability filter, Department of Computer Science and Intelligence Systems, Oita University, Oita, Japan:870-1192.

[22] Zhao Y, Shen X and Georganas ND (2008), Combining integral projection and gabor transformation for automatic facial feature detection and extraction, Proceedings of IEEE

International Workshop on Haptic Audio Visual Environment and their Applications:103-107.

[23] Erukhimov V and Lee KC (2008), A bottom-up framework for robust facial feature detection, Proceedings of 8th International on Automatic Face and Gesture Recognition:1-5.

[24] Kumatani K, Ekenel HK, Gao H, Rainer S, Aytul E (2008), Multi-stream gaussian mixture model based facial feature localization, Proceedings of Signal Processing, Communications and Applications Conference:1-4.

[25] Viola P and Jones MJ (2004), Robust real-time face detection, International Journal of Computer Vision 57(2):137-154.

[26] Liu H, Gao W, Miao J and Li J (2002), A novel method to compensate variety of illumination in face detection, Joint Conference on Information Sciences:692-695.

[27] Tan X and Bill T (2007), Enhanced local texture feature sets for face recognition under difficult lighting conditions, IEEE International Workshop on Analysis and Modelling of Faces and Gestures:235-249.

[28] Chew WJ, Ang LM and Seng KP (2008), Automatic model based face feature detection system, Proceedings of International Symposium on Information Technology:1-6.

[29] Chai TY, Mohamed R, Woo SS and M Sugisaka (2008), Automated detection of face and facial features, The 7th WSEAS Int. Conference on Signal Processing, Robotics and Automation, University of Cambridge, United Kingdom:230-234.

[30] Balasubramaniam M, Palanivel S, and Ramalingam V (2008), Real time face and mouth Recognition using radial basis function neural networks, Expert Systems with Applications:1-8.