A New Accurate Technique for Iris Boundary Detection

MAHBOUBEH SHAMSI, PUTEH BT SAAD, SUBARIAH BT IBRAHIM, ABDOLREZA RASOULI, NAZEEMA BT ABDULRAHIM
Faculty of Computer Science & Information System
University Technology Malaysia
Skudai, Johor, Malaysia - 81310
mahboubehshamsi@yahoo.com, drputeh@gmail.com, subariah@utm.my, rs.reza@gmail.com, iema_830619@yahoo.com

Abstract: - The Iris based identification systems have been noticed exceedingly, presently. In this process, the iris should be segmented from captured eye image. The low contrast between pupil and iris, usually, will harden the segmenting process and will decrease the accuracy of detecting boundary between them. In order to segment the iris more precise, we propose a new technique using a difference function and a factor matrix. We also modify the segmentation operator to detect the pupil and iris as ellipse instead of circle. Experiments show that the proposed technique can segment the iris region and pupil region precisely. Based on our result, 99.34% of eyes have been segmented accurately in 1.24s averagely.

Key-Words: - Daugman's method, Average Square Shrinking, Difference Function, Contour Factor Matrix

1 Introduction

Iris is the most reliable biometric in secure transaction proposals. Iris in an eye image is situated between sclera and pupil. Before iris can be utilized for a specific application, it has to be localized first. Iris localization is a challenging ordeal due to several reasons such as occlusions to the presence of eyelids and eyebrows and also due to the uneven textural contrast. The textural contrast between sclera and iris is high; conversely the texture contrast between iris and pupil is low. The problem is further aggravated with the presence of light reflection in the pupil. Hence, an accurate algorithm is desired to detect the subtle difference between the two regions. In this paper, we propose a new algorithm to capture a maximum difference value of both, inner and outer iris boundaries.

This paper next describes related work, followed by a brief description of Daugman's Operator. We fully explain the proposed algorithm in section 2 in three phases: Ellipse Function, Improving Difference Function and Normalization. The paper then continues on result in section 3. The paper ends with a conclusion in section 4.

1.1 Related Work

Iris is between sclera and pupil. Sclera region and iris region is easier to differentiate due to higher texture contour contrast between the two regions. However, there is a lower texture contrast between iris region and pupil region. Hence, it is difficult to automatically detect the edge between iris and pupil [1].

Daugman [2] uses a differential operator for locating the circular iris, pupil regions and the arcs of the upper and lower eyelids. The differential operator is defined as

$$\max_{(r,x_0,y_0)}\left|G_\delta(r)*\frac{\partial}{\partial_r}\oint_{r,x_o,y_0}\frac{I(x,y)}{2\pi r}d_s\right| \quad (1)$$

Where $I(x, y)$ is the gray level of image in pixel (x, y), $G_{\delta(r)}$ is Gaussian smoothing filter, s is the counter of circle represented by center, (x_0, y_0) and r is the radius. The operator searches for the circular path where there is maximum change in pixel values, by varying the radius and centre x and y position of the circular contour. However, if there is noise in the image, such as reflection, the algorithm can fail. When the iris is dark and the image is under natural light, the contrast between iris and pupil is low and it makes the segmentation process more difficult [3].

An automatic segmentation algorithm based on the circular Hough transform is employed by Wildes *et al.* [4, 5]. They also make use of the parabolic Hough transform to detect the eyelids, approximating the upper and lower eyelids with parabolic arcs, which are represented as;

$$\begin{aligned}&\left(-\left(x-h_j\right)\sin\theta_j+\left(y-k_j\right)\cos\theta_j\right)^2\\&\qquad=a_j\left(\left(x-h_j\right)\cos\theta_j+\left(y-k_j\right)\sin\theta_j\right)\end{aligned}\tag{2}$$

Where a_j is the control of the curvature, (h_j, k_j) is the peak of the parabola and θ_j is the angle of rotation relative to the x axis. For edge detection in this method we need to choose threshold values.

Camus and Wildes [6] use similar method to Daugman's method [7]. Their algorithm, finds three circumference parameters (centre (x, y) and radius z) by maximizing the following function

$$C=\sum_{\theta=1}^{n}\left((n-1)\left\|g_{\theta,r}\right\|-\sum_{\phi=\theta+1}^{n}\left\|g_{\theta,r}-g_{\phi,r}\right\|-\frac{I_{\theta,r}}{n}\right)\tag{3}$$

Where n is the total number of directions and $I_{\theta,r}$ and $g_{\theta,r}$ are the image intensity and derivatives with respect to the radius in the polar coordinate system respectively.

The performance of algorithm degraded for noisy iris images and when there is reflection in the image.

Yuen *et al.* [8] developed computational algorithm to automatically detect human face and irises from color images captured by real-time camera. Kocer [9] proposed a neural network based iris recognition approach by analyzing iris patterns. They applied Multi-Layered Perceptron (MLP) and Modular Neural Networks (MNN) to the iris feature vector for classifying the iris images. El-Taweel [10] proposed a novel and efficient approach to iris feature extraction using a set of filters, called difference-of-sum filters. These filters can take advantage of a pre-computed integral image, which makes the filtering process take constant computation time no matter how big the filters are. Fenghua [11] proposed an iris recognition method based on multi algorithmic fusion. The proposed method combines the phase information based algorithm and zero-crossing representation based algorithm at the matching score level. The fusion rule is based on support vector machine (SVM) is applied to generate a fused score which is used to make the final decision.

Normalization is the next step of iris recognition which in this step the segmented iris will be transformed to polar coordinates from Cartesian coordinates. The Wildes *et al.* [5] system employs an image registration technique. This system uses the geometrically warps of acquired image $I_a(x, y)$ with a database that include selected image $I_d(x, y)$. For closing the corresponding points of new image with reference image, we need the mapping function. The mapping function $(u(x, y), v(x, y))$ to transform the original coordinates is:

$$\int_x\int_y\left(I_d(x,y)-I_a(x-u,v-y)\right)^2 d_x d_y\tag{4}$$

While being constrained to capture a similarity transformation of image coordinates (x, y) to (x', y'), that is

$$\begin{pmatrix}x'\\y'\end{pmatrix}=\begin{pmatrix}x\\y\end{pmatrix}-sR(\phi)\begin{pmatrix}x\\y\end{pmatrix}\tag{5}$$

Where s is the scaling factor and $R(\phi)$ is the matrix representing rotation by ϕ.

We describe the Daugman's operator as based operator of this paper, in next section.

1.2 Daugman's Operator

Daugman's operator is based on the fact that the illumination difference between inside and outside of pixels in iris edge circle is maximum [12]. It means the difference values of pixel's gray level in iris circle are higher than any other circles in image. This fact is based on color of iris and color of sclera. Sclera is the white area outside of iris which is shown in Fig 1.

We are not able to calculate Daugman Operator for all feasible circles of an image. Therefore, we should restrict the space of potential circles. Many researchers assume that the center of iris is near the center of image. But in many cases, the center of iris does not fit to the center of image. Also they find a range of radius which is based on the size of image.

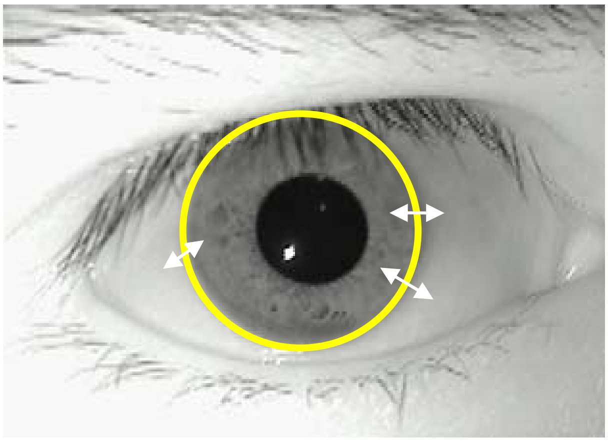


Fig 1. Difference between inside and outside of iris edge

In our first paper, we proposed a new Average Square Shrinking (ASS) Approach [13] for initializing the range of potential centers. Therefore, we restricted the algorithm to a range of (x, y) as potential centers and a range of r as potential radiuses. The ASS approach is based on this fact that all eyes' pupils are black and therefore the center of iris must be black. Gaussian Blur or other smoothing method is applied to find dark integrated pixels in image processing methods.

Therefore we used the darkest place to detect the center of the iris. We had broken the image to some small squares. Each square in a source image will be converted into one pixel in shrunken image during ASS process. The size of square and the stages of shrinking are related to shrinking (smoothing) factor S_f and number of shrinking stages N respectively. The values of all pixels inside the square will be averaged in the shrunken image. The darkest pixel (x_0, y_0) in the last image is the pupil center. The range of $[x_0 \pm S_f] \times [y0 \pm S_f]$ will be used for potential centers of Daugman operator.

Base on Daugman's Operator, the difference of all pixels on circle contour should be calculated and this was clearly impossible. So we should adjust an upper limit circle sample (CS) which assesses how many points should be accessed on circle contour to compute Daugman's integral. Higher value for CS clearly will reduce the estimated error of computing Daugman's integral.

We had converted the Daugman's environmental integral to a discreet summation of a simple difference function on circle's contour to be computable by computer programming [14]. The difference function is

$$diff(x, y) = I(x + \Delta_\alpha, y + \Delta_\alpha) - I(x - \Delta_\alpha, y - \Delta_\alpha) \quad (6)$$

We showed that the difference function also should be computed approximately, because we had only integer values for coordinates (x, y) and we could not calculate exact difference values. We had only the values of top, bottom, left, right and diagonal pixels. Due to this, we converted the Daugman operator as follows:

$$\begin{aligned}
&\max_{(x_c, y_c, r)} \sum_{j=1}^{CS} diff(x_j, y_j) \\
&for\ all (x_c, y_c) \in potential\ centers \,\&\, r \in potential\ radius \\
&x_j = x_c + r \cdot \cos(\alpha_j), y_j = y_c + r \cdot \sin(\alpha_j), \alpha_j = 2\pi * {}^{j}\!/_{CS} \\
&diff(x_j, y_j) = I_1 + I_2 + I_3 + I_4 \\
&I_1 = (I(x_j + 1, y) - I(x_j - 1, y)) \cdot \cos(\alpha_j) \\
&I_2 = (I(x, y_j + 1) - I(x, y_j - 1)) \cdot \sin(\alpha_j) \\
&I_3 = (I(x_j + 1, y_j + 1) - I(x_j - 1, y_j - 1)) \cdot \sin(45 + \alpha_j) \\
&I_4 = (I(x_j + 1, y_j - 1) - I(x_j - 1, y_j + 1)) \cdot \cos(45 + \alpha_j)
\end{aligned} \quad (7)$$

However, our previous method worked very well [15], but it did not compute the difference value correctly. The performance of our algorithm degraded for low texture contrast eye's image.

In next section, we proposed a new difference function to improve the algorithm's detectability.

2 Proposed Algorithm

We use the Average Square Shrinking Process to find the potential centers and the estimated range of radius. Then, we apply the Daugman operator for improving the iris center and radius.

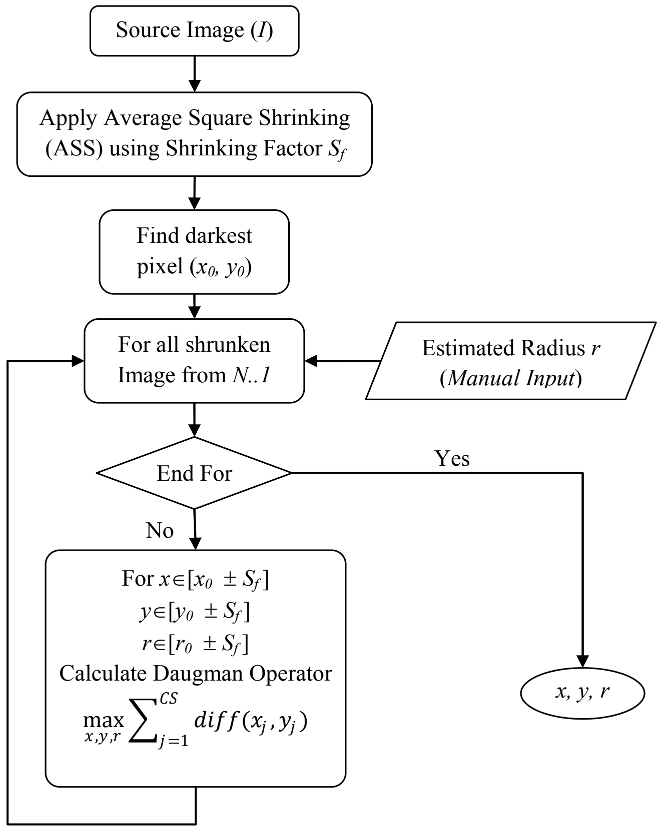


Fig 2. Proposed Algorithm

The higher value of Daugman operator correspondes to the exact center and radius of iris. These steps will be continued iteratively on higher shrunk image to find the final center and radius. The flow chart of the algorithm is illustrated in Fig 2.

2.1 Ellipse Segmentation

The experiment was conducted on many eyes' images. It has been observed that so many irises are not exactly circle. So an ellipse view will improve the accuracy of iris segmentation. We have made a slight modification to Daugman's operator to make it desirable for ellipse function. The new operator is:

$$\max_{(r_a,r_b,x_0,y_0)} \left| G_\delta(r) * \frac{\partial}{\partial_r} \oint_{r_a,r_b,x_0,y_0} \frac{I(x,y)}{2\pi\sqrt{\frac{1}{2}\left(r_a^2+r_b^2\right)}}\, d_s \right| \quad (8)$$

Two parameters r_a, r_b have been replaced by radius parameter r. The new operator is able to look for any feasible ellipse around the iris to find the best match. Regarding to this new operator, the discreet version will be as follows:

$$\max_{(x_c,y_c,r_a,r_b)} \sum_{j=1}^{CS} diff\left(x_j, y_j\right)$$
$$for\ all\left(x_c, y_c\right) \in potential\ centers\ \&\ r_a, r_b \in potential\ radius$$
$$x_j = x_c + r_a \cdot \cos(\alpha_j),\ y_j = y_c + r_b \cdot \sin(\alpha_j),\ \alpha_j = 2\pi * {}^{j}\!/_{CS}$$
$$diff\left(x_j, y_j\right) = I_1 + I_2 + I_3 + I_4$$
$$I_1 = \left(I\left(x_j+1, y\right) - I\left(x_j-1, y\right)\right) \cdot \cos\left(\alpha_j\right)$$
$$I_2 = \left(I\left(x, y_j+1\right) - I\left(x, y_j-1\right)\right) \cdot \sin\left(\alpha_j\right)$$
$$I_3 = \left(I\left(x_j+1, y_j+1\right) - I\left(x_j-1, y_j-1\right)\right) \cdot \sin\left(45+\alpha_j\right)$$
$$I_4 = \left(I\left(x_j+1, y_j-1\right) - I\left(x_j-1, y_j+1\right)\right) \cdot \cos\left(45+\alpha_j\right)$$
(9)

In order to add the new radius parameter, processing time will be increased. Let us explain the problem by semi codes. The circle operator could be implementing by three loops as:

```
For each xc in [possible x as potential center]
  For each yc in [possible y as potential center]
    For each r in [possible x as potential radius]
      Try the operator on Circle (xc, yc, r)
```

Whereas, in ellipse operator the loop's number will be increase to four loops as:

```
For each xc in [possible x as potential center]
  For each yc in [possible y as potential center]
    For each ra in [possible x as potential center]
      For each rb in [possible x as potential radius]
        Try the operator on Ellipse (xc, yc, ra, rb)
```

It means that the order of algorithm will be increased to $O(n^4)$ from $O(n^3)$ and that will raise the processing time. According to Computational Theory, all algorithms with $O(n^4)$ are time consuming. But the range of our variables is utmost image's width. This range is not big enough to give us problem. However, the time will increase, but algorithm will finish in reasonable time. Our result shows that adding ellipse parameter does not raise the processing time, significantly.

2.2 Improving Difference Function

The Daugman Operator tries to compute the difference gray level value between inside and outside pixels of iris circle. Daugman demonstrated that this value is the greatest one between all feasible circles of an eye image [16].

There are discrepancies in the difference function created in our first algorithm as described next. We have eight pixels around a contour pixel as shown in Fig 3. We represent these pixels by $P_1..P_8$. The difference function computes the difference value by subtracting the opposite pixels. These opposite pixels are right and left pixels (P_1, P_5), top and bottom (P_3, P_7), and two pair of diagonal pixels (P_2, P_6), (P_8, P_4). The function ignores the pixel on circle's contour since the outside and inside pixels of circle's contour are the same. P_1, P_2, P_3 and P_8 are assumed as outside pixels and P_5, P_6, P_7 and P_4 are inside, respectively.

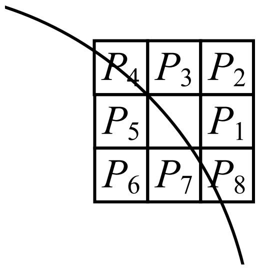


Fig 3. Outside and Inside Pixels of Iris

Fig 4 shows the problem of this assumption that should be removed. The outside and inside pixels of difference function are colored in Fig 4. The gray rectangles are outside pixels set and hatched rectangles are inside pixels. Let us highlight the problem by an example. If you focus on $Octad_5$, the gray rectangles are really inside the circle, whereas they are assumed as outside pixels in difference function.

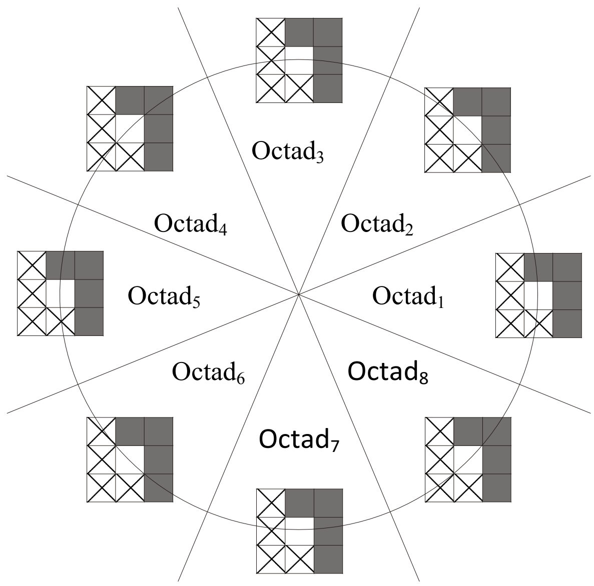


Fig 4. The Wrong Pixel Setting

The correct adjustment of outside and inside pixels around the circle's contour is shown in Fig 5.

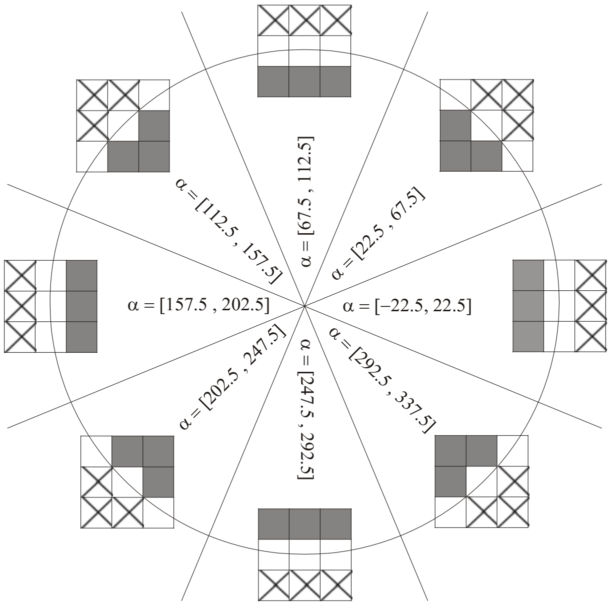


Fig 5. Correct Pixel Setting

We have only eight pixels around the main pixel. So we divide the circle to eight regions, because the outside and inside states of pixel in each region are the same. This fact is shown in Fig 5. Each region covers 2π/8 or π/4 which is equal to 45°. First the center degree of each region is computed, because each region corresponds to its center. The center of each region is represented as θ_j with 22.5° offset as follows:

$$\theta_j = \left(\left\lfloor \frac{\alpha_j + 22.5}{45} \right\rfloor \times 45 \right) \tag{10}$$

We proposed a new factor matrix with 3×3 dimension. The matrix is multiplied by the neighbors of the main pixel correspondingly. The factor matrix values are selected from (-1, 0, 1). The multiplication factors for outside pixel is 1 and for contour and inside pixels are 0 and -1, respectively. By adding the values of pixels after applying the factors, we obtain the real difference between outside and inside pixels. The factor matrix is:

$$M = \begin{bmatrix} \sin\left((\theta_j - 45)\times\left(\frac{2\pi}{360}\right)\right) & \sin\left((\theta_j)\times\left(\frac{2\pi}{360}\right)\right) & \sin\left((\theta_j + 45)\times\left(\frac{2\pi}{360}\right)\right) \\ \sin\left((\theta_j - 90)\times\left(\frac{2\pi}{360}\right)\right) & 0 & \sin\left((\theta_j + 90)\times\left(\frac{2\pi}{360}\right)\right) \\ \sin\left((\theta_j - 135)\times\left(\frac{2\pi}{360}\right)\right) & \sin\left((-\theta_j)\times\left(\frac{2\pi}{360}\right)\right) & \sin\left((\theta_j + 135)\times\left(\frac{2\pi}{360}\right)\right) \end{bmatrix} \tag{11}$$

Some examples of factor matrix are shown below.

$$\alpha = [-22.5, 22.5] \Rightarrow M = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\alpha = [\ 22.5, 67.5] \Rightarrow M = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$$

$$\alpha = [67.5, 112.5] \Rightarrow M = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, \ldots$$

The factor matrix ensures the correct setting for outside and inside pixels as shown in Fig 5. So the new difference function, *diff*(*x, y*) is:

$$\max_{(x_c, y_c, r_a, r_b)} \sum_{j=1}^{CS} diff(x_j, y_j) \quad for\ all (x_c, y_c) \in potential\ centers\ \&\ r_a, r_b \in potential\ radius$$

$$x_j = x_c + r_a \cdot \cos(\alpha_j), y_j = y_c + r_b \cdot \sin(\alpha_j), \alpha_j = 360 * {}^{j}\!/_{CS}, \theta_j = \left(\left\lfloor \frac{\alpha_j + 22.5}{45} \right\rfloor \times 45 \right)$$

$$diff(x_j, y_j) = \left| \sum_{\tilde{i}=1}^{3} \sum_{\tilde{j}=1}^{3} \tilde{I}_{\tilde{i},\tilde{j}} \right| \tag{12}$$

$$\tilde{I} = \begin{bmatrix} \sin\left((\theta_j - 45)\times\left(\frac{2\pi}{360}\right)\right) & \sin\left((\theta_j)\times\left(\frac{2\pi}{360}\right)\right) & \sin\left((\theta_j + 45)\times\left(\frac{2\pi}{360}\right)\right) \\ \sin\left((\theta_j - 90)\times\left(\frac{2\pi}{360}\right)\right) & 0 & \sin\left((\theta_j + 90)\times\left(\frac{2\pi}{360}\right)\right) \\ \sin\left((\theta_j - 135)\times\left(\frac{2\pi}{360}\right)\right) & \sin\left((-\theta_j)\times\left(\frac{2\pi}{360}\right)\right) & \sin\left((\theta_j + 135)\times\left(\frac{2\pi}{360}\right)\right) \end{bmatrix} \circ \begin{bmatrix} I(x_j - 1, y_j - 1) & I(x_j, y_j - 1) & I(x_j + 1, y_j - 1) \\ I(x_j - 1, y_j) & I(x_j, y_j) & I(x_j + 1, y_j) \\ I(x_j - 1, y_j + 1) & I(x_j, y_j + 1) & I(x_j + 1, y_j + 1) \end{bmatrix}$$

Let us to continue the example.

$$\alpha = [-22.5, 22.5] \quad \Rightarrow M = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \Rightarrow \widetilde{I} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \circ \begin{bmatrix} P_4 & P_3 & P_2 \\ P_5 & P_{x,y} & P_1 \\ P_6 & P_7 & P_8 \end{bmatrix} \Rightarrow diff(x,y) = P_1 + P_2 + P_8 - P_4 - P_5 - P_6$$

$$\alpha = [-202.5, 247.5] \Rightarrow M = \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix} \Rightarrow \widetilde{I} = \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix} \circ \begin{bmatrix} P_4 & P_3 & P_2 \\ P_5 & P_{x,y} & P_1 \\ P_6 & P_7 & P_8 \end{bmatrix} \Rightarrow diff(x,y) = P_5 + P_6 + P_7 - P_1 - P_2 - P_3$$

We delete the number of stages parameter N that is required to be input manually. This is accomplished by adopting a threshold for the size of the last shrunken image. The image is shrunk until the size of image will be less than the threshold. For example, if we select the threshold as 10 pixels, then the size of last shrunken image is less than 10×10 pixels. It means that we continue the shrinking until the size of shrunken image becomes less than 10 pixels. Consequently, the range of radius parameter can also be ignored, since in an eye image with width 10 pixels, the radius can be estimated by 2 pixels.

2.3 Normalization

In the normalization, iris region is transformed, so that it has fixed dimensions to allow comparisons between the same iris images. The inconsistencies between the same eye images are due to stretches of the iris caused by dilation of pupil from different illumination. Among other factors that cause dilation are: eye rotation, camera rotation, head tilt and varying image distance. The good normalization process must produce different iris regions for different iris, in the same condition and it must produce constant dimensions for the same iris in different condition. Another great challenge is that the pupil region is not always concentric within the iris region, and is usually slightly nasal [17]. Daugman's rubber sheet model [7] explains remap of each iris region's point to the polar coordinates (r,θ). r is moved in distance [0, 1] and θ is moved in angle $[0, 2\pi]$. The Transform is shown in Fig 6.

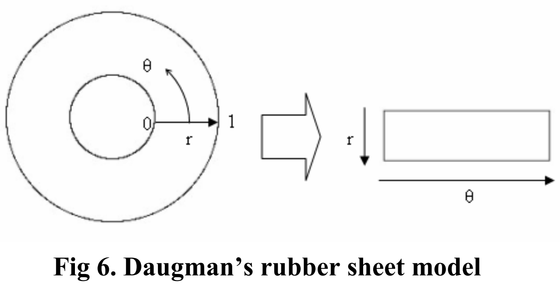


Fig 6. Daugman's rubber sheet model

The remapping of the iris region from (x, y) Cartesian coordinates to the normalized non-concentric polar representation is modeled as

$$I(x(r,\theta), y(r,\theta)) \rightarrow I(r,\theta) \tag{13}$$

With

$$\begin{aligned} x(r,\theta) &= (1-r)x_p(\theta) + rx_l(\theta) \\ y(r,\theta) &= (1-r)y_p(\theta) + ry_l(\theta) \end{aligned} \tag{14}$$

The coordinates of the pupil and iris boundaries along the θ direction are x_p, y_p and x_l, y_l. The rubber sheet model is useful for accounting pupil dilation and size inconsistencies. This model however does not compensate for rotational inconsistencies.
Two iris templates are aligned with matching in shifting the iris templates in the θ direction to solve this problem.
Our algorithm on normalization of iris regions is based on Daugman's rubber sheet model [7]. Since the pupil can be non-concentric to the iris, a remapping formula is needed to rescale points depending on the angle around the circle. This is given by

$$r' = \sqrt{\alpha}\beta \pm \sqrt{\alpha\beta^2 - \alpha - r_I^2} \tag{15}$$

With

$$\begin{aligned} \alpha &= o_x^2 + o_y^2 \\ \beta &= \cos\left(\pi - arcton\left(\frac{o_y}{o_x}\right) - \theta\right) \end{aligned} \tag{16}$$

The displacement of the centre of the pupil relative to the centre of the iris is given by o_y, o_x. The distance between the edge of pupil and the edge of iris at an angle θ is r' and the radius of the iris is r_I.
In order to prevent non-iris region data from corrupting the normalized representation, data points which occur along the pupil border or the iris border are discarded same as Daugman's rubber sheet model.

3 Experimental Result

The algorithm is developed using Delphi programming language. It is tested on 2.4 GHz CPU with Windows Vista and 2 GB Ram. Two famous iris databases have been selected for experiments.

CASIA-IrisV3 [18] includes three subsets which are labeled as CASIA-IrisV3-Interval, CASIA-IrisV3-Lamp, CASIA-IrisV3-Twins. CASIA-IrisV3 contains a total of 22,051 iris images from more than 700 subjects. All iris images are 8 bit gray-level JPEG files, collected under near infrared illumination. MMU iris database [19] contributes a total number of 450 iris images. Subjects come from Asia, Middle East, Africa and Europe. Each of them contributes 5 iris images for each eye.

The maximum value of difference function for edge between sclera and iris (outer boundary) and for edge between iris and pupil (inner boundary), are computed using the previous algorithm and the current algorithm. The results are depicted in Figures 7 - 8. The number of circle contour sampling (*CS*) adopted is 128 samples.

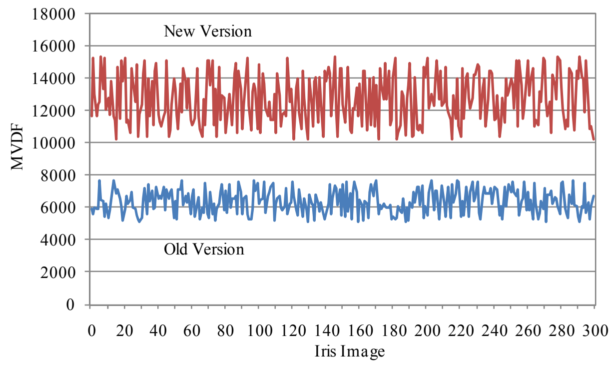


Fig 7. Maximum Value of Difference Function for Outer Boundary

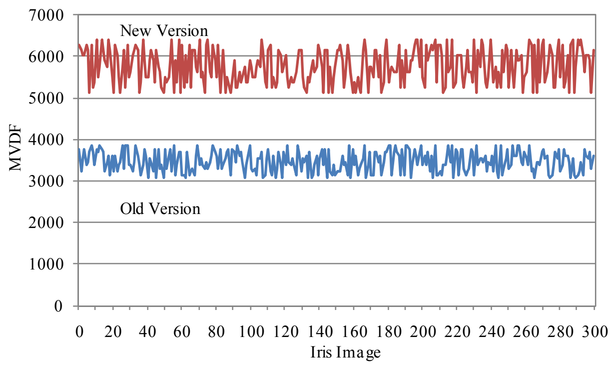


Fig 8. Maximum Value of Difference Function for Inner Boundary

It is noticed that the maximum difference value is doubly increased in the new algorithm for both outer and inner iris boundaries when compared to our previous algorithm. The maximum difference value is bigger for outer boundaries as compared to inner boundaries. This is due to the high texture contrast between sclera and iris. Conversely, the small difference value for inner boundary is due to the low texture contrast between iris and pupil.

In addition, we investigate the effect of circle sample (*CS*) on the maximum value of difference function. The results are illustrated in Fig 9, 10. It is noticed that, as the circle sample increases, so does the maximum value of difference function. Hence, the circle sample is relatively proportional to maximum value of difference function. The results show that with the low value of *CS* both algorithms display a similar performance and there is no significant variation between difference values. However, by increasing *CS,* the distinction of difference value on the new algorithm will be extremely improved.

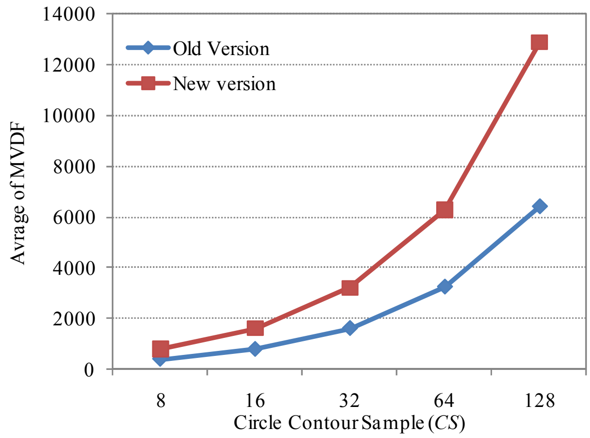


Fig 9. Maximum Value Related to CS for Iris Circle

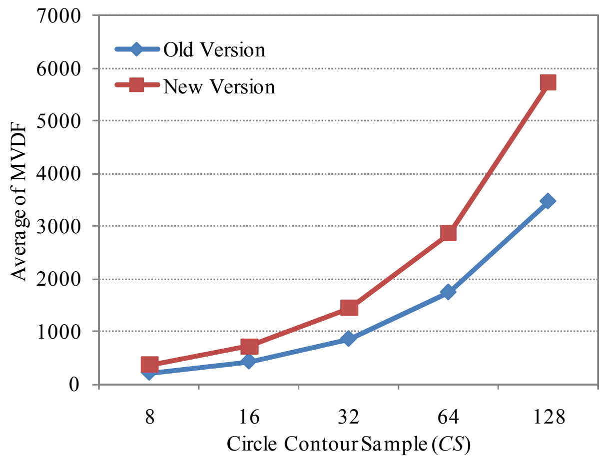


Fig 10. Maximum Value Related to CS for Pupil Circle

The amount of circle contour sample (*CS*) versus detection accuracy and time consuming has been studied in [13]. The correct value of *CS* decreases the manual inference of user. The result of this investigation is shown in Fig 11.

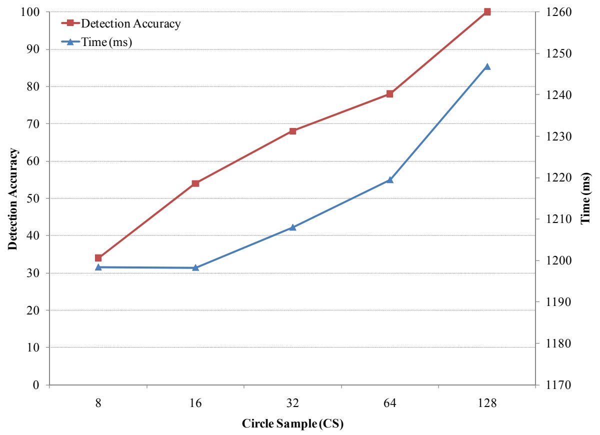
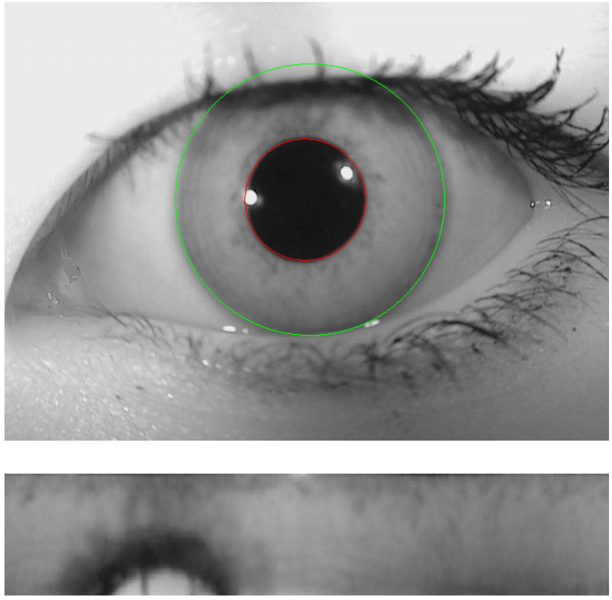


Fig 11. Detection Accuracy and Time vs. Increasing Circle Sample

The results show that increasing *CS* is strongly related to the raising of accuracy and also it affect strongly on processing time. It is clear that a long processing time for boundaries detection is undesirable for real life application. It is observed that in Fig 11 the processing time difference between small value of *CS* and higher value of CS is only 1.24s, thus it convinced the practicability of the algorithm.

3.1 Normalization Result

The normalization process proved to be successful and some results are shown in Fig 12. However, the normalization process is unable to perfectly reconstruct the same pattern from images with varying amounts of pupil dilation, since deformation of the iris results in small changes of its surface patterns.



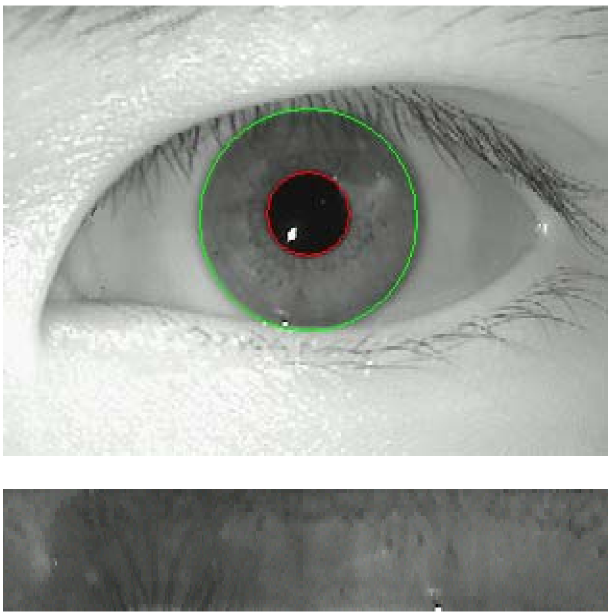


Fig 12. Illustration of the normalization process for two images of the same iris taken under varying conditions.

Normalization of two eye images of the same iris is shown in Fig 12. The pupil is smaller in the bottom image, however the normalization process is able to rescale the iris region so that it has constant dimension. In this example, the rectangular representation is constructed from (360 * (Iris Radius – Pupil Radius)) data points in each iris region. Note that rotational inconsistencies have not been accounted for by the normalization process, and the two normalized patterns are slightly misaligned in the horizontal (angular) direction. Rotational inconsistencies will be accounted for in the matching stage.

3.2 Ellipse Operator Result

The result of ellipse operator is shown in Fig 13. It is disclosed that the ellipse operator redounded to better segmentation and accuracy is extremely increased. But, due to increasing one more radius parameter, we expect that the processing time increases, respectively. So we compare the performance and processing time of our new algorithm with other algorithms obtained from [20]. Table 1 shows the boundary detection rate of various algorithms in comparison to our algorithm. With concern to result, although, the processing time earned by proposed algorithm is not the best one, but with relation to its accuracy, it is the best segmentation algorithm. It also takes no long as usual.

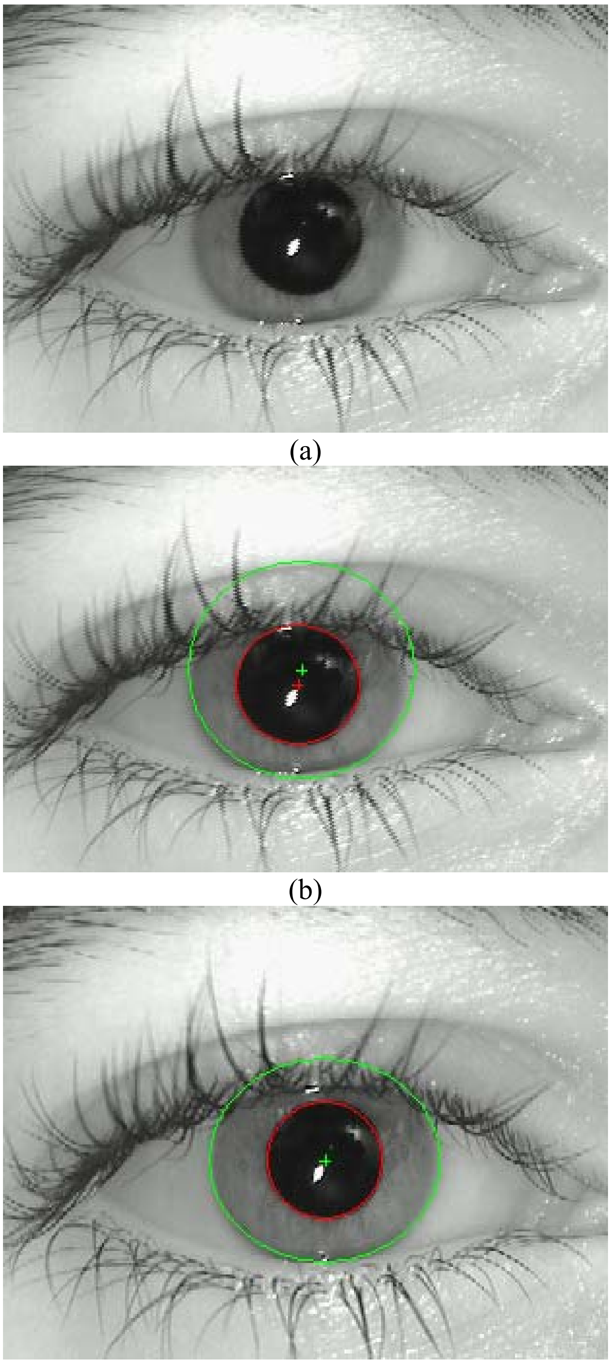


Fig 13. a) The Original image – b) The result of Circle operator - c) The result of Ellipse operator

Table 1. Detection Rate of Proposed Algorithms

Algorithm	Time (s)	Detect Rate
Daugman [21]	5.36	98.58%
Daugman [22]	0.984	54.44%
Wildes [21]	6.34	99.82%
Wildes [22]	1.35	86.49%
Masek [22]	7.5	83.92%
Proposed Algorithm	1.24	99.34%

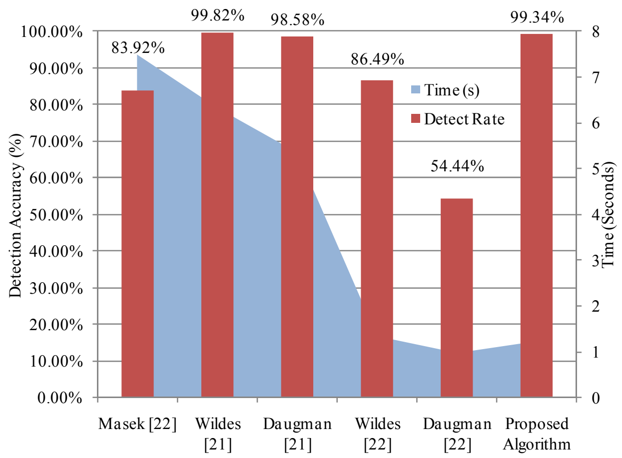


Fig 14. Detection Accuracy vs. Processing Time

With regards to Table 1 as represented in Fig 14, it is noticed that our new algorithm shows the highest rate of iris boundaries detection related to its execution time in comparison to existing algorithms. Based on these findings, we are confident that the subsequent step of feature extraction will aid us to produce good quality textural features for further analysis.

4 Conclusion

The detection of iris boundaries is a difficult task in iris recognition systems. This is due to the low texture contrast between pupil region and iris region. In this paper, we hybrid summation function and factor matrix to be able to detect the iris boundaries. Both theoretically and our experimental results show that the proposed new algorithm strongly improved the Daugman Operator Difference Function and the detection accuracy has been significantly improved by new algorithm. The segmentation rate using this algorithm is 99.34%.
In future, we shall improve the texture contrast on iris boundaries by applying the high contrast image processing technique to obtain better results.

Acknowledgment

This project is supported by Ministry of Science, Technology and Innovation (MOSTI) through the E-Science Fund grant titled Iris Cryptography for Identity Document.

References:

[1]. Daugman, J.G., *Biometric personal identification system based on iris analysis.* United States Patent, 1994. **5**: p. 560.

[2]. Mazur, J., *Fast Algorithm for Iris Detection.* Advances in Biometrics, 2007: p. 858-867.

[3]. Daugman, J.G., *How iris recognition works.* Proceedings of 2002 International Conference on Image Processing, 2002.

[4]. Wildes, R., *Iris recognition: an emerging biometric technology.* Proceedings of the IEEE, 1997. **85**(9).

[5]. Wildes, R., J. Asmuth, G. Green, and S. Hsu, *A system for automated iris recognition.* Proceedings IEEE Workshop on Applications of Computer Vision, 2004.

[6]. Camus, T.A. and R. Wildes, *Reliable and fast eye finding in closeup images.* IEEE 16th Int. Conf. on Pattern Recognition, Quebec, Canada, 2004: p. 389-394.

[7]. Daugman, J.G., R. Anderson, and F. Hao, *Combining Crypto with Biometrics Effectively.* IEEE Trans. on Computers, 2006. **55**(9).

[8]. Yuen, C.T., M. Rizon, and M. Shazri, *Real-Time Detection of Face and Iris.* WSEAS TRANSACTIONS on SIGNAL PROCESSING, 2009. **5**(6).

[9]. Kocer, H.E. and A. Novruz, *An efficient iris recognition system based on modular neural networks*, in *Proceedings of the 9th WSEAS International Conference on Neural Networks*. 2008: Sofia, Bulgaria.

[10]. El-taweel, G.S. and A.K. Helmy, *Efficient iris recognition scheme based on difference of filters.* WSEAS Trans. Comp. Res., 2008. **3**(3): p. 152-161.

[11]. Fenghua, W., H. Jiuqiang, and Y. Xianghua, *Iris recognition based on multialgorithmic fusion.* WSEAS Trans. Info. Sci. and App., 2007. **4**(12): p. 1415-1421.

[12]. Shamsi, M., P.B. Saad, and A. Rasouli, *Iris Image Localization using Binning Approach.* Proceedings of 2008 Student Conference on Research and Development (SCOReD 2008), 2008.

[13]. Shamsi, M., P.B. Saad, S.B. Ibrahim, and A. Rasouli, *Fast Algorithm for Iris Localization Using Daugman Circular Integro Differential Operator.* International Conference of Soft Computing and Pattern Recognition, SoCPaR09, 2009.

[14]. Shamsi, M., P.B. Saad, and A. Rasouli, *Creation of a Unique Private Crypto Key from Iris Image.* Proceeding of Postgraduate Annual Research Seminar, UTM, 2009.

[15]. Shamsi, M., P.B. Saad, S.B. Ibrahim, and A. Rasouli. *Iris Boundary Detection Using A Novel Algorithm*. in *The 9th WSEAS International Conference on Applications of Computer Engineering (ACE'10) - (ISI, Elsevier Indexed)*. 2010. Penang, Malaysia: WSEAS.

[16]. Shamsi, M., P.B. Saad, and A. Rasouli, *Iris Segmentation and Normalization Approach.* Journal of Information Technology (Universiti Technologi Malaysia), 2008. **19**(2).

[17]. Jen-Chun, L., P.S. Huang, C. Chien-Ping, and T. Te-Ming. *Novel and Fast Approach for Iris Location*. in *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2007*. 2007.

[18]. *CASIA iris image database*, in *http://www.sinobiometrics.com*, Chinese Academy of Sciences Institute of Automation.

[19]. *MMU iris image database,* , in *http://pesona.mmu.edu.my/~ccteo*, Multimedia University.

[20]. Proenca, H. and L.A. Alexandre, *Iris segmentation methodology for non-cooperative recognition.* IEEE Proceedings on Vision, Image and Signal Processing, 2006. **153**(2): p. 199-205.

[21]. Nguyen Van, H. and K. Hakil, *A Novel Circle Detection Method for Iris Segmentation*, in *Proceedings of the 2008 Congress on Image and Signal Processing, Vol. 3 - Volume 03*. 2008, IEEE Computer Society.

[22]. Narote, S.P., A.S. Narote, L.M. Waghmare, and A.N. Gaikwad. *An Automated Segmentation Method For Iris Recognition*. in *TENCON 2006. 2006 IEEE Region 10 Conference*. 2006.