International Journal of Medical, Medicine and Health Sciences
ISSN: 2517-9969
Vol:1, No:5, 2007

A Novel Approach to Iris Localization for Iris Biometric Processing

Somnath Dey, and Debasis Samanta

Abstract—Iris-based biometric system is gaining its importance in several applications. However, processing of iris biometric is a challenging and time consuming task. Detection of iris part in an eye image poses a number of challenges such as, inferior image quality, occlusion of eyelids and eyelashes etc. Due to these problems it is not possible to achieve 100% accuracy rate in any iris-based biometric authentication systems. Further, iris detection is a computationally intensive task in the overall iris biometric processing. In this paper, we address these two problems and propose a technique to localize iris part efficiently and accurately. We propose scaling and color level transform followed by thresholding, finding pupil boundary points for pupil boundary detection and dilation, thresholding, vertical edge detection and removal of unnecessary edges present in the eye images for iris boundary detection. Scaling reduces the search space significantly and intensity level transform is helpful for image thresholding. Experimental results show that our approach is comparable with the existing approaches. Following our approach it is possible to detect iris part with 95-99% accuracy as substantiated by our experiments on CASIA Ver-3.0, ICE 2005, UBIRIS, Bath and MMU iris image databases.

Keywords—Iris recognition, iris localization, biometrics, image processing.

I. Introduction

THE recent advances in information technology and increasing emphasis on security have resulted in more attention to automatic personal identification system based on biometrics. Biometric technology is an automated method for recognizing an individual based on physiological or behavioral characteristics. Among the present biometric traits, iris is found to be the most reliable and accurate [1] due to the rich texture of iris patterns. The human iris is an annular part between the pupil and the white sclera (see Figure 1). The iris has distinct charasteristics such as freckles, coronas. stripes, furrows, crypts, and so on. Each eye cotains unique iris pattern that is stable throughout ones life. These characteristics make it attractive for used as a biometric feature to identify individuals.

In iris biometric system, an important task is to extract iris feature from a given eye image. Human iris recognition process is basically divided into two phases. The phase, which is dealt with the extraction of iris features from an eye image and store them into database is called the "enrollment process". At the time of matching we capture the iris features of a human and compare it with the stored features, which is called the "matching process". Each of the above phases are complex and hence is divided into several sub tasks. Figure 2 shows the different steps involved in the two phases. First four tasks in both the phases are common as it is evident in Figure 2. The task namely, "matching features" is extra in the matching process. Out of the several tasks involved in the iris recognition process it has observed that the iris localization is the most computationally intensive task.

S. Dey and D. Samanta are with School of Information Technology, Indian Institute of Technology Kharagpur, India. e-mail:somnath_dey2003@yahoo.co.in and dsamanta@iitkgp.ac.in

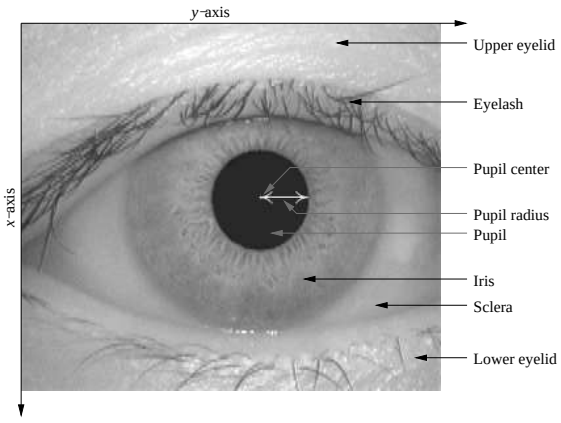

Fig. 1. The typical components in an eye image.

In iris localization task, we try to locate iris part in an eye image. Iris part localization is necessary to isolate the iris part of the image in between the iris boundary (between sclera and iris) and outside the pupil. This task mainly consists of two sub tasks: detecting pupil boundary (between pupil and iris), and detecting iris boundary (between iris and sclera). There are several methods for detecting the iris part from am eye image. Integro-differential operator is used in several work [1], [2] for detecting pupil and iris boundary. Another method called Hough transform [3] is adopted in some work [4], [5], [6], [7], [8], [9]. Further, Laplacian of Gaussian (LOG) [10], active countour model [11] and Gaussian mixture model (GMM) [12] are also known for iris boundary detection. All these methods use total image for locating the iris boundary, although approaches are different. Existing methods are computationally expensive. Further, existing methods may lead to false results when images are of inferior quality due to noise such as, occlusion of eyelid boundaries, light reflection, non-uniform illumination and low contrast between iris and sclera part etc. Another drawback in the existing approaches is that pupil and iris are assumed as concentric and circular countour of pupil and iris, which are not true in majority cases.

In practical situations, it is observed that pupil and iris boundary are not circle and image quality is degraded because

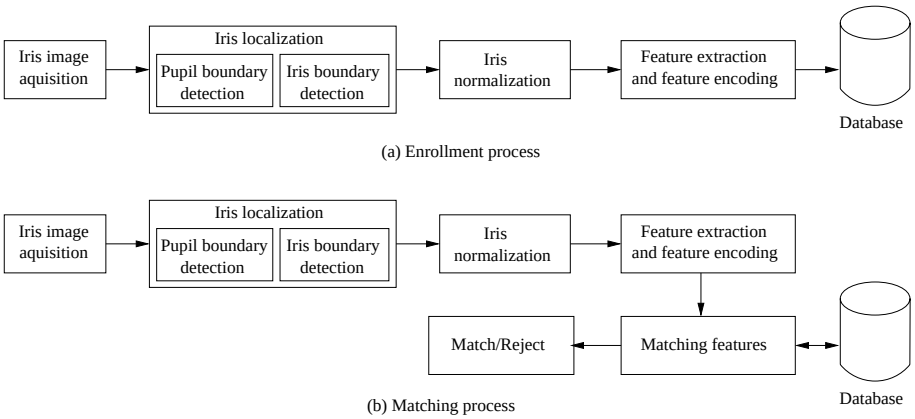

Fig. 2. Two basic processes in iris recognition.

of low contrast image, iris part is occluded bye eyelashes, improper eye open and light reflection. In such a situation existing approaches are fail to localize iris part correctly. In this paper, we have addressed these limitations. We also focus on reducing the searching space while detecting pupil and iris boundary.

In this paper, we propose a novel approach to localize iris part from an eye image efficiently and accurately. We propose scaling of image prior to its processing. We apply intensity transform to threshold the image. We also develop our own algorithm for detecting pupil boundary. For iris boundary detection, we apply intensity transform and dilation to threshold the image. After thresholding the image, we again apply dilation and then find vertical edges. Before finding the iris boundary, we eliminate the unnecessary edges and finally detect eyelids boundary.

The rest of the paper is organized as follows. In Section 2, we discuss the related work. Section 3 presents our proposed approach to detect iris boundary. In Section 4, we give the implentation of our approach and experimental results. Finally, the paper is concluded in Section 5.

II. Related work

Daugman's system [1], [2] uses integro-differential operator to detect pupil and iris boundary. Integro-differential operator fits the circular contours via gradient ascent on the parameterized center and radius of the circular contour. This operator is sensitive to the specular spot reflection of non diffused artificial light. Daugman new system [13] uses active contour model to detect pupil and iris boundary.

Wildes [4] and Masek [6] use binary edge map and voting each edge points to instantiate particular contour parameter values to detect pupil and iris boundary. The edge map is recovered via gradient-based edge detection. Prior to fit iris boundary contours, the derivatives are weighted to select vertical edges.

Ma et al. [5] calculate the summation of intensity value along each row and each column. They choose the particular row and column along which summations are minimum. That row and column are used as approximate x- and y-coordinate of pupil center. Once the approximate pupil center is chosen they apply Canny edge detection [14] and Hough transformation [3] in a rectangular region centered at pupil center to detect pupil and iris boundary circle.

In [10], Laplacian of Gaussian (LOG) operator is used for edge detection and median filter is used to remove the garbage pixels contain in the edge of iris image. This is followed by the counting of black pixels in each row and column. They use first pixel and last pixel positions of the row and column which contain the maximum number of black pixels to find out the pupil center and pupil radius. Subsequently, they use merging of existing edge segment into boundary by edge linking. Mid-point algorithm of circle and ellipse are used to fit pupil boundary. Similarly, boundary fitting technique by using a coarse scale is applied to locate iris boundary.

Tisse et al.[7] use integro-differential operator [2] with Hough transform [3] strategy. They use gradient decomposed Hough transform to find the pupil center and the iris center.

In [15], a linear threshold and Freemans chain code is used to isolate the pupil region and then central moment is used to find the pupil center. Next they create left and right vectors using the pixel values corresponding to left and right side of the pupil center. Left and right vectors start at left and right fringe of the pupil and ends at left and right side of image boundary. From these two vectors they find the iris radius.

Sung et al. [8] use Canny edge detection [14] method and bisection method to find the pupil center and shortest distance between the pupil center and the edges of the eyelids is set as iris boundary. Cui et al. [9] use Hough transform [3] following the Harr wavelet [16] transform for pupil segmentation and differential integral operator for localize the iris. Hough transform [3] followed by the edge detection is used to detect pupil and iris boundary in [9]. Theresholding and morphological opening is used to detect the pupil region and the center of the pupil region is calculated for the pupil center in [11]. They use active contour models (snakes) assuming the constraint that there is no internal energy to detect the iris boundary. Kim et al. [12] presents a method based on Gaussian mixture model (GMM) for segmentation of iris from an eye image.

From the reported work we may note the following. No work explicitly take into account the image quality and noises

like occurences of eyelids, eyelashes etc. Further, all work grossly assume circularness of pupil and pupil and iris are concentric, which is not true in most of the cases.

III. Proposed approach

In this section, we discuss our proposed approach of iris localization. We divide our approach in three tasks: *preprocessing*, *pupil boundary detection* and *iris boundary detection*. In our approach, we first preprocess an eye image and then we use that preprocessed image for the pupil and iris boundary detection. We consider *x*- axis of an eye image towards vertical and *y*- axis toward horizontal as shown in Fig. 1. We discuss these three tasks in the following sub sections.

A. Preprocessing

We consider scaling of the image as a preprocessing task to reduce the search area for the pupil centroid, pupil, and iris boundary. This follows the intensity transform to minimize the influence of the irrelevant edges as much as possible. In the following sub sections, we discuss these preprocessing tasks in details.

1) Down scaling: This is the first step in our proposed pupil detection method. We reduce the image size, which we refer to as down scaling. Down scaling reduces the search region for pupil boundary. The scale operator performs a geometric transformation, which can be used to shrink or zooming the size of an image (or part of an image). Image reduction, commonly known as sub sampling, is performed by replacement (of a group of pixel values by one arbitrarily chosen pixel value from within this group) or by interpolating between pixel values in a local neighborhoods. Image zooming is achieved by pixel replication or by interpolation. We use the *B*-spline interpolation [17] for both shrinking (down scaling) and zooming (up scaling). The higher order *B*-spline interpolation losses the less edge information than the nearest neighbor interpolation. We can commonly write scaling in homogeneous coordinates as shown in Equation (1).

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = S \times \begin{pmatrix} x \\ y \end{pmatrix} \tag{1}$$

Scaling transformation maps the pixel intensity value located at position (x, y) in an input image into new position $(x^{'}, y^{'})$ in an output image by applying a linear combination. S is the scaling transformation matrix, which is defined in Equation (2).

$$S = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \tag{2}$$

Here, S_x and S_y are the scaling factor in *x*- and *y*- directions, respectively. It is observed that the scaling factors influence the time of the pupil detection as well as the accuracy. Scaling factor may be chosen any value between 1 and 0. We have done experiments with scaling factors in this range of values. Down scale factor 0.50 in both *x*- and *y*- directions means that original image is reduced to $\frac{1}{2}$ of the original in both the directions. So the resultant image is $\frac{1}{4}$ of the original image.

Table I shows the experimental results with down scale factor 0.25, 0.50 and 0.75 in both *x*- and *y*- directions with respect to the search area and search time. In Table I, we see that the best result in terms of the search area and time for detecting pupil boundary when scaling factor is 0.25 (S_x = 0.25 and S_y = 0.25).

We further analyze the scaling effect and results of which is furnished in Table II. Table II shows the loss of pupil information with respect to three types of error, which are defined below.

- *Average pupil radius difference (R_{diff})* is the difference between actual pupil radius ($R_{pactual}$) and calculated pupil radius ($R_{pdetected}$) at a particular scale factor (i.e. $R_{diff} = |R_{pactual} - R_{pdetected}|$).
- *Distance between pupil centroid (D_{pc})* represents the distance between actual pupil center ($X_{pactual}$, $Y_{pactual}$) and calculated pupil center ($X_{pdetected}$, $Y_{pdetected}$) at a particular scale factor (i.e. $D_{pc} = |X_{pactual} - X_{pdetected}| + |Y_{pactual} - Y_{pdetected}|$).
- *% of failure to detect pupil area* with a particular scale factor (see Equation (3).

$$\%PA_{fail} = \frac{|PA_{actual} - PA_{detected}|}{PA_{actual}} \times 100 \tag{3}$$

In Equation (3), PA_{actual} and $PA_{detected}$ represent the actual pupil area and detected pupil area, respectively. In *Table II*, we see that the detection of pupil with scaling factor 0.25 (S_x = 0.25 and S_y = 0.25) produces maximum errors so far the radius of pupil is concerned. So far the detection of pupil center is concerned, we see that the results with scaling factor 0.50 (S_x = 0.50 and S_y = 0.50) are more or less similar to that of with scaling factor 0.75 (S_x = 0.75 and S_y = 0.75). Due to the loss of more edge information at scale factor 0.25 (S_x = 0.25 and S_y = 0.25) it will not be able to map the appropriate pupil boundary at the time of up scaling the pupil information. On the other hand, using the scale factor 0.50 (S_x = 0.50 and S_y = 0.50) and scale factor 0.75 (S_x = 0.75 and S_y = 0.75) we can accurately map the pupil boundary. But, scale factor 0.75 (S_x = 0.75 and S_y = 0.75) requires more number of computation as well as time without considerable improvement in pupil boundary result. Therefore, we choose the moderate scaling factor 0.50 (S_x = 0.50 and S_y = 0.50) to scale an eye image for the pupil boundary detection which gives the reduced search space and hence less searching time for the pupil detection.

2) Intensity transform: Prior to the pupil and iris detection we perform intensity transform. The intensity value at the pupil region is smaller than the iris region and intensity value at the sclera part is higher than the iris part. First, we create histogram of an eye image. Histogram of a typical eye image is shown in Fig. 3. In Fig. 3, we see that there are several peaks in the histogram. The peaks at lower intensity level represent the pupil region. The next peak in the histogram represents the iris region etc. Based on the histogram, we use contrast stretching [18] for removing the irrelevant part from the eye image for pupil detection. We apply Equation (4) for contrast stretch [18] operation.

$$s = \frac{s_{max}}{r_{max} - r_{min}} \times (r - r_{min}) \tag{4}$$

		Im-1	Im-2	Im-3	Im-4	Im-5
Sf= 0.25	Avg. pupil radius (in pixel)	16	24	20	16	20
	Pupil centroid	140, 176	112, 116	188, 120	112, 192	144, 132
	Pupil area (in pixel)	1168	1936	1312	1168	1328
	Search area	9×9	12×10	10×8	9×8	11×8
	Search time (ms)	1.92	2.123	1.964	1.952	2.056
Sf= 0.50	Avg. pupil radius (in pixel)	20	24	20	20	21
	Pupil centroid	138, 178	112, 116	188, 120	113, 193	142, 132
	Pupil area (in pixel)	1248	2020	1308	1138	1400
	Search area	21×17	26×22	22×17	25×20	22×19
	Search time (ms)	3.492	3.821	3.576	3.416	3.331
Sf= 0.75	Avg. pupil radius (in pixel)	20	25	20	19	20
	Pupil centroid	138, 177	112, 117	189, 120	113, 193	142, 133
	Pupil area (in pixel)	1266	2067	1316	1136	1382
	Search area	32×27	41×35	34×26	30×25	32×29
	Search time (ms)	6.524	7.889	6.385	5.987	7.293

TABLE I
PUPIL CENTROID AND RADIUS SEARCH WITHIN CONNECTED COMPONENT AT DIFFERENT SCALE FACTORS.

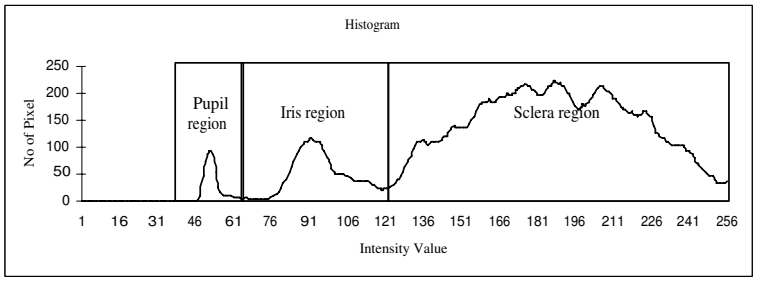

Fig. 3. Histogram of an eye image.

		Im-1	Im-2	Im-3	Im-4	Im-5
Sf = 0.25	R_{diff} (in pixel)	4	1	0	3	0
	D_{pc} (in pixel)	3	0	0	0	0
	$\%PA_{fail}$ (in pixel)	7.66	5.28	0.30	2.46	3.90
Sf = 0.50	R_{diff} (in pixel)	0	1	0	1	0
	D_{pc} (in pixel)	1	0	0	0	0
	$\%PA_{fail}$ (in pixel)	1.34	1.17	0.60	0.17	1.28
Sf = 0.75	R_{diff} (in pixel)	0	0	0	0	1
	D_{pc} (in pixel)	0	1	1	0	1
	$\%PA_{fail}$ (in pixel)	0.08	1.12	0.00	0.35	0.00

TABLE II
ERRORS IN DIFFERENT SCALE FACTORS.

In Equation (4), r_{min} and r_{max} are the minimum and maximum input intensity values, respectively. s_{max} is the maximum output intensity value. r and s are the input and output intensity values, respectively. For pupil detection, we use minimum mean value as r_{min} and add an offset ($\delta 1$) with r_{min} to obtain r_{max}, which are obtained in the following Equation (5).

$$\begin{aligned} r_{min} = \quad & min(\textstyle\sum_{i=-1}^{1} I(x+i, y+i)) \quad \textit{for x=1 to IH-1} \\ & \qquad\qquad\qquad\qquad\qquad\qquad \textit{and y=1 to IW-1} \\ r_{max} = \quad & R_{min} + \delta 1 \end{aligned} \tag{5}$$

$I(x, y)$ is the intensity value at (x, y) position in the eye image. IH and IW are the image height and image width, respectively. We decide the value of $\delta 1$ by experiment with training images. In our experiment, $\delta 1 = 30$ is taken empirically. Figure 5(a) shows an input eye image and Fig. 5(b) shows the eye image after the intensity transformation before the pupil boundary detection.

B. Pupil boundary detection

We use the preprocessed image obtained as discussed in the last section for pupil boundary detection. We divide the pupil boundary detection task into several sub tasks which are shown in Fig. 4. All the tasks as shown in Fig. 4 are discussed in the following.

1) Creating binary image: This is the first step in our pupil detection approach. We eliminate the high intensity region from the eye image because the pupil region consists of low

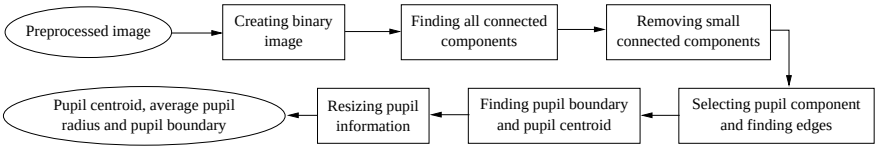

Fig. 4. Tasks in pupil detection.

intensity value. We use Equation (6) to create binary image from the intensity transformed image. The pixels intensity values in an eye image are set to 0 in the binary image if intensity values are less than 128 and otherwise 255, as shown in Equation (6).

$$I^{'}(x,y) = \begin{cases} 0 & if I(x,y) \leq 127 \\ 255 & \text{otherwise} \end{cases} \quad (6)$$

$I^{'}(x,y)$ and $I(x,y)$ are the intensity value at (x, y) position in the binary image and intensity transformed eye image, respectively. Figure 5(c) shows one result obtained in our experiment.

2) Finding all connected components: After creating the binary image, we find out all connected components by checking the neighborhood of each pixel in a binary image. We put all detected connected components in a list. Each connected component is defined with Len, L, R, T, B, *x-span*, *y-span*. These parameters are defined bellow for a connected component C_i, say.

Len_i = Total no of connected points in C_i.
L_i = The point on C_i with minimum *y*-coordinate value.
R_i = The point on C_i with maximum *y*-coordinate value.
T_i = The point on C_i with minimum *x*-coordinate value.
B_i = The point on C_i with maximum *x*-coordinate value.
x-span$_i$ = |*x*-coordinate of B_i − *x*-coordinate of T_i|
y-span$_i$ = |*y*-coordinate of R_i − *y*-coordinate of L_i|.

Figure 5(d) illustrates the above definition. Figure 5(d) shows an example of connected components C_1 and C_2, which are obtained for an input image and followed by the previously stated operations.

3) Removing small connected components: Irrelevant connected elements occur due to eyelids, eyelashes, light reflection and non uniform illumination. We try to remove the irrelevant components as much as possible from the eye image before choosing the pupil component. As a process of finding all connected components, we already identified a set of all connected components C, say. From this set C, we identify another set of connected components $C^{'}$ such that $Len_i \geq 0.2 * MaxLength$ && $(x_{span_i} \leq 1.5 * y_{span_i})$ && $(y_{span_i} \leq 1.5 * x_{span_i})$ && $(x_{span_i} \neq 0)$ && $(y_{span_i} \neq 0)$, for any *i-th* component $\in C$.

Here, $MaxLenght$ is the maximum length of the connected component among all connected components. Now, it is obvious that $C^{'} \subseteq C$. It is also experimentally observed that the connected components in $C^{'}$, form an rectangular block around that connected component. The height-width or width-height ratio of the rectangular block is less than 1.5. In our subsequent task, we consider all connected components

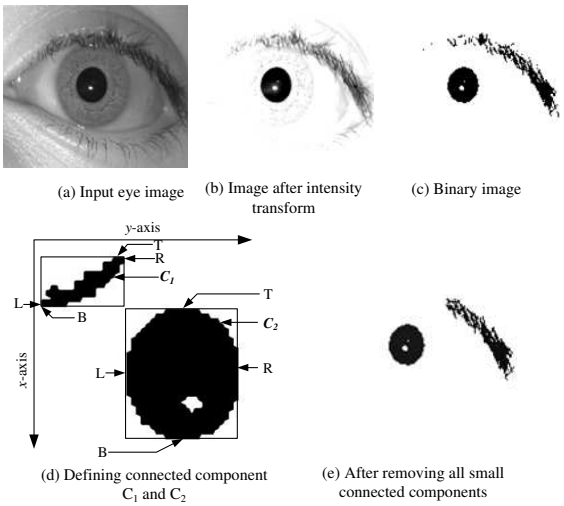

Fig. 5. Eye image on different steps.

in $C^{'}$ only. Figure 5(e) shows the connected components after removing small connected components in our running example.

4) Selecting pupil component and finding edges:

After eliminating the small components from an image there further may present extra connected components which are actually not related to pupil. So, our next task is to choose the pupil component only from $C^{'}$. To do this, we first calculate the centroid and the average radius for each component $C_i^{'} \in C^{'}$. We use Equation (7) to calculate the centroid and average radius for the i^{th} component.

$$x_{c_i} = \frac{\sum x_i}{N_i} \quad \text{and} \quad y_{c_i} = \frac{\sum y_i}{N_i}$$
$$R_{c_i} = \frac{\sum \sqrt{(x_i - x_{c_i})^2 + (y_i - y_{c_i})^2}}{N_i} \quad (7)$$

where x_{c_i} and y_{c_i} is the *x*- and *y*- coordinates of centroid, respectively and R_{c_i} is the average radius of the i^{th} component. (x_i, y_i) is the pixel position of the i^{th} component.

After calculating the centroid and average radius, we fit the circle of radius R_{c_i} at (x_{c_i}, y_{c_i}) and count the pixels say, P_{b_i} which belongs to the i^{th} component and the number of pixels say, P_{n_i} which does not belong to the i^{th} component within that circular region fitted with R_{c_i} at (x_{c_i}, y_{c_i}). We then select the component as pupil component using Equation (8).

$$p = \max_i \{\frac{P_{b_i}}{P_{b_i} + P_{n_i}}\} \qquad \textit{for all } i \quad (8)$$

5) Finding pupil boundary and pupil centroid: Now, we fill the small white region within pupil component, which

occur due to the light reflections with black pixels and then find the edge point of the pupil component. All edge points which we have obtained from the previous step may not be connected. There may be broken edge due to light reflection at the pupil border or wrong edge points due to eye lashes. So, our next task is to construct the actual pupil boundary edge. We bounded the pupil edge points by a rectangular block which again divides into four quadrants as shown in Fig. 6.

We started the pupil edge point finding from that edge point where the y-axis intersect the pupil boundary. From that point we check the 1, 2, 3 pixel positions (see Fig. 6) in the image to find the next edge point. If edge points occur at 1 or 2 position for first 45 degree from the starting point we left that point as original edge point. Similarly, we left the edge point as actual edge point if edge point occurs at position 2 or 3 for the next 45 degree. If edge point occurs at both 1 and 3 positions we choose the position 2 as the next actual pupil edge point. For the first 45 degree if the next edge points occurs at position 3, we back trace the previously determined actual edge point, if there exist an edge point which has an edge point in horizontal position, then we assign the position 2 as new pupil edge point, otherwise we choose position 3 as new edge point. For the next 45 degree, if the next edge points occurs at position 1, then we back trace the previously determined actual edge point upto 45 degree form the upper x axis and if we find a point which has an edge point in vertical position we assign the position 2 as new pupil edge point, otherwise we choose position 1 as new point.

We apply the similar approach for other three quadrants to find the actual pupil boundary. After completion of finding actual pupil boundary we fill inside the pupil boundary with black pixel and recalculate the pupil centroid (x_{dp}, y_{dp}) and average pupil radius (R_{dpavg}).

6) Resizing pupil information: We get the pupil centroid, radius and boundary points for down scale image. So, we need to up scale these information to get pupil information for original eye image. We multiply the average radius and centroid by two to up scale. We up scale the pupil boundary image with scale factor 2 using Equation (1).

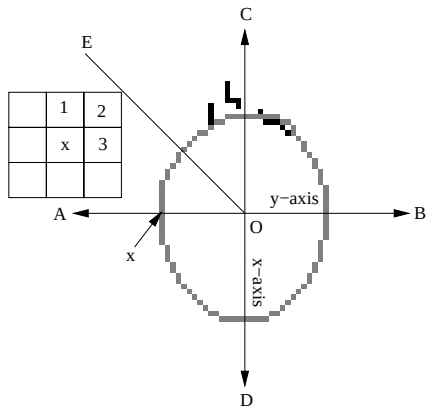

Fig. 6. Pupil boundary finding.

C. Iris boundary detection

We consider down scaled [18] input eye image with scale factor 0.50 which we obtain as discussed in Sec 3.1.1. We also use the pupil information, which we obtain following the approach discussed in Sec 3.2. In our approach of the iris boundary detection, we divide this step into several sub tasks. An overview of our iris detection approach is shown in Fig. 7. All the tasks mentioned in Fig. 7 are discussed in the following sub sections.

1) Intensity transformation: Prior to the pupil and iris detection we perform intensity transform [18]. This operation is already discussed in Section 3.1.2. Based on the histogram properties we use contrast stretching [18] to increase the contrast between iris and sclera part. We divide the eye image into left and right sub images at pupil centroid. We apply contrast stretch [18] on the left and right images separately because illumination in the both sides of iris are not necessarily same.

We divide the eye image into two sub images, left and right sub images at the y- coordinate of pupil centroid (y_{pc}). Then we apply Equation (9) for contrast stretch [18] in the both sub images.

$$s = \frac{s_{max}}{r_{max} - r_{min}} \times (r - r_{min}) \tag{9}$$

In Equation (9), r_{min} and r_{max} are the minimum and maximum input intensity values, respectively. s_{max} is the maximum output intensity value. r and s are the input and output intensity values at each pixel position in the eye image, respectively. For iris detection, the values of r_{min} and r_{max} are determined using Equation (10).

$$r_{min} = \frac{1}{64} \sum_{x=x_{min}}^{x_{max}} \sum_{y=y_{min}}^{y_{max}} I(x,y)$$
$$r_{max} = r_{min} + \delta \tag{10}$$

where, $x_{min} = x_{pc} - 8$ and $x_{max} = x_{pc} + 8$. $y_{min} = y_{pc} - R_{pavg}$ and $y_{max} = y_{pc} - R_{pavg} - 15$ for left sub image and $y_{min} = y_{pc} + R_{pavg}$ and $y_{max} = y_{pc} + R_{pavg} + 15$ for right sub image. R_{pavg} and (x_{pc}, y_{pc}) are the average pupil radius and pupil centroid, respectively. δ is an offset value. We decide the value of δ by experiment with training images. In our experiment, $\delta = 50$. Fig. 8(b) shows the result after applying intensity transformation with r_{min} and r_{max} to image shown in Figure 8(a).

2) Dilation: The dilation [18] is an morphological operation in image processing. We apply dilation preprocessing operation on the intensity transformed image. This reduces the intensity value within the iris region and increase the contrast between iris and sclera region. We consider a 3×3 block centred at each pixel in the intensity transformed image. Dilation is done by replacing the intensity value at each pixel with the minimum intensity value of that 3×3 block. We apply this operation on the whole image obtained in the previous step (intensity transformation). Figure 8(c) shows the dilated image corresponding to the image shown in Fig. 8(b).

3) Thresholding image: In this step, we threshold the image based on some threshold value. We divide the eye image into

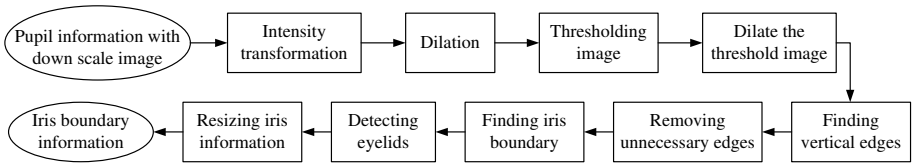

Fig. 7. Tasks in the proposed iris detection technique.

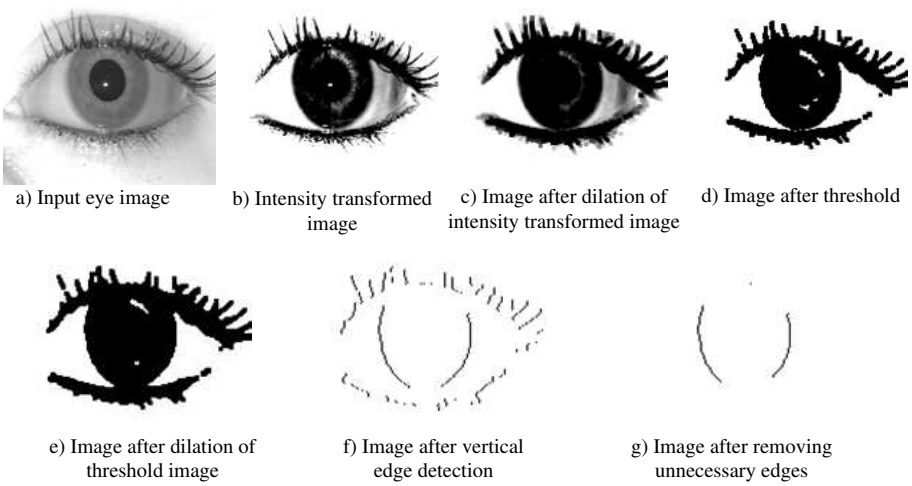

Fig. 8. Eye images at different steps.

two sub images, left and right sub images at the y- coordinate of pupil centroid (y_{pc}). We apply Equation (11) to threshold left and right sub images.

$$I^{'}(x,y) = \begin{cases} 255 & if I(x,y) \geq Th \\ 0 & \text{Otherwise} \end{cases} \quad (11)$$

where, Th is the threshold value. We calculate the separate threshold values Th_L and Th_R for the left and right sub images, respectively. To calculate these threshold values, we use Equation (12).

$$Th_L = \frac{\sum_{x=0}^{IH-1}\sum_{y=0}^{y_{pc}-1} I(x,y)}{IH \times y_{pc} - N_{255}} \quad if\ I(x,y) \neq 255$$
$$Th_R = \frac{\sum_{x=0}^{IH-1}\sum_{y=y_{pc}}^{IW-1} I(x,y)}{IH \times (IW - y_{pc}) - N_{255}} \quad if\ I(x,y) \neq 255 \quad (12)$$

where, IH and IW represents the height and width of the image, respectively. In Equation (12), y_{pc} is the pupil centroid and N_{255} is the number of pixel with intensity value 255. Figure 8(d) shows the threshold image obtained from the image shown in Fig. 8(c).

4) Dilate the threshold image: After thresholding the image, there may present white region inside the iris region due to high intensity value in iris texture pattern. We try to fill the white spot with black pixel within the iris region as much as possible so that no irrelevant edges are created within the iris region. We then apply the dilation [18] operation. We consider 5×5 block around each pixel in the image and count the black pixel. We replace the pixel with minimum intensity value if count is greater than the half of the block size. We choose 5×5 block so that it preserves the white region between iris and sclera region. Figure 8(e) shows the dilated image as obtained starting with the the threshold image in Fig. 8(d).

5) Finding vertical edges: Iris boundary mainly formed by vertical edges between iris and sclera region, as maximum time upper and lower parts of images are occluded by eyelids. In this step, we find the vertical edges. We find the vertical edge by traversing each row of the image. We consider a point as a vertical edge point if intensity value change from white to black up to y-coordinate of the pupil center and black to white if y- coordinate of the pixel is within y- coordinate of pupil centroid to image width. Figure 8(f) shows the result of detection of vertical edges of the eye image in Fig. 8(e).

6) Removing unnecessary edges: The resultant image as obtained in the last step usually contains many edges which are not due to iris boundary. So, in this step, we eliminate the unnecessary vertical edges which are not relevant to iris boundary. We try to select edge pixels present in the image which is mainly for the iris boundary. We calculate the distance of each edge points from the pupil centroid for left and right sub image. Then create two distance histogram for two sub image. Distance histogram represents the number of pixels in each distance. We normalize the distance histogram dividing the number of pixels by corresponding distance. For left sub image we select those connected component which has any pixel with distance corresponding to the maximum normalize value in the distance histogram of the left sub image. We apply similar approach for the right sub image. Figure 8(g) shows the edge corresponding to the iris boundary following the image

in Fig. 8(f).

D. Finding iris boundary

Now we check the connectivity of the iris boundary in this step. We divide the edge image into four quadrants namely, *LT, RT, LB* and *RB* (see Figure 9(a)). Then we fit the edge points in each quadrant. For *LT* and *LB* quadrants, we start searching from the intersection point of *y*- axis and iris edge point in the left sub image. Suppose, *x* is the current point at *LT* shown in Figure 9(a). We search the position 1, 2, 3 with respect to *x* in edge image and also check a 5×3 block at left top of position 1 in the dilated image (see Figure 9(b) and (c)). If in the dilated image, density of black pixel is greater than 10 (i.e. $\geq 5 \times 2$) in the 5×3 block, then we terminate the search and choose this point as starting eyelids points in *LT* quadrant because after that position iris boundary is occluded by eyelids. If no pixel is found in 1, 2 or 3 position then we search the next point along the each upper row for *LT* quadrant. At a point when it finds an edge point, we connect these two points with a line for iris boundary. In the similar way, just changing the direction of search points and block we find the starting point of eyelids in *RT, LB* and *RB* quadrants. After finding the starting points of eyelids in each quadrant we detect the eyelids. Eyelids detection is discussed in the next sub section.

E. Detecting eyelids

We have already got the starting points of eyelids from the previous section. Using those points we draw some line in particular angular direction to detect eyelids if any. Algorithm 1 describes the upper eyelid detection using *LT* and *RT* quadrants. For detecting lower eyelid we apply same algorithm in *LB* and *RB* quadrants but in opposite direction. Figure 9(d) illustrate the eyelids detection procedure. Figure 9(e) shows the detected iris and eyelids from eye image.

```
1  x1 and x6 are starting point in LT and RT quadrant,
   respectively;
2  AB and CD are y- and x- axis, respectively;
3  ∠AOE = ∠BOH = 45° ;
4  ∠DOF = ∠DOG = 15° ;
5  if ∠AOx1 < 45° then
6      draw line from x1 to OE with ∠Ox1x2 = 45°;
7      draw line from x2 to OF with ∠Ox2x3 = 60°;
8  else
9      if ∠AOx1 < 75° then
10         draw line from x1 to OF with
           ∠Ox1x3 = 60°;
11     else
12         if ∠AOx1 <= 90° then
13             assign x3 = x1;
14         end
15     end
16 end
17 if ∠BOx6 < 45° then
18     draw line from x6 to OH with ∠Ox6x5 = 45°;
19     draw line from x5 to OG with ∠Ox5x4 = 60°;
20 else
21     if ∠HOx6 < 75° then
22         draw line from x6 to OG with
           ∠Ox6x4 = 60°;
23     else
24         if ∠AOE <= 90° then
25             assign x4 = x6;
26         end
27     end
28 end
29 draw line from x3 to x4 ;
```

Algorithm 1: Algorithm for detecting upper eyelid

F. Resizing iris information

We get the iris boundary points for down scale image. So, we need to up scale this information to get iris information for the original eye image. We up scale the iris boundary image with scale factor 2 using Equation (1). After getting the iris boundary for original image we remove the outside of the iris boundary.

IV. Experimental results

We have implemented our proposed iris localization approach using C programming language in Fedora Core 5 operating system environment. We use GNU compiler GCC version 4.1.0 for compiling and executing our program. For plotting graph we use GNU plot version 4.0. Our approach has been tested with 1000 images of [19], 1800 images of UBIRIS [20], 450 images of [21], 22,000 images of CASIA ver-3.0 [22] and 2900 images of ICE 2005 [23] iris image databases. We compare our approach with some best known algorithms [2], [13], [4], [6], [24]. There are some reported results of these approaches; however, those results are with CASIA [22] ver-1.0 iris database. CASIA ver-1.0 database is now obsolete because data are hand-edited by painting the entire pupil with a circular disk of uniformly dark pixels and hence making the database trivial. To compare our approach with others we implement their iris localization approaches in the same experimental setup as our approach.

Table III shows the errors on average pupil radius, distortion of pupil center and detected pupil area according to our approach and existing approaches [2], [13], [4], [6], [24]. In Table III, we see that our approach takes minimum time than the other approaches as well as with higher accuracy rate of the iris localization.

We measure the *Accuracy rate* (AC_{rate}) based on the *Accuracy error* (Err_{ir}). *Accuracy error* is defined in Equation (13).

$$Err_{ir} = \frac{|N_{iact} - N_{idet}|}{N_{iact}} \times 100 \qquad (13)$$

where, N_{iact} and N_{idet} are the number of actual and detected iris pixels, respectively. We calculate the actual iris pixels (N_{iact}) manually using GIMP image processing tool. The detected iris pixels (N_{idet}) are counted from the detected iris part with a simple row-major scanning method. If Err_{ir} is

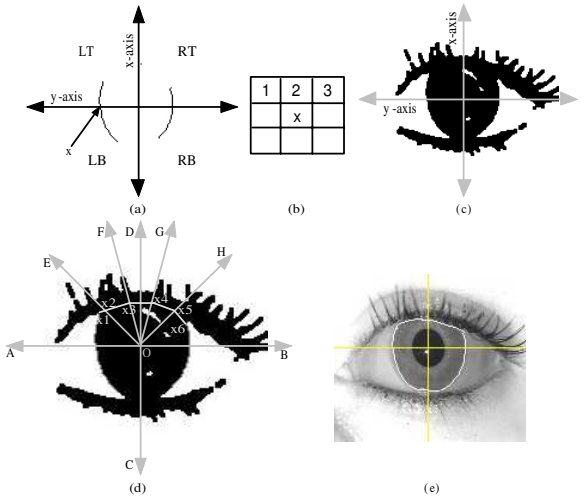

Fig. 9. Iris boundary finding.

less than 10% then we consider the iris part is successfully detected. *Accuracy rate* is defined in Equation (14).

$$AC_{rate} = \frac{N_s}{N_t} \times 100 \tag{14}$$

where, N_s is the number of eye images in which iris part is successfully detected and N_t is total number of eye images.

In Table IV, we show the error rate to localize iris part for the five benchmark images which are selected randomly from the three different image databases using Equation (13). Here, N_{iact} and N_{idet} are the number of actual and detected iris pixels, respectively. If Err_{ir} is less than 10% then we consider the iris part is successfully detected. *Accuracy rate* is defined in Equation (14). We see that average error rate is less 10% irrespective of the type of databases according to our approach.

We compare the accuracy rate of our approach with others. Accuracy rate is measured using Equation (14). The result of comparisons is shown in Table V. From Table V, we see that the proposed approach has the best performance. The accuracy rate is up to 94.51% for CASIA, 95.49% for ICE 2005, 98.58% for Bath, 94.26% for UBIRIS and 98.47% for MMU iris database. We also show that our approach requires less time than the others.

A few sample runs of iris detections on some iris images

		Daugman [2]	Daugman New [13]	Masek [6]	Wildes [4]	Ma et. al. [24]	Proposed
CASIA Ver-3.0	Mean errors in R_{diff} (in pixel)	4.39	3.02	5.15	6.12	5.39	1.14
	Mean errors in D_{pc} (in pixel)	2.15	1.19	3.67	5.37	4.79	1.89
	% ofPA_{fail}	6.71	4.56	6.59	7.77	7.34	4.12
	Average time (ms)	523.14	29.17	97.52	379.61	363.64	28.57
	Accuracy (%)	97.12	97.78	91.12	89.32	87.57	97.63
ICE 2005	Mean errors in R_{diff} (in pixel)	3.53	3.11	7.23	7.51	7.38	2.95
	Mean errors in D_{pc} (in pixel)	2.13	1.79	5.59	6.37	5.92	1.95
	% ofPA_{fail}	5.02	4.93	7.67	7.89	8.57	4.41
	Average time (ms)	498.21	27.61	112.25	407.51	378.87	25.33
	Accuracy (%)	96.21	97.05	89.41	87.11	85.29	96.79
UBIRIS	Mean errors in R_{diff} (in pixel)	2.23	2.21	4.65	2.96	3.68	0.32
	Mean errors in D_{pc} (in pixel)	1.97	1.82	3.24	2.15	2.97	0.97
	% ofPA_{fail}	3.95	3.33	8.46	5.46	7.19	1.82
	Average time (ms)	305.76	20.23	85.34	276.16	256.39	19.45
	Accuracy (%)	95.78	96.91	91.67	94.45	93.66	96.67
Bath	Mean errors in R_{diff} (in pixel)	4.54	3.19	6.72	4.38	4.65	0.56
	Mean errors in D_{pc} (in pixel)	3.73	3.27	5.32	3.44	4.29	1.53
	% ofPA_{fail}	4.89	3.62	8.96	8.46	7.81	1.67
	Average time (ms)	478.47	26.55	108.26	394.16	376.83	24.89
	Accuracy (%)	99.2	99.3	94.5	98.9	98.1	100
MMU	Mean errors in R_{diff} (in pixel)	4.18	3.76	5.78	3.96	4.67	0.41
	Mean errors in D_{pc} (in pixel)	2.61	1.14	4.98	3.15	3.92	1.02
	% ofPA_{fail}	4.65	2.39	8.49	7.89	8.14	1.16
	Average time (ms)	398.98	25.47	99.78	354.55	317.18	24.95
	Accuracy (%)	98.89	99.13	93.33	98.22	97.87	99.11

TABLE III
COMPARISON OF ACCURACY RESULTS ON PUPIL DETECTION.

	Images	Err_{ir}					
		Daugman [2]	Daugman New [13]	Masek [6]	Wildes [4]	Ma et. al. [24]	Proposed
CASIA Ver-3.0	S1009L01	7.12	4.21	8.25	5.31	7.19	3.56
	S1025L04	7.90	5.61	9.10	6.45	8.20	4.18
	S1095R03	9.61	8.29	10.22	8.03	8.76	7.55
	S1090R02	8.22	10.51	13.52	11.12	9.12	10.12
	S1137L04	3.15	2.12	5.21	2.89	4.91	3.61
ICE 2005	239766	7.91	6.21	8.88	13.21	12.59	4.49
	254451	8.12	11.51	9.85	12.67	9.18	9.97
	236657	11.12	10.25	14.67	8.12	14.52	12.33
	227542	12.56	9.59	12.12	7.61	9.83	8.27
	215461	4.51	5.23	6.22	8.23	3.12	4.19
UBIRIS	Img_2_1_3	5.67	4.51	7.63	8.98	6.21	4.56
	Img_9_1_2	7.55	6.23	8.07	9.34	8.73	6.29
	Img_216_1_5	9.98	8.12	11.92	12.59	12.88	9.36
	Img_130_2_4	9.27	8.32	10.07	11.08	13.32	8.26
	Img_201_2_2	10.12	8.99	13.17	14.11	16.71	9.47
Bath	0001-L-0005	11.57	7.52	17.89	20.56	19.21	9.34
	0004-R-0015	8.76	6.21	9.85	10.06	8.75	4.27
	0008-R-0007	19.97	7.89	17.99	18.96	12.35	7.12
	0014-L-0020	5.45	5.12	8.7	9.87	4.62	3.18
	0025-L-0014	5.01	4.91	10.03	11.05	9.25	5.22
MMU	aeval3	4.67	5.61	8.77	9.16	7.62	2.94
	chingycl3	9.36	8.91	15.12	17.31	12.13	6.51
	chongpkr4	4.43	3.12	6.83	8.72	2.25	2.79
	fatmal1	4.26	3.97	7.98	8.99	4.62	4.73
	mimil1	10.07	11.41	17.17	11.22	13.55	7.23

TABLE IV
COMPARISON OF ERROR RATES RESULTS ON IRIS DETECTION.

		Daugman [2]	Daugman New [13]	Masek [6]	Wildes [4]	Ma et. al. [24]	Proposed
AC_{rate}	CASIA Ver-3	96.21	96.12	81.55	91.41	94.11	94.51
	ICE 2005	92.15	97.52	79.51	82.12	87.55	95.49
	UBIRIS	90.27	89.21	82.65	87.24	86.22	94.26
	Bath	91.2	98.1	82.5	89.6	87.5	98.5
	MMU	85.64	98.23	83.92	2.48	91.02	98.41
Average time (ms)	CASIA Ver-3	923.7	35.6	685.2	935.7	679.5	24.1
	ICE 2005	870.5	30.2	623.2	828.4	612.2	22.1
	UBIRIS	674.4	23.2	412.4	537.3	491.5	16.9
	Bath	889.6	33.7	647.7	789.5	589.6	23.8
	MMU	823.1	29.6	587.1	694.8	552.4	20.3

TABLE V
COMPARISION OF ACCURACY RATES ON OUR IRIS LOCALIZATION.

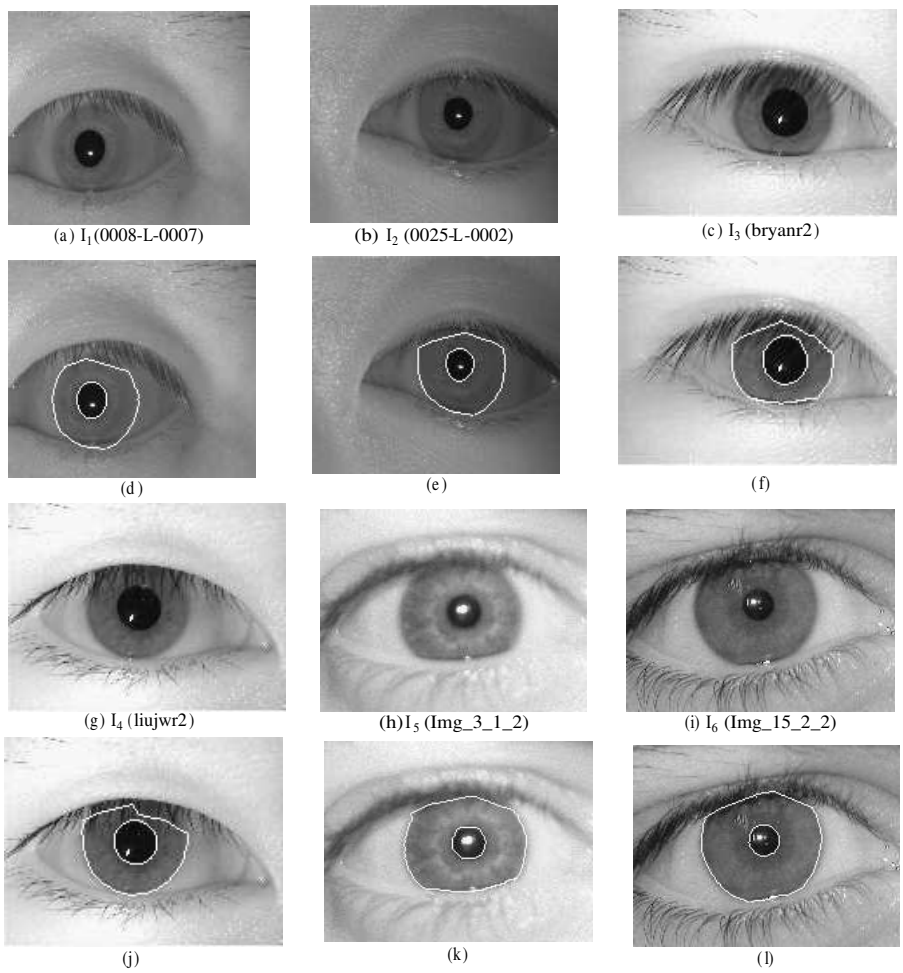

Fig. 10. Iris boundary detection on different iris images.

following our approach is shown in Figure 10. Eye images I_1 (Figure 10(a)) and I_2 (Figure 10(b)) chosen from the Bath iris database where pupils are not circular. For these two images, our iris detection approach successfully able to detect iris parts, which are shown in Figure 10(d) and Figure 10(e) (corresponding to the eye images I_1 (Figure 10a) and I_2 (Figure 10(b)), respectively). Figure 10(c) and Figure 10(g) show two iris images I_3 and I_4, respectively chosen from the MMU iris database. We see that these two images are highly occluded with eyelashes. Our iris detection approach also able to detect iris part successfully for these two images, which are shown in Figure 10(f) and Figure 10(j) (corresponding to the eye images I_3 (Figure 10(c)) and I_4 (Figure 10(g)), respectively). Two eye images I_5 (Figure 10(h)) and I_6 (Figure 10(i)) are chosen from the UBIRIS database, which are of low quality images. For these images too our iris detection approach successfully detects the iris parts which are shown in Figure 10(k) and Figure 10(l) corresponding to the images I_5 (Figure 10(h)) and I_6 (Figure 10(i)), respectively.

V. Conclusion

Iris localization is the beginning task in any iris-based biometric authentication system and if the iris part of an eye image is not detected accurately then it leads to errors in overall identification method. In this work, we focus on efficient and accurate iris localization method for developing better biometric identification system in widespread application areas. Our approach is able to isolate iris part from iris images with inferior image quality, occlusion of eye-lashes. Our approach also addresses the issue of processing iris images where pupil and iris boundaries are not necessarily perfect circular. To deal with these problems we consider different operations such as, binary image creation, finding all connected component, removal of small connected component, selecting pupil component and finding pupil component for pupil boundary detection and intensity level transformation, dilation, image thresholding, removal of irrelevant edges and eyelid detection for iris boundary. These sub tasks, nevertheless, without much computational overhead compared to the existing approaches of iris detection. Experimental results reveal that our approach is approximately 75% faster than the existing approaches.

More significantly, accuracy rate in iris detection according to our approach is comparable to that of all the reported work. With our approach, we achieve two objectives: speed and accuracy in iris localization, which are important to realize high speed and more reliable biometric authentication systems based on iris.With the encouraging results as substantiated by our thorough experiment we may claim that our approach increases the potentiality of iris biometric to be applied in real-life applications.

Acknowledgments

The work is supported by Indian Institute of Technology, Kharagpur under the scheme of Institute Sponsor Innovative Research and Development (ISIRD). Authors wish to acknowledge Multimedia University for providing MMU iris database and Department of Computer Science, University of Beira Interior for providing UBIRIS dataset and University of Bath for providing Bath iris image database.

REFERENCES

[1] J. G. Daugman, "High Confidence Visual Recognition of Persons by a Test of Statistical Independence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1148–1161, November 1993.
[2] J. Daugman, "How iris recognition works," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 21– 30, 2004.
[3] P. Hough, "Method and means for recognizing complex patterns." U.S. Patent 3,069,654, December 1962.
[4] R. P. Wildes, "Iris Recognition: An Emerging Biometric Technology," *Proceedings of the IEEE*, vol. 85, pp. 1348–1363, September 1997.
[5] L. Ma, T. Tan, Y. Wang, and D. Zhang, "Efficient Iris Recognition by Characterizing Key Local Variations," *IEEE Transactions on Image Processing*, vol. 13, pp. 739–750, June 2004.
[6] L. Masek, "Recognition of human iris patterns for biometric identification." http://www.csse.uwa.edu.au/ pk/studentprojects/libor, 2003.
[7] C.L.Tisse, L.Martin, L.Torres, and M.Robert, "Person Identification Technique Using Human Iris Recognition," in *Proceedings of Vision Interface*, (Canada), pp. 294–299, 2002.
[8] H. Sung, J.Lim, J.Park, and Y.Lee, "Iris Recognition Using Collarette Boundary Localization," in *Proceedings of 17th International Conference on Pattern Recognition (ICPR'04)*, vol. 4, pp. 857–860, August 2004.
[9] J. Cui, Y. Wang, T. Tan, L. Ma, and Z. Sun, "An iris recognition algorithm using local extreme points," in *First International Conference Biometric Authentication* (D. Zhang and A. K. Jain, eds.), vol. 3072 of *Lecture Notes in Computer Science*, pp. 442–449, Springer, 2004.
[10] J. M. H. Ali and A. E. Hassanien, "An Iris Recognition System to Enhance E-security Environment Based on Wavelet Theory," *AMO - Advanced Modeling and Optimization journal*, vol. 5, no. 2, pp. 93–104, 2003.
[11] T. Mäenpää, "An Iterative Algorithm for Fast Iris Detection," in *Advance in Biometric Person Authentication,LNCS 3781*, vol. 5404, pp. 127–134, 2005.
[12] J. Kim, S. Cho, J. Choi, and I. Robert J. Marks, "Iris recognition using wavelet features," *J. VLSI Signal Process. Syst.*, vol. 38, no. 2, pp. 147–156, 2004.
[13] J. Daugman, "New Methods in Iris Recognition," *IEEE Transaction on Systems, Man and Cybernatics-Part B: Cybernatics*, vol. 37, pp. 1167–1175, October 2007.
[14] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.
[15] M. Vasta, R. Singh, and A.Noore, "Reducing the False Rejection Rate of Iris Recognition Using Textural and Topological Fearures," *International Journal of Signal Processing*, vol. 2, no. 2, pp. 66–72, 2005.
[16] A. Jensen and A. la Cour-Harbo, *Ripples in Mathematics: The Discrete Wavelet Transform*. Springer, 2001.
[17] P. Thévenaz, T. Blu, and M. Unser, "Interpolation revisited," *IEEE Transactions on Medical Imaging*, vol. 19, pp. 739–758, July 2000.
[18] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2nd ed., 2002.
[19] "University of Bath iris image database, 2007." http://www.bath.ac.uk/elec-eng/research/sipg/irisw (accessed July, 2007).
[20] H. Proença and L. A. Alexandre, "Ubiris: A noisy iris image database.," in *ICIAP* (F. Roli and S. Vitulano, eds.), vol. 3617 of *Lecture Notes in Computer Science*, pp. 970–977, Springer, 2005.
[21] "Multimedia University iris image database." http://pesona.mmu.edu.my/čcteo/ (accessed July, 2007).
[22] "CASIA iris image database." http://www.sinobiometrics.com (accessed July, 2007).
[23] "Iris Challenge Evaluation,2005." http://iris.nist.gov/ice/ (accessed July, 2008), 2005.
[24] L. Ma, T. Tan, D. Zhang, and Y. Wang, "Local Intensity Variation Analysis for Iris Recognition," *Pattern Recognition*, vol. 37, no. 6, pp. 1287–1298, 2004.

Somnath Dey received B. Tech in Information Technology from Kalyani University in 2004 and his MS (by research) degree in Information Technology from School of Information Technology, Indian Institute of Technology, Kharagpur in 2008. Presently, he is pursuing his Ph.D. in School of Information Technology, Indian Institute of Technology, Kharagpur. His research interests include biometrics, image processing, pattern recognition and Human Computer Interaction.

Debasis Samanta received his B. Tech. degree in Computer Science and Engineering from the Calcutta University, in 1993, and his M. Tech. in Computer Science and Engineering from the Jadavpur University, in 1995 and Ph.D. degrees in Computer Science and Engineering from Indian Institute of Technology, Kharagpur in 2002. He is currently an Assistant Professor in the School of Information Technology at the Indian Institute of Technology, Kharagpur. His research interests include Low Power VLSI Systems Design, Human Computer Interaction, and Information System Design.