Robust iris biometric system for visible wavelength data

Farmanullah Jan[1,2*], Imran Usman[3], and Shahid A. Khan[2]

[1]*Department of Physics, COMSATS Institute of Information Technology, Park Road, Chak Shahzad, 44000, Islamabad, Pakistan*

[2]*Department of Electrical Engineering, COMSATS Institute of Information Technology, Park Road, Chak Shahzad, 44000, Islamabad, Pakistan*

[3]*Center for Advanced Studies in Telecommunications, COMSATS Institute of Information Technology, Park Road, Chak Shahzad, 44000, Islamabad, Pakistan*

[*]*Corresponding author: farmanullah_jan@comsats.edu.pk*

Received March 20, 2013; accepted May 13, 2013; posted online August 2, 2013

Commercial iris biometric systems exhibit good performance for near-infrared (NIR) images but poor performance for visible wavelength (VW) data. To address this problem, we propose an iris biometric system for VW data. The system includes localizing iris boundaries that use bimodal thresholding, Euclidean distance transform (EDT), and a circular pixel counting scheme (CPCS). Eyelids are localized using a parabolic pixel counting scheme (PPCS), and eyelashes, light reflections, and skin parts are adaptively detected using image intensity. Features are extracted using the log Gabor filter, and finally, matching is performed using Hamming distance (HD). The experimental results on UBIRIS and CASIA show that the proposed technique outperforms contemporary approaches.

OCIS codes: 150.0150, 100.0100, 110.0110.

doi: 10.3788/COL201311.081501.

Biometric technology relies on physiological and/or physical traits (e.g., iris, fingerprint, signature, voice, and face)[1−5] to recognize individuals. Among these well-known traits, faces, fingerprints, and voices are commonly used for human identification; such an approach suffers from implicit drawbacks, including the fact that the aforementioned attributes may change as a subject ages, and the possibility that individual characteristics may be artificially duplicated[6]. By contrast, the iris texture approach is unique, stable, and non-invasive[1]. Literatures revealed that iris texture remained stable over almost an entire lifetime, except for some small variations occurring in a person's early life[7,8]. A typical iris biometric system involves image acquisition, iris segmentation, feature extraction, and matching and recognition[6,7,9]. Notably, the overall accuracy of the system depends on the efficient implementation of each of the aforementioned modules. In contrast to this idea, our study emphasizes the efficient implementation of the segmentation module, which generally involves two basic functions: i) localizing the inner (pupillary) and outer (limbic) boundaries of the iris; ii) detecting and removing noise, such as eyelashes, eyelids, and reflections.

To localize iris boundaries, most state-of-the-art iris localization techniques[1] involve Daugman's integro-differential operator (IDO)[7] and/or circular Hough transform (CHT)[8]. IDO directly operates on the input eye image to localize the desired boundary with circle approximation. A drawback to this method is that it is sensitive to saturated light reflections and exhibits poor performance. Visible wavelength (VW) data contain relatively more light reflections. CHT requires an edge map of the eye image to localize the desired boundary, but this technique consumes a relatively large amount of memory[6,9]. Some recent techniques[1] feature a number of simple strategies[1,2] (e.g., image projection functions) for localizing seed pixels with the lowest gray intensity in the iris/pupil region; these strategies effectively work for near-infrared (NIR) data but may fail for VW data. This discrepancy occurs because VW images contain more reflections than do NIR images, which may result in artificial bias with regard to gray intensity in the iris/pupil region. In Ref. [10], the authors presented a new iris recognition method that was tested on both NIR and VW data. The proposed method involves localizing pupillary boundary using an adaptive threshold and a spectrum image (i.e., Euclidean distance transform (EDT) image), localizing limbic boundary using a circular summation of intensities, and extracting features using a speeded-up robust features approach. The main drawback of this method lies in extracting the pupillary boundary, whose center is taken as the peak point in the EDT image and whose radius is taken from the binary image. This approach may work better for relatively good NIR data but may work poorly for VW data because the pupil region in VW data may not always be the darkest part given light reflections and similar problems. Other histogram and thresholding-based techniques[2,6,9] may also fail for VW data because these techniques rely on an image's gray-level intensity.

To solve the highlighted issues, we formulate a scheme as follows. Firstly, we minimize the effect of light reflections in an eye image using a median filter, binarize the resultant preprocessed image using Otsu's thresholding[11,12], and then compute its EDT. The pixel with the highest value in the EDT image generally represents a point (seed pixel) in the pupil/iris region. Secondly, We propose a new circular pixel counting scheme (CPCS) for localizing limbic boundary, with its center confined to a small region in the middle of the seed pixel.

Finally, we reuse the CPCS to localize pupillary boundary within the iris region.

The second state of our segmentation module involves the removal of noise, such as eyelashes and eyelids. Although eyelashes and reflections also distort iris texture, the main sources of occlusion are the eyelids, particularly the upper eyelid, which relatively covers a larger portion of iris texture. Masek *et al.*[13,14] used line Hough transform, which may remove a significant amount of useful iris texture or may leave some occlusions undetected because an eyelid best fits parabolic arcs. In Refs. [8, 15, 16], the authors used the parabolic Hough transform to localize eyelids—an approach that requires a relatively large amount of memory. An important issue in CHT- and IDO-based eyelid localization is that these entail relatively more time if the iris in an eye image is not horizontally oriented. To localize eyelids in a rotated iris image, CHT and IDO use a fourth parameter, that is, the angle of rotation—an expensive computation process. Other researchers[1] also used the parabolic version of IDO to localize eyelids; this approach is sensitive to reflections. To solve these issues, we propose a reliable parabolic pixel counting scheme (PPCS) for localizing eyelids in the normalized iris region. We also adaptively detect eyelashes, reflections, and skin parts.

As shown in Fig. 1(a), we firstly pass the gray format $g(x, y)$ of the input eye image through a median filter (window size, (15×15)) to smooth rapid gray-level variations; with median filtering, any small object in a digital image, whose area is less than one-half the filter's area, is removed[17]. Next, we binarize the resultant preprocessed image $r(x, y)$ using a well-known Otsu's thresholding method[12] (Fig. 1(b)). In the optimal mode, however, Otsu's thresholding method categorizes the gray-level intensities of all the pixels in $r(x, y)$ into two unique classes and then computes an appropriate threshold for binarizing the pixels. The resultant binary image is $b(x, y)$ (Fig. 1(c)). We likewise fill holes in $b(x, y)$ by a bilinear interpolation technique[18]. A hole is a region of dark pixels that are completely surrounded by light pixels and inaccessible from image dimensions. We then compute the EDT[19] of $b(x, y)$. For each pixel in $b(x, y)$, EDT assigns a number that represents the distance between that pixel and the nearest non-zero pixel of $b(x, y)$. Figure 1(d) shows that the resultant image is $\mathrm{dt}(x, y)$.

Let (S_x, S_y) be the coordinates of a pixel (seed pixel) with a maximum value in $\mathrm{dt}(x, y)$. We then mark a circular disk (region of interest (ROI), green) with a 40-pixel radius centered on (S_x, S_y), as shown in Fig. 2(a). Moreover, let $(\bar{x}_i, \bar{y}_i)$ for i=1, 2, 3, $\cdots$, N_i be the coordinates of the pixels in this ROI, where N_i is the total number of pixels. Then, we use the Canny edge-detecting operator[20] to generate edge map $e(x, y)$ for $r(x, y)$ (Fig. 2(b)). Subsequently, we use Algorithm $\sim$1, which is based on the CPCS strategy, to localize the limbic boundary with circle approximation (x_c, y_c, r_c), where (x_c, y_c) and r_c represent the center and radius parameters, respectively.

Algorithm $\sim$1 functions as follows. In steps 2–4, this algorithm draws a band of circles in $e(x, y)$, which is centered on $(\bar{x}_i, \bar{y}_i)$ for $i \in$(1, 2, 3, $\cdots$, N_i), and has radii-range $(r_{i1} : r_{i2})$ (Fig. 2(b)). r_{i1} and r_{i2} represent the

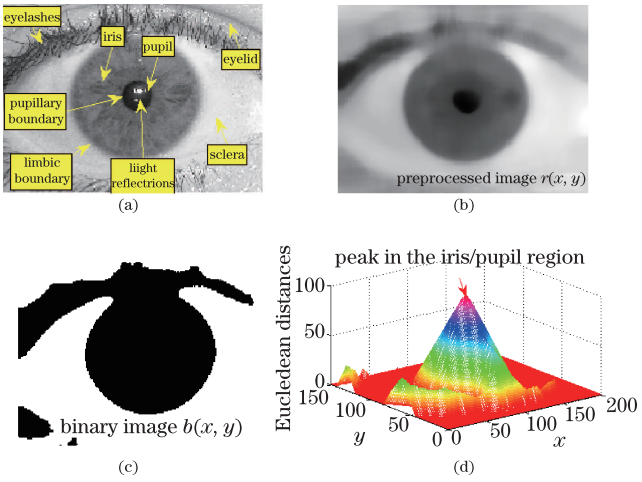

Fig. 1. (a) $g(x, y)$ showing different eye parts. (b) Preprocessed image $r(x, y)$. (c) Holes-filled binary image $b(x, y)$ and (d) EDT image $\mathrm{d}t(x, y)$. Eye image is taken from the UBIRIS V1[17].

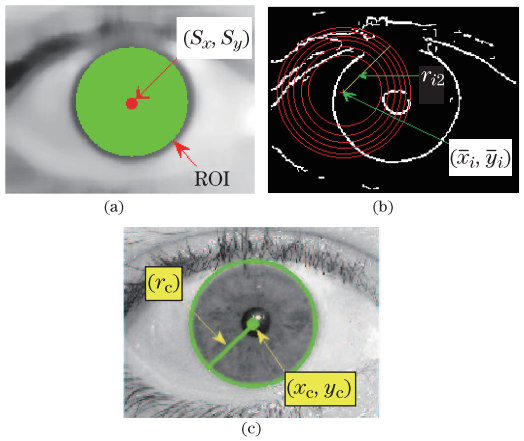

Fig. 2. (a) ROI centered on (S_x, S_y) in $r(x, y)$. (b) Edge-map $e(x, y)$ showing a sample of the band of circles (for simplicity, some of the circles are drawn) centered at a ROI-point $(\bar{x_1}, \bar{y_1})$. (c) Limbic boundary localized with (x_c, y_c, r_c).

lower and upper radii limits of this band, respectively. This kind of radii-range is most often used in all state-of-the-art iris localization methods[1,7,8,13,18], which are based on IDO and/or CHT operators. For generalization, however, we experimentally set it as $(r_{i1} : r_{i2})$=(0.1 w:0.4 w), where w represents the width of $g(x, y)$. For each circle in the current band, the total pixels (which are covered by their boundaries in $e(x, y)$) are counted and then the parameter vector $(\bar{x}_i, \bar{y}_i, \tilde{r})$ of a circle with the maximum count is registered into (X, Y, R). The maximum count (i.e., Count) is also registered into Acc. This process is repeated for all $(\bar{x}_i, \bar{y}_i)$ for i=1, 2, 3, $\cdots$, N_i in the ROI. Given that $e(x, y)$ may contain edges for irrelevant regions, such as skin wrinkles and eyebrows, a peak in Acc is therefore not necessarily a true candidate of limbic boundary. To solve this issue, steps 5 and 6 verify the average gray-level intensity of the iris region in $r(x, y)$. We firstly extract maximum (ψ) and its associated location (x_c, y_c, r_c) from Acc and (X, Y, R), and then compute the average (mean) gray-level value (μ_c) of the pixels in a circular region described by (x_c, y_c, r_c) in

Algorithm∼1: Limbic Boundary Demarcation Using CPCS Strategy

Input-step: $(r_{i1} : r_{i2})$, $e(x, y)$, and $(\bar{x}_i, \bar{y}_i)$,
for i=1, 2, 3, $\cdots$, N_i.
  Step 1: Ind←1; ϕ ←0; and Acc←0.
for $i = (1 : N_i)$ do
  for $\tilde{r} = (r_{i1} : r_{i2})$ do
  Step 2: Count←0.
    for $\theta = (0 : 2\pi)$ do
  Step 3: $(x_\theta, y_\theta) \leftarrow (\bar{x}_i + \tilde{r}\cos\theta, \bar{y}_i + \tilde{r}\sin\theta)$,
      Count←(Count+$e(x_\theta, y_\theta)$).
    end
  Step 4: if ($Count > \phi$) then
      $\phi \leftarrow Count$; $(X_{\rm ind}, Y_{\rm ind}, R_{\rm ind}) \leftarrow (\bar{x}_i, \bar{y}_i, \tilde{r})$;
    $\rm Acc_{ind}$ ←Count; and Ind← $(ind + 1)$.
    end
  end
end
  Step 5: extract a peak (ψ) and its corresponding location (x_c, y_c, r_c) from
  Acc and (X, Y, R), respectively; next, compute mean (μ_c) of pixels
  lying within a circular region described by (x_c, y_c, r_c) in $r(x, y)$.
  Step 6: if ($\mu_c < \mu_r$) then // where μ_r is the mean of $r(x, y)$
    go to output-step;
  else if ($\psi > \psi_o$)
    suppress ψ to zero in Acc and repeat process from step 5;
      else
        abort the iris localization process.
    end
  Output-step: (x_c, y_c, r_c) is the accurate parameter vector of limbic boundary.

$r(x, y)$. If $(\mu_c < \mu_r)$, then (x_c, y_c, r_c) represents the accurate parameter vector of the limbic boundary. Otherwise, it suppresses ψ to zero in Acc. This process is repeated from step 5 provided that Acc is not scanned below its lower limit (ψ_o), which is experimentally set to 10. However, if Acc is scanned below its lower limit, then iris localization is aborted because of low quality (e.g., closed eye).

Figure 3 illustrates numerous peaks in Acc; these peaks are candidates for the circles and/or circular arcs in $e(x, y)$. These multiple candidates exist because of the probable circular arcs in $e(x, y)$, which are usually caused by skin wrinkles, eyebrows, and eyelids. The proposed CPCS strategy embedded in Algorithm ∼1 considers these arcs because of the tolerance of this strategy to broken contours of circular shapes in $e(x, y)$. However, robustly localizing a target object using a coarse-to-fine strategy is a positive sign because this localization is adopted in steps 5 and 6 of Algorithm ∼1. Figure 2(c) shows the iris boundary localized using Algorithm ∼1.

After localizing the limbic boundary, we then demarcate the pupillary boundary by circle approximation (x_p, y_p, r_p), where (x_p, y_p) and r_p are the center and radius of the circular pupillary boundary, respectively. To begin, we first mark a ROI (green), centered on iris center (x_c, y_c), with a 10-pixel radius within the coarse iris region (Fig. 4(a)). N_p is the total number of pixels in this ROI. The radii-range for the circular pupillary boundary is experimentally set as $(r_{p1} : r_{p2})$=(5:0.6 r_c). For clarity, r_{p2} is marked with a blue circle in Fig. 4(a). Corresponding to $(r_{p1} : r_{p2})$, we mask out the edges from $e(x, y)$(Fig. 4(b)) and reuse the CPCS strategy embedded in Algorithm ∼1 (except for steps 5 and 6) to extract (x_p, y_p, r_p). Note that instead of applying steps 5 and 6, we simply extract (x_p, y_p, r_p) from (X, Y, R) corresponding to a peak value in Acc. Figure 4(c) shows the pupillary boundary marked with (x_p, y_p, r_p).

After localizing the iris boundaries, the next step in iris segmentation is noise detection and exclusion. Prior to removing noise, we firstly apply Daugman's rubber sheet model[7,13] to normalize annular iris region $i(x, y)$ (Fig. 4), which is sandwiched between the pupillary and limbic boundaries. The physical geometry of the iris of a specific subject may vary across various eye images for the following reasons[7]: different image acquisition setups, different distances between a subject's eye and the camera, and pupil dilation. To minimize the effects of these defects, the rubber sheet model is used to transform $i(x, y)$ into a fixed rectangular strip called polar image $p(\bar{r}, \bar{\theta})$. This model transforms each point in iris region (x, y) to a pair of polar coordinates $(\bar{r}, \bar{\theta})$, where $\bar{r}$ and $\bar{\theta}$ represent the radial and angular resolutions, which are experimentally set to 60 and 360 pixels, respectively.

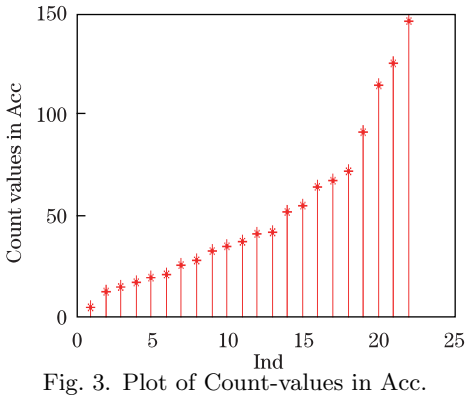

Fig. 3. Plot of Count-values in Acc.

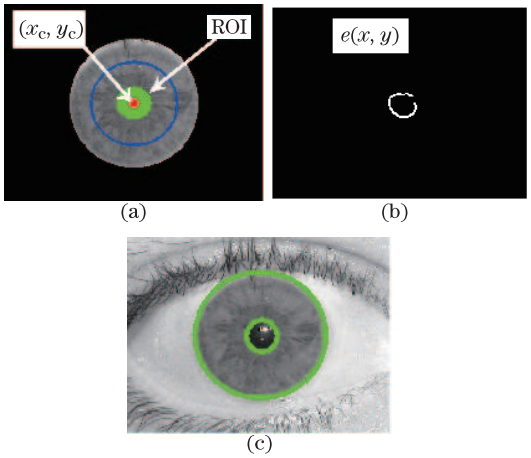

Fig. 4. (a) ROI, centered on the iris center (x_c, y_c), is marked within the coarse iris region. (b) Edge-map (x, y) containing edges corresponding to the pupil region only. (c) Pupillary boundary localized with (x_p, y_p, r_p).

Figure 5 shows the normalization of $i(x, y)$ in a new gray-level eye image $g(x, y)$, where the iris region is occluded by the lower and upper eyelids.

To localize the eyelids, we firstly pass $p(\bar{r}, \bar{\theta})$ through a median filter with a window size of (5×5) to smooth rapid gray-level variations, such as hair threads. In this study, we adopt a smaller window size that will keep the edges of the eyelids relatively intact. The resultant edge map $E(x, y)$ is shown in Fig. 6(b). Next, we bisect $p(\bar{r}, \bar{\theta})$ into two halves—the left and right halves—which generally contain the upper and lower eyelids, respectively (Fig. 6(a)). These eyelids are best fit with the following parabolic expression:

$$(x - h)^2 = \lambda(y - k), \tag{1}$$

where (h, k) and λ represent the vertex and curvature of a parabola, respectively. To localize the accurate location (h, k, λ) of a potential parabolic arc in $p(\bar{r}, \bar{\theta})$, we should vary only λ and (h, k). Therefore, we experimentally set λ as $(\lambda_1 : \lambda_2)$=(20:0.5 L), where L is the length of $p(\bar{r}, \bar{\theta})$, and λ_1 and λ_2 represent the lower and upper curvature limits of this band, respectively. The red regions in Fig. 6(a) represent $(\lambda_1 : \lambda_2)$. Similarly, for vertex shifting, we set two ROIs, namely, $\text{roi}_1 = (h_{1\,\min} : h_{1\,\max}, k_{1\,\min} : k_{1\,\max})$ and $\text{roi}_2 = (h_{2\,\min} : h_{2\,\max}, k_{2\,\min} : k_{2\,\max})$, in the left and right halves of $p(\bar{r}, \bar{\theta})$, respectively. The two regions are experimentally set as $\text{roi}_1 = (1 : w,\ 0.25\ L \pm 10)$ and $\text{roi}_2 = (1 : w,\ 0.75\ L \pm 10)$, where w represents the width of $p(\bar{r}, \bar{\theta})$. Note that roi_1 and roi_2 are marked with yellow bars centered at 0.25, and 0.75 L, respectively, in $p(\bar{r}, \bar{\theta})$ (Fig. 6(a)).

We now use Algorithm ∼2, which is based on PPCS, to localize the eyelid in the left half of $p(\bar{r}, \bar{\theta})$ as follows. In steps 2–4, Algorithm ∼2 draws a band of parabolic arcs in $E(x, y)$. The band is centered on $(h_i, k_i) \in \text{roi}_1$ for $i \in (1, 2, 3, \cdots, N_e)$, with a curvature range of $(\lambda_1 : \lambda_2)$ (Fig. 6(b)), where N_e is the total number of pixels in roi_1. The total pixels covered by the boundary of each parabolic arc in $E(x, y)$ are counted, and the parameter vector (h_i, k_i, λ) of the parabola with the maximum count is registered into (H, K, P); the maximum count of that parabola is also registered as Acc ← Count. This process is repeated for all (h_i, k_i) for i=1, 2, 3, $\cdots$, N_e

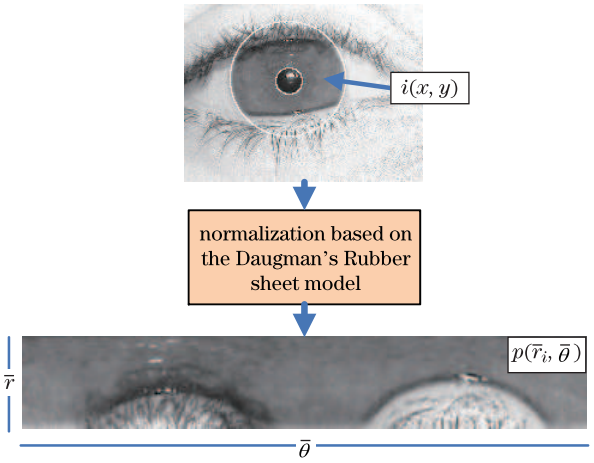

Fig. 5. Normalization of $i(x, y)$ to $p(\bar{r}, \bar{\theta})$.

Algorithm ∼2: Eyelid Detection Using PPCS Strategy

Input-step: $E(x, y)$, $(\lambda_1 : \lambda_2)$, roi_1, and roi_2.
  Step 1: Ind←1; ϕ ←0; and Acc←0.
for i=(1:N_e) do
  for $\lambda = (\lambda_1 : \lambda_2)$ do
  Step 2: Count←0.
    for $x = (x_{\min} : x_{\max})$ do // $(x_{\min} : x_{\max})$= $(1 : 0.5\,L)$
  Step 3: $y \leftarrow \lambda^{-1}(x - h_i)^2 + k_i$,
    Count← (Count+$E(x, y)$).
  end
  Step 4: if (Count> ϕ) then
    ϕ ←Count; $(H_{\text{ind}}, K_{\text{ind}}, P_{\text{ind}}) \leftarrow (h_i, k_i, \lambda)$;
    Acc_{ind} ←Count; Ind←(ind+1).
  end
  end
end
  Step 5: extract a peak (ψ) and its corresponding location (h, k, λ)
    from Acc and (H, K, P), respectively.
Output-step: (h, k, λ) is the accurate parameter vector.

in roi_1. Finally, a maximum (ψ) and its associating location (h, k, λ) from Acc and (H, K, P), respectively, are extracted, where (h, k, λ) represents the accurate location of the eyelid in the left half of $p(\bar{r}, \bar{\theta})$. The eyelid in the fight half is localized in a similar manner. Figure 6(c) shows the resultant eyelid mask $Em(x, y)$ obtained using Algorithm ∼2.

After generating the eyelid mask, we adaptively develop reflection mask $Rm(x, y)$ as

$$Rm(x, y) = \begin{cases} 1, & \text{if } p(\bar{r}, \bar{\theta}) < T_1 \\ 0, & \text{otherwise} \end{cases} \quad \text{with } T_1 = 0.85(\alpha_{\text{u}}), \tag{2}$$

where α_{u} represents the saturated gray-level value[18] of the top 1% of all the gray values in $g(x, y)$; T_1 is experimentally established. Figure 6(d) shows $Rm(x, y)$, which also contains the eyelid regions. The skin regions are detected and removed even if the eyelid detection scheme fails because the gray-level intensity of the skin parts is generally higher than the iris texture. We then adaptively develop eyelash mask $em(x, y)$, thus:

$$em(x, y) = \begin{cases} 1, & \text{if } p(\bar{r}, \bar{\theta}) < T_2 \\ 0, & \text{otherwise} \end{cases} \quad \text{with } T_2 = 1.2(\alpha_{\text{l}}), \tag{3}$$

where α_l represents the saturated gray-level value[18] of the bottom 1% of all the gray values in $g(x, y)$. T_2 is also experimentally established. Figure 6(e) shows $\text{em}(x, y)$. Finally, overall noise mask $n(x, y)$(Fig. 6(f)) is obtained as

$$n(x, y) \leftarrow [Em(x, y)|Rm(x, y)|em(x, y)]. \tag{4}$$

We use one-dimension (1D) log Gabor filter $G(f)$ (Fig. 7(a))[7,13] instead of the Gabor filter[13] to extract features from $p(\bar{r}, \bar{\theta})$ because Gabor filters have a direct

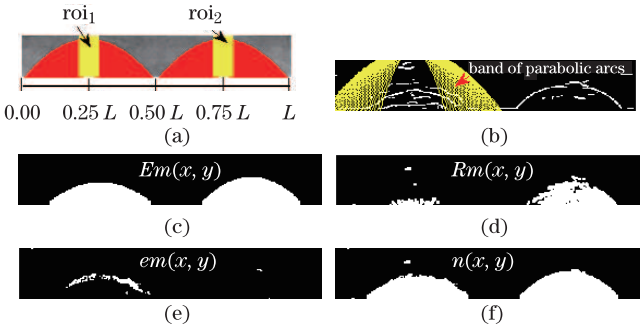

Fig. 6. (a) roi_1 and roi_2 which centered at 0.25 L and 0.75 L in the left and right halves of $p(\bar{r}, \bar{\theta})$, respectively. Similarly, the convex-type region in each of these halves represent $(\lambda_1 : \lambda_2)$. (b) Edge-map $E(x, y)$ showing a band of parabolic arcs (for simplicity, some of the parabolic arcs are drawn) located at an roi-point (h_i, k_i) in the Left-half. (c)-(f) Noise masks: $Em(x, y)$, $Rm(x, y)$, $em(x, y)$, and $n(x, y)$, respectively.

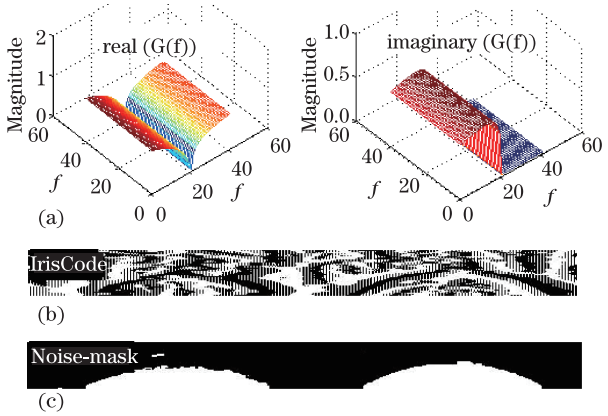

Fig. 7. (a) Real and imaginary part of $G(f)$. (b) the IrisCode and (c) its corresponding Noise_mask, respectively.

current component whenever bandwidth is greater than 1 octave. This problem does not occur in $G(f)$[13], defined as

$$G(f) = \exp\left\{\frac{-\left[\lg\left(\frac{f}{f_o}\right)\right]^2}{2\left[\lg\left(\frac{f}{f_o}\right)\right]^2}\right\}, \tag{5}$$

where f_0 and σ represent the central frequency and bandwidth of $G(f)$, respectively; these are experimentally set as $\sigma = 0.025$ and $f_0 = 0.05$, respectively. We then use the same procedure as that in Ref. [13] to generate the IrisCode (also called the template) and its corresponding Noise_mask using $p(\bar{r}, \bar{\theta})$ and $n(x, y)$, respectively. The resultant IrisCode and its corresponding Noise_mask are shown in Figs. 7(b) and (c), respectively.

For matching, we use the Hamming distance (HD)[13] as a recognition metric. Only the bits in the probe (X) (target) and gallery (Y) (stored in the database) IrisCodes match and correspond to zero bits in their corresponding noise masks; that is, (X_{n}) and (Y_{n}), respectively.

$$\text{HD} = \frac{1}{\overline{N} - \sum_{k=1}^{\overline{N}} (X_{\text{n}k}(\text{OR})Y_{\text{n}k})} \cdot \sum_{j=1}^{\bar{N}} (X_j(\text{XOR})Y_j)(\text{AND})(X'_{\text{n}j}(\text{AND})Y'_{\text{n}j}), \tag{6}$$

where $\overline{N}$ is the number of bits per IrisCode, and X'_{n} and Y'_{n} are the complements of X and Y, respectively.

The rotational inconsistency of the iris in an eye image also affects intraclass comparisons. When matching two IrisCodes, this issue is solved by shifting one IrisCode and its corresponding Noise_mask to the left and right, respectively, bit wise. A number of HD values are then calculated from the successive shifts. This bit-wise shifting in the horizontal direction corresponds to the rotation of the original iris region by an angle given by the angular resolution used. During the matching process, this shifting strategy[7] corrects for any misalignments in the normalized iris pattern caused by rotational differences during imaging. Finally, the lowest value of HD, which corresponds to the best match between the probe and gallery IrisCodes, is selected.

To test the authenticity of the system, we simulate the proposed technique using: i) MATLAB V7; ii) a laptop equipped with CORETM i5, a 2.4-GHz CPU, and 2-GB RAM; iii) standard iris databases UBIRIS V1.0[21] and CASIA-Iris-Interval[22]. UBIRIS V1.0 has 1 877 eye images acquired from 241 individuals in two sessions using VW light sources. Only five images of the same eye are taken from each subject. Each image is stored in an 8-bit JPEG gray format with a resolution of 200×150 pixels. This database is specifically designed to simulate less-constrained iris biometric systems[21], and contains non-ideal issues, such as eyelids, eyelashes, light reflections, and low contrast. CASIA-Iris-Interval comprises 2 639 eye images acquired from 249 individuals using a CASIA-made close-up iris camera equipped with a circular NIR LED array. Most of the images were acquired in two sessions. In each image, the pupil contains eight white dots arranged in a circle. The image resolution is 320×280 pixels. Figure 8 shows sample images (taken from the databases) that are accurately localized using the proposed scheme.

To compute recognition accuracy, we develop intra- and inter-class comparisons as follows. A total of 249 main folders can be found in CASIA-Iris-Interval, each

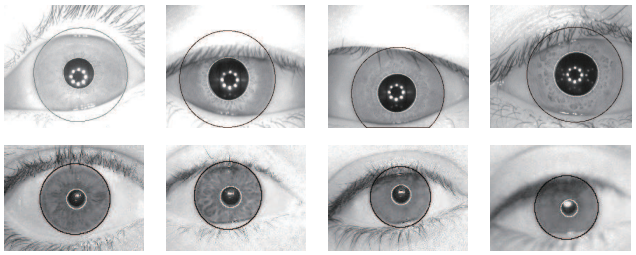

Fig. 8. First and second rows show some eye images taken from the CASIA-Iris-Interval and UBIRIS V1.0, respectively, which were accurately localized by the proposed method.

containing two sub-folders. Numerous sub-folders in this database are empty because they were not filled with eye images by the concerned developers. For this reason, we select only 121 main folders from CASIA-Iris-Interval, so that each sub-folder contains at least four images. Given at most four images in each sub-folder, we perform 1,452 intra-class and 232,320 inter-class comparisons. Similarly, UBIRIS V1.0 (session 1 only) has 241 folders, each containing five images, amounting to 2,410 intra- and 289,200 inter-class comparisons.

Figure 9 shows the intra- and inter-class distributions for the CASIA and UBIRIS iris databases. Corresponding to these distributions, Figs. 10(a) and (b) show the equal error rate (EER) versus HD and the genuine accept rate (GAR) versus false accept rate (FAR), respectively. EER is the error rate where both the FAR and false reject rate (FRR) are equal. In this study, FAR is the error rate of accepting an unenrolled person as an enrolled one, FRR is the error rate of rejecting an enrolled person as an unenrolled one, and GAR is the ratio of enrolled persons accepted by the system to the total number of enrolled persons entered into the system. For optimum performance, FRR and FAR should be as small as possible. For more information, the reader is referred to Refs. [7,13].

Table 1 compares the accuracy of the proposed technique with those of recent iris recognition methods as applied to the UBIRIS and CASIA databases. The proposed technique outperforms UBIRIS because of the relatively large distance between the means of the intra- and inter-class distributions of the latter. By contrast, the distance between the means of the intra- and inter-class distributions of the CASIA database is smaller (Fig. 9).

We also resolve the computational complexity of the proposed technique as follows. Let $C_{\rm t}$ be the total computational complexity of the proposed algorithm. It is defined as

$$C_{\rm t} = C_{\rm pre} + C_{\rm iris} + C_{\rm pupil} + C_{\rm eyelid} + C_{\rm refl\_eyelash} + C_{\rm iriscode}, \quad (7)$$

where $C_{\rm pre}$ is the total number of steps implemented by the median filter, Otsu's thresholding, and EDT. These processes are simple and therefore do not involve any iterative strategies. $C_{\rm iris}$ represents the total number of steps required to localize the limbic boundary by using the CPCS strategy (iterative); it is defined as $C_{\rm iris} = t_{\rm o} + N_i(r_{i2} - r_{i1})$, where $t_{\rm o}$ represents the number of steps required to generate $e(x, y)$. Similarly, $C_{\rm pupil}$ is

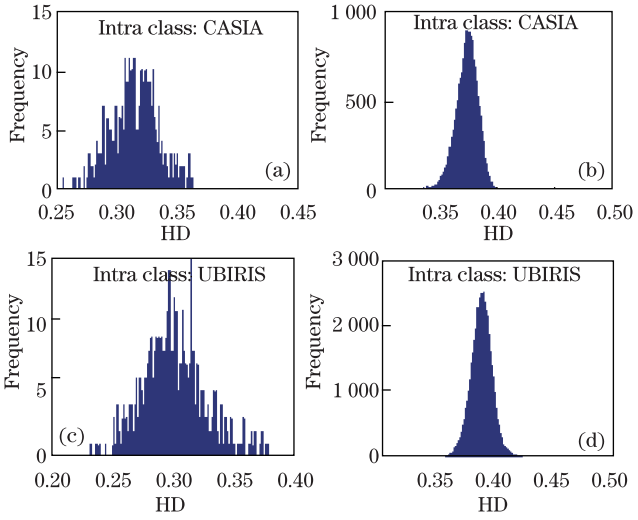

Fig. 9. First and second rows show the intra and inter class distributions for the CASIA and UBIRIS databases, respectively.

the total number of steps required to localize the pupillary boundary by reusing the CPCS strategy; it is defined as $C_{\rm pupil} = N_{\rm p}(r_{\rm p2} - r_{\rm p1})$. $C_{\rm eyelid}$ is the total number of steps required to localize both eyelids using the PPCS strategy; it is defined as $C_{\rm eyelid} = 2N_{\rm e}(\lambda_2 - \lambda_1) + t_1$, where t_1 is the number of fixed steps required to generate $E(x, y)$. $C_{\rm refl\_eyelash}$ represents the total number of steps required to compute $Rm(x, y)$ and $em(x, y)$, and finally, $C_{\rm iriscode}$ is the total number of steps required to generate the desired IrisCode and its corresponding Noise_mask. $C_{\rm t}$ can be minimized in the following ways: i) by processing a scaled-down version of $g(x, y)$; ii) by using the alternate locations in all the ROIs; iii) by reducing the radii-ranges for the inner and outer contours of the iris and the curvature range of the parabolic arcs. On average, the optimized code of the proposed algorithm takes 0.8 and 0.5 s to generate an IrisCode and its Noise_mask for $g(x, y)$, respectively, from the CASIA-Iris-Interval and UBIRIS V1.0, respectively.

In conclusion, we present a robust and reliable iris biometric system for processing VW and NIR data. The proposed system involves localizing a seed pixel in the pupil/iris region using an efficient method that includes median filtering, Otsu's thresholding, and EDT. We also propose a novel pixel counting scheme that can localize both the iris contours and eyelids with great precision; that is, the pupillary and limbic boundaries are localized

Table 1. Recognition Accuracy Comparison

Method	UBIRIS V1.0			CASIA-Iris-Interval		
	FAR(%)	FRR(%)	Accuracy(%)	FAR(%)	FRR(%)	Accuracy(%)
Belcher & Du[23]a(SIFT: Fixed strip)	13.08	32.91	77.00	7.16	19.64	86.59
Belcher & Du[23]a(SIFT: Adaptive strip)	6.16	0.00	96.91	3.74	4.34	95.95
Mehrotra *et al.*[10](SURF: Fixed strip)	7.34	4.11	94.27	3.39	4.78	95.91
Mehrotra *et al.*[10](SURF: Adaptive strip)	5.02	1.80	96.58	1.55	3.80	97.32
Kang *et al.*[24]	–	–	–	0.05	–	98.60
Proposed	0.10	0.00	99.6	0.047	0.011	99.40

a: Results are taken from Ref. [10]

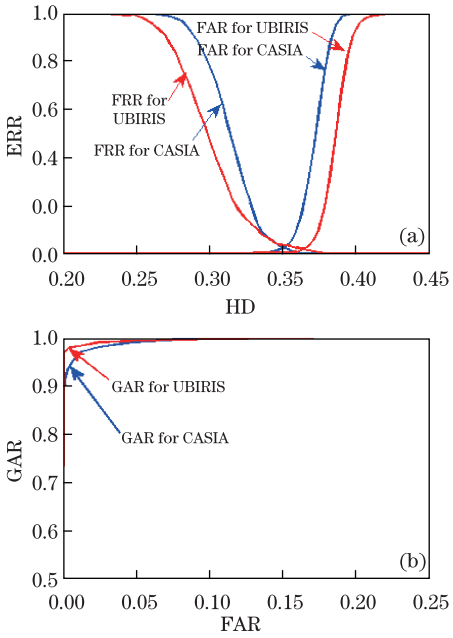

Fig. 10. (a) EER versus HD. (b) GAR versus FAR

using CPCS, and eyelids are localized using PPCS. In addition, light reflections, skin parts, and eyelashes are adaptively marked, thereby considerably enhancing overall system accuracy. Finally, the recognition accuracy of the proposed technique for both VW and NIR databases is better than those of recent state-of-the-art iris recognition techniques.

We acknowledge the COMSATS Institute of Information Technology (CIIT), Islamabad Campus Pakistan for its in-house PhD program. We also thank the University of Beira Interior and the Chinese Academy of Sciences for granting free access to their respective databases.

References

1. K. W. Bowyer, K. Hollingsworth, and P. J. Flynn, Comput. Vis. Image Und. **110,** 281 (2008).
2. F. Jan, I. Usman, and S. Agha, Chin. Opt. Lett. **10,** 111501 (2012).
3. F. Wang, X. Yao, and J. Han, Chin. Opt. Lett. **6,** 824 (2008).
4. X. Ye, Z. He, and Z. Zhao, Chin. Opt. Lett. **6,** 487 (2008).
5. H. Proenca and G. Santos, Comput. Vis. Image Und. **116,** 167 (2012).
6. T. M. Khan, M. A. Khan, S. A. Malik, S. A. Khan, T. Bashir, and A. H. Dar, Opt. Lasers Eng. **49,** 177 (2011).
7. J. G. Daugman, IEEE Trans. Pattern Anal. Machine Intell. **15,** 1148 (1993).
8. R. P. Wildes, Proc. IEEE **85,** 1348 (1997).
9. M. T. Ibrahim, T. M. Khan, S. A. Khan, M. A. Khan, and L. Guan, Opt. Lasers Eng. **50,** 645 (2012).
10. H. Mehrotra, P. K. Sa, and B. Majhi, Math. Comput. Model. **58,** 132 (2013).
11. Z. Lin and H. Yu, Procedia Environ. Sci. **8,** 352 (2011).
12. N. Otsu, IEEE Trans. Sys. Man. Cyber. **9,** 62 (1997).
13. L. Masek, "Recognition of Human Iris Patterns for Biometric Identification", B.Sc. Thesis (University of Western Australia, 2003).
14. N. B. Puhan, N. Sudha, and A. S. Kaushalram, Signal Image Video Process. **5,** 105 (2011).
15. B. J. Kang and K. R. Park, Pattern Recogn. Lett. **28,** 1630 (2007).
16. T. H. Min and R. H. Park, Pattern Recogn. Lett. **30,** 1138 (2009).
17. R. C. Gonzalez and R. E.Woods, *Digital Image Processing* (Prentice Hall, Upper Saddle River, 2007).
18. F. Jan, I. Usman, and S. Agha, Digital Signal Process. **22,** 971 (2012).
19. Mathworks, http://www.mathworks.com/ (March 19, 2013).
20. J. Canny, IEEE Trans. Pattern Anal. Machine Intel. **PAMI-8,** 679 (1986).
21. UBIRIS iris database, http://iris.di.ubi.pt/ (March 19, 2013).
22. CASIA iris database, http://www.idealtest. org/ findTotalDbByMode. do? mode=Iris (March 19, 2013)
23. C. Belcher and Y. Du, Opt. Lasers Eng. **47,** 139 (2009).
24. B. J. Kang, K. R. Park, J. H. Yoo, and K. Moon, Opt. Eng. **49,** 067001 (2010).