Fast and Efficient Iris Image Segmentation

Lee Luan Ling   Daniel Felix de Brito*

Department of Communications, School of Electrical and Computer Engineering, State University of Campinas, Campinas, SP, Brazil

Received 2 Apr 2010; Accepted 25 Sep 2010; doi: 10.5405/jmbe.769

Abstract

Iris segmentation is one of the crucial operations involved in iris recognition. Accurate iris segmentation is fundamental for the success and precision of the subsequent feature extraction and recognition, and consequently the high performance level of the iris recognition system. Most iris segmentation approaches proposed in the literature require highly complex-exhaustive search and learning of many modeling parameters and characteristics, which prevents their effective real-time applications and makes the system highly sensitive to noise. This paper presents a fast and efficient iris segmentation methodology to address relatively simple solutions to these problems. Three major procedures involved in the proposed iris segmentation approach, namely pupil detection, limbic boundary localization, and eyelid and eyelash detection, were carefully designed in order to avoid unnecessary and redundant image processing, and most importantly, to preserve the integrity of iris texture information. The proposed iris segmentation algorithm has the following particular properties and advantages: (a) avoidance of complex geometric and mathematical modeling; (b) no need of a training phase for algorithm design and implementation; (c) guaranteeing real-time iris segmentation even for iris images with severe occlusions; (d) high accuracy in iris segmentation and therefore low segmentation error rate. Experimental results, reported in this paper, demonstrate that the proposed iris segmentation algorithm outperforms some well-known methods in both accuracy and processing speed. As a consequence, the iris recognition system that incorporates the proposed iris segmentation algorithm is capable of offering recognition performances comparable with those reported by other state-of-the-art methods.

Keywords: Iris segmentation, Iris recognition system, Biometrics, Image processing

1. Introduction

The term biometrics, in the context of this paper, refers to the identification of an individual based on his/her physical or behavioral characteristics [1]. The keys to the growth of biometric technology are manifold, including (a) the consequence of positive proofs of physical presence and the elimination of the need of remembering passwords or carrying identifiers that could be easily forgotten, borrowed or stolen, and (b) the increasing demands of security in quotidian environments, especially in personal identification. Iris recognition has rapidly become one of the most researched biometric topics due to its high potentiality in practical applications and is probably one of the most reliable biometric identification methods [1-6]. Iris biometrics makes use of the highly rich and discriminative texture information contained in the annular region between the dark pupil and white sclera.

Iris segmentation is one of several major processing steps in an iris recognition task. The main goal of this iris segmentation step is to determine the valid region of the iris for recognition purposes [7]. Basically this region is delimited by the pupil and sclera. However, frequently iris texture regions are occluded by upper and lower eyelids, eyelashes, light reflections, shadows, etc. Therefore, iris segmentation also includes localizing the eyelids and eliminating the effect of occlusions caused by the eyelashes, shadows and light reflections. The quality of the adopted iris segmentation method affects directly the overall iris recognition performance. On one hand, it is crucial for high quality of extracted iris features used for recognition, while, on the other hand, it is a determinant of biometrically real-time response due to fact that it is the most time-consuming module in an iris recognition system.

Among many research works focusing on iris segmentation approaches, there are two well-known algorithms, reported, respectively, by Daugman [2] and Wilders [8]. Daugman, in [2], applied an integrodifferential operator to delimitate the circular boundaries of irises, while Wilders, in [8], used Hough transforms to locate iris boundaries. Both algorithms grant good performances but computationally are highly time-consuming. In addition, the two segmentation methods do not take into account the obstructions caused by eyelids, reflections, eyelashes and shadows. In order to overcome these challenging drawbacks, initially many subsequent research works attempted to improve or to optimize the methods based on circular modeling [9-12].
Sooner after, deeper investigations focusing on occlusion detection, such as eyelid, eyelash and specular reflection detection, were reported [13,14]. Most approaches for occlusion detection were based on the obstruction objects’ edge detection. More recently, some scientific groups have concentrated their research studies on so-called nonideal iris images, for instance, non-circular and non-concentric iris images [7,15-20].

Based on the above descriptions, there are three important and challenging problems in iris image segmentation: (a) precise determination of pupillary and limbic boundaries, especially for nonideal iris and noisy images; (b) adequate treatment for occlusions caused by reflections, eyelids, eyelashes, etc.; (c) real-time iris recognition in practical applications. Among many tentative approaches for solving these challenging problems, recently He et al. in [21] have proposed a novel algorithm dealing with great parts of these problems with the intention of achieving accurate and fast iris segmentation performance. More precisely, the algorithm intends to solve most occlusion problems, especially on highly noisy images: reflection removal and iris detection, pupillary and limbic boundaries determination, eyelid localization, and eyelash and shadow detection.

With rapid advance of electronic technologies and popularization of biometric applications, the production cost of much specialized biometric equipment, including eye/iris image cameras, has become lower and lower. Simultaneously, the quality of captured eye/iris images has become higher and higher. No doubt better iris image quality can contribute to even higher performance of iris recognition systems, and also simplification of iris segmentation algorithms without compromising the recognition performance. Also, it is worth mentioning that occluded iris areas invariably have their iris texture information lost. Fortunately redundant information provided by the extracted iris features is able to minimize, if not to completely eliminate, the bad effect of iris area occlusion. Therefore, aiming at efficient and real-time iris segmentation, in this paper, we describe a new heuristic approach-based iris segmentation algorithm. The proposed method should attend simultaneously real-time application requirements and high iris image segmentation accuracies, without impacting the desired final iris recognition performance.

2. Materials and methods

As mentioned above, many different iris segmentation methods have been proposed in the literature. Unfortunately, many of them are unsuitable for real time applications, or present relatively high or unacceptable segmentation error rates. Here, error in iris segmentation means failure in correctly localizing and/or isolating iris texture regions for posterior iris feature extraction and recognition. In other words, being unsuccessful in correct iris image segmentation will compromise the overall iris recognition rate. The iris image segmentation algorithm proposed in this paper consists of three major modules, namely pupil detection, limbic boundary localization, and eyelid/eyelash detection. The image processing procedures implemented in these modules were designed in order to be executed sequentially without repetition and redundancy, and consequently minimize the overall processing time. More precisely, the implemented algorithms avoid unnecessary processing over image regions that do not contain relevant information for iris image segmentation, and consequently iris recognition. Figure 1 shows a flowchart of the proposed iris segmentation algorithm with its major processing steps.

![Figure 1. Flowchart of the proposed iris image segmentation algorithm.](image)

**2.1 Pupil detection**

The module of pupil detection was designed to localize pupillary boundaries. More specifically, the module is responsible for performing the following four processing steps, namely: (1) Gaussian filtering, (2) binary image generation (binarization), (3) pupil region detection and spectral removal and (4) pupil center and boundary localization.

A detailed description of the signal processing procedures involved in each step is given below. For the illustration purpose, Fig.2 shows some major involved processing procedures and the obtained results through their respective sample iris image representations.

In this work, we assume that a pupil has circular shape. Thus, under this circular modeling assumption, pupil detection consists of localization of the center of the pupil circle and estimation of the radius of the circular pupillary boundaries. However, due to specular reflections on the pupillary region, the pupil detection (the determination of the center and radius) may not be easily executed. Therefore, a specular reflection removal procedure also is needed and pre-performed. The sub-regions in the pupil area where specular reflections occur should be isolated in advance; then, the specular reflections can be eliminated.

![Figure 2. Flow chart of the pupillary detection algorithm. The major processing procedures, namely, (b) Gaussian filtering, (c) image binarization, (d) pupillary region detection and spectral removal.](image)
2.1.1 Gaussian filtering

The input raw eye image is initially filtered by a Gaussian filter before being submitted to the binarization step. The main objective of this filtering procedure is to reduce or eliminate some possible biased effects provoked by isolated peak noise, generally appearing during eye image acquisition via optical sensors. In other words, the Gaussian filter intends to smooth the acquired eye image and also enhance the image contrast, mainly on the edge. Here, the resulted filtered image is called “the improved image”, which, to our belief, it is more appropriate for segmentation. Equation 1 shows the matrix representation of the adopted Gaussian filter.

\[
F_x = F_y = \begin{bmatrix}
1.3 & 3.2 & 3.8 & 3.2 & 1.3 \\
10.7 & 10.7 & 10.7 & 10.7 & 10.7
\end{bmatrix}
\]

2.1.2 Binary image generation (binarization)

This processing step has the following major objective: to learn the grey level distribution of the improved filtered image in order to be able to enhance the whole pupil area, as well as the sub-regions inside the pupillary regions where specular reflections appear. Most importantly, this learning procedure allows us to obtain an optimal threshold value for binary image generation. The result of this investigation reveals that invariably all improved images present their grey level histograms similar to that shown in Fig. 3. There are three local maximum peaks localized in the following grey level intervals: [0, 100], [100, 250], and [250, 255]. The first, second and third peaks are mainly due to the contribution of pixels localized in the pupil, iris, and sclera regions, respectively. A natural question is how to choose a good threshold value so that the generated binary image is able to retain most pupil area for good segmentation purposes. The binarization operation here mentioned aims at identifying and isolating some regions of interest in the raw input image from its background. The threshold value used for generating the desirable binary image is automatically estimated in advance from the gray level histogram of the improved image. Under the 256 gray level (0-255) standard, we define parameter S as the total amount of pixels in the improved image with their grey levels ranging from 0 to 150. The chosen threshold for binarization has its numerical value corresponding to the upper bound of grey levels that covers exactly the 70% of the total number of the lower grey levels pixels determined by S.

Notice that this 70% numerical value was determined empirically after an extensive experimental investigation by varying the threshold value in the range of 50-100%. We judge that this 70% threshold is able to preserve as much as possible the integrity of the pupil region. The gray level interval (0 to 150) for calculating S was chosen due to the fact that seldom pupil has its brightness superior to 150. Notice that the 70% threshold is not a particular characteristic of an iris image database. On the contrary, it is common feature for any eye image under its gray level representation. Therefore, the 70% threshold was chosen for our image binarization operation. Particularly, for one of our iris image database used in our experimental investigation, called CASIA-IrisV3-Lamp, this 70% threshold has its numerical value equal to the gray level of 100.

2.1.3 Pupil region detection and spectral removal

Although the binary images obtained in the previous step (image binarization) are capable of enhancing most pupil area, they also, however, preserve many irregular and non-connected stains having varying sizes and formats, both inside and outside of the true pupil region (see Fig. 2(c)). Many elements can contribute to this result, including light reflections and shadows. One of two major goals of this step is to identify a unique, valid, and connected pupil region candidate even with the presence of these reflections. In principle, one expects that, once a valid pupil candidate has been identified, the whole pupil and its boundaries can be easily and accurately localized in the next processing steps. Unfortunately, many pupillary boundary detection methods are not indeed efficient if the amount of noise present in eye images is significant, especially due to specular reflections inside the pupil area. Thus, the second goal of the current image processing step is the spectral reflection removal, inside the pupil candidate area.

Pupil region detection is carried out relying on the following premise: the valid pupillary region is the one having the largest connected area. This premise is the consequence of the application of the accurate threshold value, which is also optimal in the sense of the unique, largest, and connected pupillary area, after intensively and recursively being tested and analyzed. For measuring the size of a connected area in a binary image, here we apply the area growing technique proposed in [22].

For reflection noise removal, we apply the following filtering criterion: A pixel (point) is considered as a noise point if (a) it has the background grey level (255) and is removed (changed to grey level 0), and (b) any one of the four neighbor pixels, horizontally and vertically, retains the zero grey level value. Notice that this spectral reflection removal procedure is executed only after the detection of the valid pupillary candidate area. Figure 2(d) shows the obtained pupillary area after with spectral reflection removal, which is ready for pupil localization (pupillary boundary detection) to be executed in the next step.

2.1.4 Pupil center and boundary localization

As mentioned before, in this work, we assume that pupillary boundaries are circularly modeled. Therefore, boundary detection consists of estimating the pupil center...
(X_p, Y_p) and radius (R_p) of the circle based on the binary image obtained in the previous step, i.e., after applying the reflection noise removal procedure. Technically speaking, the boundary detection is composed of two operations: (1) pupil edge detection (the delimited pupil region may not be perfectly circular) and (2) center localization and radius estimation (under circular modeling).

The pupil edge detection has the goal of delimiting edge points of the region of interest obtained in the previous step. For this end, we apply a gradient technique, i.e., filtering through two Sobel operators. Equation 2 depicts the matrix representation of two Sobel operators. Figure 4 shows all detected edge pixels (in white color) obtained from processing the binary image in Fig.2(d). The detected edge basically corresponds to a contour of the valid pupil candidate area. Here, for simplicity, the gradient vector at each edge point is expressed in its polar representation in order to facilitate the operation of boundary localization later.

\[ G_x = G^T_y = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \] (2)

Figure 4. The enhanced edge of the delimited sub-region.

The estimation of the center and radius of the obtained circle is carried out based on the detected edge points. First, these edge points are grouped into one of four quadrant sets according to their corresponding gradient angles: Quadrant I (the [0°, 90°] sector), Quadrant II (the [90°, 180°] sector), Quadrant III (the [180°, 270°] sector), and Quadrant IV (the [270°, 360°] sector). Therefore, the determination of the circle parameters, the pupil center (X_p, Y_p) and radius R_p, consists of searching for two points on the edge belonging to two opposite quadrants (Quadrant I and III or Quadrant II and IV) but separated by a distance approximately equal to the diameter of the circle. If this is the case, as a result, it is expected that the midpoint of these two detected points of the edge indicates the most probable position of the pupil center (X_p, Y_p). For reliable detection, the search, in principle should be carried out through every pair of edge points of the circle. However, in practice, we only need to test the half of the total edge pixels and possibly much less to speed up the estimation process. In other words, the center (X_p, Y_p) is determined by the mid-point with the highest occurrence frequency among all possible combined candidate pairs selected from the other pixels localized on the two opposite quadrants as well as the radius (R_p), by the most probable candidate value.

Notice that for the center (X_p, Y_p) estimation, we apply the concept of Hough accumulator cells, instead of employing any traditional mathematical circumference formula, what implies significant gains in terms of processing time. Also, the proposed method is attractive even when the pupil is altered by dilation or the detected contour is considerably irregular.

2.2 Limbic boundary localization

After pupil detection, the remaining segmentation operation of the iris area that covers the iris texture region consists of limbic boundary and eyelid localizations. This subsection is devoted to a detailed description of left-and-right-side limbic boundary detection. It is interesting to mention that we do not attempt to detect entire limbic boundaries. Instead, we simply estimate part of them: left and right hand side limbic boundaries. The proposed approach, to be described below, proves to be a simple, fast and efficient operation. Most importantly, we are able to preserve all readable iris areas as well as all extractable iris texture information even when limbic boundaries are only partially detected. Such quality and simplicity are consequences of the fact that (a) we assume that the whole limbic boundary can be modeled by a circle with the center (X_p, Y_p) and radius (R_p), and (b) we know the exact localization of the pupil (obtained from the previous processing module). Thus, the limbic boundary localization here is nothing more that the estimation of these two parameters from partially detected iris outskirt. In detail, the procedure for limbic boundary detection can be divided into the following four steps, namely: (1) delimiting of the region of interest; (2) identification of the best median filter; (3) determination of the left and right borders of the iris; (4) verification of the detected iris boundaries.

Before we proceed with a detailed description of each processing step above mentioned, it is worth observing some facts and problems one may encounter in limbic boundary detection which justify our approach. In general, limbic boundaries are not clearly defined due to low contrast between iris texture and sclera. Also, pupillary and limbic boundaries are by no means concentric. Therefore, the estimation of the center (X_lim, Y_lim) and radius (R_lim) of circular limbic boundaries is based on the iris edge points confined inside this rectangle strip region. We determine this delimited strip area using the parameters Y_p and R_p estimated from the pupil detection procedure. In other words, we drop two horizontal lines through pixel points, (X_p, Y_p + R_p) and (X_p, Y_p - R_p), to limit the rectangle in the binary image obtained previously. Only for better illustration purposes, Fig.s 5(a) and 5(b) show the positions of these two delimitation lines drawn in the original eye image and the delimited areas (1 and 2), respectively, to be used for limbic boundary detection.

2.2.1 Delimitation of the region of interest

The region of interest here declared is the smallest strip area cropped from the improved image, which covers the whole pupil. The estimation of the center (X_lim, Y_lim) and radius (R_lim) of circular limbic boundaries is based on the iris edge points confined inside this rectangle strip region. We determine this delimited strip area using the parameters Y_p and R_p estimated from the pupil detection procedure. In other words, we drop two horizontal lines through pixel points, (X_p, Y_p + R_p) and (X_p, Y_p - R_p), to limit the rectangle in the binary image obtained previously. Only for better illustration purposes, Fig.s 5(a) and 5(b) show the positions of these two delimitation lines drawn in the original eye image and the delimited areas (1 and 2), respectively, to be used for limbic boundary detection.
2.2.2 Identification of the best median filter

This step can be viewed as a further but necessary image pre-processing step in iris detection before we are in fact performing the limbic boundary detection procedure, in order to minimize the interference of iris texture. Here, we show how to choose a suitable median filter used for this end. This median filter is determined through an interactive process involving iris edge detection and verification, recursively. In other words, we vary the dimension (size $n \times n$) of a mask filter until the detection of iris edges is declared. Operationally, the filter dimension parameter $n$ varies from 10 to 55 (pixels) in a 5-pixel step until edge points are detected. (See the next subsection for details). Note that this filtering operation is always performed over the original raw eye image.

Figures 6(a) and 6(b) show the original example image and the delimited/filtered image by a $15 \times 15$ median filter, respectively. Note that the iris texture details are completely erased while limbic boundaries are enhanced (see Fig. 6(b)).

2.2.3 Limbic boundary localization

The purpose of this processing step is to determine limbic boundaries based only on the information given by delimited regions and assumption of circular limbic boundary modeling. Unfortunately, the filtered image (e.g., Fig. 6(b)) is still not suitable for detection. Further image processing is necessary which consists of binarization of the filtered image. The threshold value to be used for binary image generation is exactly the average gradient value estimated from the median filtered image after it has been submitted to the same two Sobel operators used for pupil localization. Figure 7(b) shows the obtained binary image from the original eye image in Fig. 7(a).

Having a suitable binary image in hand, in order to execute limbic boundary localization fast and efficiently, once again, we restrict our search inside of some limited areas of interest as depicted by Fig. 7(b). In addition, in order to achieve high detection accuracy, once again, we make use of pupil localization information (the center and radius). That is, the limbic boundary search is restricted to the box areas, the regions 1 and 2 only. An advantage of this approach is the elimination or avoidance of interferences of eyelids and eyelashes in limbic localization for most real situations.

2.2.4 Optimization of limbic boundary detection

This step aims at achieving reliable limbic boundary detection, i.e., to reliably determine the circle-model-based parameters, $(X_i, Y_i)$ and $R_i$. This detection procedure can be viewed as one of optimization, which consists of maximizing the total amount of overlaps between valid (survived) pair pixels in the restricted regions (1 and 2) and an artificially generated boundary circle. This circle is generated based on the circle center and diameter information estimated from using only the surviving pairs of pixels, as described in the last subsection, B3. The criterion of optimization is, therefore, in function of the size $n$ of the median filter, which is responsible
for generated binary images, from which candidate boundary points are enhanced. However, in order to meet real-time response requirement, the optimization procedure was elaborated in the following manner. The detection of limbic boundary is declared when there are 30% or more overlapping pixels, for \( n \) varying from 10 to 55 in step of 5 pixels. In practice, a few numbers of \( n \) is needed to be tested. The experimental investigations indicated that, in general, no more than three different values of \( n \) needed to be tried for boundary detection. Notice that the 30% decision threshold value for the overlapped pixel amount was empirically determined and proved efficient.

Figure 8 shows 6 filtered images derived from two individuals’ original eye images; each eye image was processed by three median filters with different sizes (Subject 1: (a) 15 \( \times \) 15, (b) 25 \( \times \) 25, (c) 35 \( \times \) 35; Subject 2: (d) 15 \( \times \) 15, (e) 25 \( \times \) 25, (f) 35 \( \times \) 35). For Subject 1, both 15 \( \times \) 15 and 25 \( \times \) 25 median filters meet 30% decision threshold while, for subject 2, all three filters satisfy this 30% requirement.

The proposed detection method can be divided into two steps: (1) localization of areas of interest; (2) determination of edge pixels (of lower eyelid, upper eyelid, and possibly iris edges).

The step of localization of areas of interest becomes easy and straightforward once the pupil localization and limbic boundaries are determined in advance. The area of interest for upper lid is a rectangular window encompassing the most relevant region in which the eyelid should be localized. The position of this rectangular window is specified by the pupil-and-iris circular model parameters. That is, for upper eyelid detection, the coordinates of the four corners of the rectangular window are: (1) \((x_u, y_u)\): the upper-left corner; (2) \((x_u, y_l)\): the upper-right corner; (3) \((x_l, y_u)\): the lower-left corner; (4) \((x_l, y_l)\): the lower-right corner, where \( x_u = X_i - R_u \), \( y_u = Y_p + R_p \), \( x_l = X_i + R_l \) and \( y_l = \) bottom line number of the image.

A similar procedure is applied for lower eyelid detection by specifying the position of a window below the pupil. That is, the coordinates of the four corners of this bottom rectangular window are: (1) \((x_l, y_l)\): the upper-left corner; (2) \((x_u, y_l)\): the upper-right corner; (3) \((x_l, y_u)\): the lower-left corner; (4) \((x_u, y_u)\): the lower-right corner, where \( x_l = X_i - R_l \), \( y_l = Y_p - R_p \), \( x_u = X_i + R_u \) and \( y_u = 0 \). For illustration purposes, Figs. 9(a) and 9(b) show the positions of the delimited areas for lower and upper eyelid detection on the original and filtered images, respectively.

2.3 Eyelid and eyelash detection

This sub-section presents the proposed procedure for upper and lower eyelid detection. Eyelid detection is an even tougher problem one may face in iris segmentation. Many factors contribute to high detection complexity. In general, the shape of eyelids is highly irregular, making assumption of any particular shape or mathematical model unrealistic. Also, the presence of eyelashes makes eyelid detection difficult, especially for the upper eyelids. This is because usually more eyelashes are present in upper eyelid regions.

In general, model-based approaches require considerably more processing time in order to attend the requirement of real time applications. We avoid any specific shape modeling approach, but preserve high detection efficiency and accuracy, and probably mostly important, the detection of both sources of occlusion (eyelids and eyelashes) should be carried out simultaneously. Also, once again, we take advantage of knowing the exact localization of the pupil as well as the limbic boundary (due to circular modeling) to reduce significantly the area of eyelid search.

The obtained image after filtered images by median filters with different sizes. Subject 1: (a) 15 \( \times \) 15, (b) 25 \( \times \) 25, (c) 35 \( \times \) 35. Subject 2: (d) 15 \( \times \) 15, (e) 25 \( \times \) 25, (f) 35 \( \times \) 35.

Figure 9. Eyelid/eyelash detection. (a) The delimited area for lower and upper eyelid detection. (b) The obtained image after threshold decision.

Notice that two distinct methods were elaborated for upper eyelid and lower eyelid detection in order to achieve an overall good performance. The interference introduced by the upper eyelashes is, in general, considerably more severe than that by the lower ones. For upper eyelid detection, linear approximation (straight-line modeling) is adopted, while for lower eyelid detection, a parabolic curve modeling is considered. Two detection procedures are performed on the same binary image for limbic boundary localization. Figure 10(a) shows an example of the straight line and parabolic curve fitting for upper and low eyelid detection, respectively.

Each of these two modeling methods (straight line and parabolic curve) can be done in either one of two manners. The first one will involve simultaneously all occluded pixels (due to both eyelids and eyelashes) inside the image areas delimited by the corresponding rectangle windows. Figure 10(b) shows, in white color, all eyelid candidate pixels inside the delimited...
regions of an eye image example. Notice that, the upper eyelids are severely obscured by the interference of eyelashes.

On the other hand, the second modeling approach makes use of only one detected eyelid edge pixel (i.e., the most representative one) of each column inside the rectangular window. The search for eyelid candidate pixels is performed sequentially column by column. The technical procedure for the search is similar to that used for the pupil edge and limbic boundary detection. Before the proper eyelid detection, a median filter is used to eliminate iris texture details, and then the same Sobel filter is applied to enhance every pixel occluded by eyelids, eyelashes, and possibly iris outskirt edges. Notice that iris outskirt edges can be easily marked when the iris area is free from eyelid and eyelash occlusions (see Fig. 10(b)). In other words, for each column, if the number of detected candidate pixels is more than one, the one closest to the pupil will be selected as the eyelid edge pixel. Figure 10(c) illustrates the results of the second possible eyelid detection approach applied to an eye sample image. Notice that in the absence of a detected candidate pixel in a column, the candidate pixel in that column is artificially generated by interpolation using its neighbors’ already detected candidate pixels.

![Figure 10. (a) Linear fitting (upper eyelid) and parabolic fitting (lower eyelid). (b) All candidate pixels for eyelid detections. (c) Detected eyelids: upper eyelid (white pixels) and lower eyelid (black pixels).](image)

It is worth mentioning that both modeling fitting approaches described above offer good and close approximation to real eyelid edges, even with the presence of high-level noise. This is due to the fact that both approaches can detect most real eyelid pixels, which was experimentally verified. However, the second approach (one candidate pixel representation in each column) is more attractive in the sense of less processing time. Also, it is interesting to observe that even though eyelashes can turn an eyelid detection task more difficult, in fact they affect little overall iris recognition performance.

### 2.4 Low-quality iris image segmentation

As mentioned in Subsection 2.1.4, the proposed pupillary boundary detection method makes use of candidate pixel points detected in two opposite quadrants (quadrants I-III and/or II-IV). In general, plenty of candidate pixels can be found from good-quality iris images, e.g., those in CASIA-IrisV3-Interval [23], and consequently, a high success rate of iris segmentation can be achieved. On the contrary, low-quality iris images (e.g., those in CASIA-IrisV3-Lamp and CASIA-IrisV3-Twin), due to excessive occlusions (eyelashes, eyelids, hairs, etc.), make the segmentation tasks extremely difficult. It is common that considerable half-upper iris regions are largely occluded by upper eyelids and eyelashes. Therefore, few candidate pixel points are available from the quadrants I and II for limbic boundary detection, which definitely compromises good segmentation results. Fortunately, minor modifications were needed to adjust our segmentation algorithm described in Subsection 2.1.4 for solving this problem, and the adjusted algorithm was consequently successful in dealing with the low-quality iris images. These two minor adjustments are described as follows: (1) estimation of limbic boundaries based on limbic boundary candidate pixels selected from quadrants III and IV; (2) divide the delimited region for upper eyelid detection into 3 sub-regions and eyelid detection is carried out individually in each sub-region.

For illustration purposes, Fig. 11(a) shows some unsuccessfully segmented iris images by the approach proposed in [24]. But, in contrast, both their pupillary and limbic boundaries were correctly localized by our adjusted method, illustrated by Fig. 11(b).

![Figure 11. (a) Examples of pupil detection and limbic boundary localization by the iris segmentation algorithm described in [24]. (b) Examples of pupil detection and limbic boundary localization by the proposed iris segmentation algorithm.](image)

It is worth mentioning that, recently, an interesting and complex iris segmentation method has been recommended for extremely low-quality iris images [21]. This method, during its design phase, required intense training of its segmentation
module through a huge amount of iris images, and according to the authors, very low iris recognition error rates and online responses were achieved. In spite of the remarkably good results, in terms of iris recognition rate on successfully segmented iris images, reported in the paper, it is still unclear about real iris segmentation performances of their iris segmentation algorithm, especially in terms of iris segmentation error rates, with varying sizes of training databases (small, medium and large sizes).

2.5 Normalization of iris areas

Eye images acquired from using different optical sensors may vary largely in terms of size and quality. In particular, iris regions can be easily altered under different environmental conditions, e.g., variations in illumination, which make pupils more or less dilated, and consequently with pupils and irises more or less concentric. Such elastic changes in shape as well as in texture of iris areas will affect largely matching results, and consequently recognition performance.

The iris area normalization algorithm proposed here aims at minimizing the effects caused by iris texture deformation mainly due to pupil dilation. The basic idea is to divide the entire iris area into 21 distinct and concentric rings. The common center of all these circular rings is exactly the estimated pupil circle center. The inner radius of the smallest ring will be exactly the radius of the pupillary boundary circle. However, the first ring is discarded for iris information extraction and recognition, due to the imperfection of the real pupillary boundaries. The external radius of the largest ring should have the Euclidean distance between the farthest point of the limbic boundary and the pupil circle center. All 21 rings have the same width and each ring is further divided into 180 arc segments of equal size. Therefore, there are in total 3600 distinct arc segments to be used for iris recognition.

Since the pupillary and limbic boundaries may not be concentric, it may happen that some arc segments do not contain iris texture pixels. In this case, the information extracted from those arc segments is considered as noise and should be discarded. Fortunately, in general, they are in small number; therefore, the quality of the analysis is hardly affected. For each arc segment, we also verify whether it belongs to the eyelids or not. If it indeed does, this arc segment is discarded also. Notice that this additional verification step is fundamentally important to avoid significant degradation of the recognition performance due to eyelid interferences. Figure 12 shows two examples of ring formation and arc segments on two iris sample images with detected eyelid/eyelash frontiers. A more detailed description of the normalization algorithm used can be found in [24].

2.6 Gabor filters and convolution

After having obtained normalized arc segments for iris texture areas, we execute iris feature extraction for later comparison/verification tasks. Basically, our iris feature extraction procedure is composed of two operations: fast Fourier transforms on the normalized images and convolution between the transformed image and a Gabor filter. The outcome of these two operations is a binary vector, often known as the Iris Code. For a detailed description of these two operations, please refer to [3] and [21].

2.7 Storage and Comparison

Once there is a set of iris codes available for each input iris image, the recognition/verification task consists of comparing this new iris code set with all reference sets stored in a previously built database. In other words, the Hamming distance between two binary iris code vectors is calculated, which provides an index of similarity between two iris images. For details, see [24].

3. Results and discussion

A complete iris recognition system with the methods described in the last section was implemented in Java in order to evaluate the performance of the proposed iris segmentation algorithm. We simultaneously obtained the performances of the proposed iris image segmentation algorithm and overall iris recognition system by testing some sets of eye images from the database mentioned in [23]. This database contains three major group sets with some distinct properties. A complete description of each group set can be found in the referred site [23]. In Table 1, we list some relevant information about these database sets.

The images in CASIA-IrisV3-Interval presented better quality than those in CASIA-IrisV3-Lamp, possibly also due to better illumination condition offered during the acquisition procedure. Initially, we evaluated the proposed iris segmentation method by testing all images of these two database sets. This phase of test occurred automatically without any external intervention. However, for verification purposes, one-by-one, we also manually checked the accuracy of two segmentation steps involved, pupil detection and limbic boundary localization.

Table 2 summarizes the results of the tests on CASIA-IrisV3-Interval, including the performances of the two involved partial segmentation steps, in terms of segmentation error rate. For comparison purposes, we also tested the segmentation method proposed in [24] by CASIA-IrisV3-Interval, and obtained a 17% segmentation error rate. Notice that this segmentation error rate (17%) is considerably higher than those listed in Table 2. Even the segmentation algorithm
Fast and Efficient Iris Image Segmentation

was tested by the images in CASIA-IrisV3-Interval, which are composed of iris images with good quality.

Table 2. Segmentation performances of the proposed segmentation algorithm.

<table>
<thead>
<tr>
<th>Segmentation step</th>
<th>Segmentation error (%)</th>
<th>Segmentation error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CASIA-IrisV3-Interval</td>
<td>CASIA-IrisV3-Lamp</td>
</tr>
<tr>
<td>Pupil detection</td>
<td>0.05</td>
<td>-</td>
</tr>
<tr>
<td>Limbic boundary</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>localization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall iris</td>
<td>3</td>
<td>6.2</td>
</tr>
<tr>
<td>segmentation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Our tests on a more challenging iris image database set, CASIA-IrisV3-Lamp, confirmed that our segmentation methods are indeed robust in dealing with low-quality images, and most importantly, offer comparable and attractive performance (6.2% segmentation error rate, see Table 2). This performance measure is indeed encouraging due to the low quality of the tested iris images.

In terms of processing speed, Table 3 lists the average processing time needed for each involved processing step in an iris recognition process. More precisely, the processing times were computed for the iris images of CASIA-IrisV3-Interval and CASIA-IrisV3-Lamp. Clearly, more processing time was needed for segmenting low-quality iris images, due to more time being consumed in searching for validating candidate pixels for boundaries localization.

Table 3. Processing speed.

<table>
<thead>
<tr>
<th>Processing step</th>
<th>Average time (seconds)</th>
<th>Average time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CASIA-IrisV3-Interval</td>
<td>CASIA-IrisV3-Lamp</td>
</tr>
<tr>
<td>Pupil segmentation</td>
<td>0.466</td>
<td>2.088</td>
</tr>
<tr>
<td>Iris segmentation</td>
<td>0.202</td>
<td>0.599</td>
</tr>
<tr>
<td>Eyelid detection</td>
<td>0.167</td>
<td>0.647</td>
</tr>
<tr>
<td>Normalization of iris</td>
<td>0.001</td>
<td>0.024</td>
</tr>
<tr>
<td>area</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gabor filter &amp;</td>
<td>0.046</td>
<td>0.059</td>
</tr>
<tr>
<td>convolution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total processing time</td>
<td>0.882</td>
<td>3.397</td>
</tr>
</tbody>
</table>

Comparatively, we also evaluated the processing time of the recognition method proposed in [24] by using CASIA-IrisV3-Interval. In this case, the average time required for iris recognition was about 15 seconds, which is considerably longer than that based on our approach (0.883 seconds, see Table 3). Our explication for this huge discrepancy in processing time is the following: our approach makes the search simple and fast, because only a small quantity of pixels needed to be tested. In addition, in general, the number of tests is limited to 3 for our method, while in [24], each pixel needs to be tested exhaustively for all possible values of \( (X_p, Y_p, R_p) \) through several highly complex mathematical operations.

The tests performed on low-quality iris images (CASIA-IrisV3-Lamp) prove that significantly more additional processing time is needed. It took 3.397 seconds, on the average, for our approach, compared with about 40 seconds for the method proposed in [24]. Evidently, the latter case is far from acceptable for real-time applications. Interesting enough, Table 3 also demonstrates that the iris segmentation procedure consumes most of the entire processing time needed in an iris recognition process.

Comparatively, Table 4 summarizes the performance of the proposed iris segmentation algorithm together with that of the algorithm described in [24], in terms of processing speed and segmentation accuracy. Without doubt, our method largely outperformed the approach suggested in [24]. Unfortunately there is no information available (in terms of segmentation error rate) in [21] for comparison. Mostly important and remarkably interesting, one important advantage of our approach is that no training samples are needed.

Table 4. Performance comparison: the proposed methods versus that in [24].

<table>
<thead>
<tr>
<th>Method</th>
<th>Speed (seconds)</th>
<th>Speed (seconds)</th>
<th>Segmentation error rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed method</td>
<td>0.882</td>
<td>3.397</td>
<td>3</td>
</tr>
<tr>
<td>Method proposed in [24]</td>
<td>15</td>
<td>40</td>
<td>17</td>
</tr>
</tbody>
</table>

For overall performance of our iris recognition system, Fig. 13 shows two receiver operating characteristic (ROC) curves, which were obtained by considering all input iris images and excluding those not correctly segmented, respectively. Their equal error rates (EERs) were 3.5% and 0.57%, respectively. In this case, we used CASIA-IrisV3-Interval for test. The discrepancies between two ROCs confirm the importance of a good and accurate iris segmentation module in an iris recognition system.
Additionally, we also evaluated the proposed iris recognition system by low-quality iris images (CASIA-IrisV3-Lamp). In a similar manner, Fig. 14 shows the results through two ROC performance curves, with or without those images not correctly segmented, respectively, involved in the final pattern matching process. For easy illustrations, two corresponding EERs, 6.4% and 1.48%, are also explicitly printed. Higher error rates obtained for the CASIA-IrisV3-Lamp database set are by no means surprising since iris images with worse quality evidently make both segmentation and recognition tougher.

4. Conclusion

Iris segmentation is probably one of the most crucial operations involved in iris recognition. Accurate iris segmentation is fundamental for the success and precision of the subsequent feature extraction and recognition, and consequently allowing the iris recognition system to achieve desired high performances. Most iris recognition approaches proposed in the literature require exhaustive search and learning of many modeling parameters and characteristics, which prevents their effective application in real time, and turns the system highly sensitive to noise. Very few iris segmentation algorithms reported in the literature have indeed been tested by or are capable of accurately segmenting iris images with very low quality, mainly due to severe occlusions, and also providing online responses. This report has presented a fast and efficient iris segmentation methodology to address a solution to these problems. Three major procedures involved in the proposed iris segmentation approach, namely pupillary detection, limbus boundary localization, and eyelids and eyelash detection, were carefully designed in order to avoid unnecessary and redundant image processing, and most importantly, to preserve the integrity of iris texture information.

Besides its real-time applications, the proposed iris segmentation method is also intended to maximize correct iris segmentation rate whether occlusions on iris texture are severe or not. Such a quality makes the proposed approach less dependent, if not completely independent of employed optical cameras for iris image acquisition. Another interesting feature of the proposed algorithm is that almost no training phase is needed. The proposed segmentation approach is simple but extremely robust in terms of being less sensitive to large variations of input iris images which can be applied to other similar eye images [27]. Experimental results demonstrate that the proposed algorithm outperforms some well-known methods in both accuracy and processing speed.

References

Fast and Efficient Iris Image Segmentation


[23] Institute of Automation: CASIA Iris Image Database (v3.0), available online: http://www.sinobiometris.com


