Biometric Security via Infrared Iris

M.S. Al-Rawi
Computer Science Dept, King Abdullah the Second School for Information Technology.
Jordan University
rawi@ju.edu.jo:
phone: 06-5355000-4518
POB 13496, Amman 11942, Jordan

Samer Mashaqba
Progress soft cooperation
Jordan

samerjo1982@yahoo.com
Amman Jordan

Abstract:
In this paper, new and simple techniques are presented to recognize iris images scanned by an infrared iris scanner. An infrared iris may suffer from many disadvantages compared to iris scanned via ordinary (visible) scanners. These disadvantages may include (but not limited to); size variant pupil, low contrast, heavily occluded eyelashes, semi closed eyelids, more background illumination, and more shape geometrical changes. Each eye image is preprocessed in several stages, smoothing and enhancement, edge detection, interpolation, pupil and iris localization algorithms, eyelids detection, eyelashes isolation, and removing background illumination. This preprocessing aims to extract the iris image. Gabor features are then calculated for the extracted iris image after decomposing it into several blocks. Afterwards, the weighted Euclidean distance is calculated between an iris test image and the Gabor features stored in the database obtained from iris images used in training. The minimum weight Euclidean vector associates the test iris to one of that of the database. Experimental results on the well known standard CASIA database shows that the system has 94% of recognition accuracy.

Keywords: *iris recognition, biometric, security, image processing, Gabor filter, CASIA, infrared, texture analysis*

1. Introduction

Iris identification methods are quite new in the computational world. The idea that people can be identified by the shape of their irises was first documented in an ophthalmology textbook by James Doggarts in 1949. After that the idea lie dormant for decades until 1987 two ophthalmologists, Aran Safir and Leonard Flom, patented this idea. In 1989 they asked John Daugman to create an actual algorithm for the iris recognition problem. These algorithms that he had developed and patented in 1994 form the entire basis for the current iris recognition research and products [3] [4].

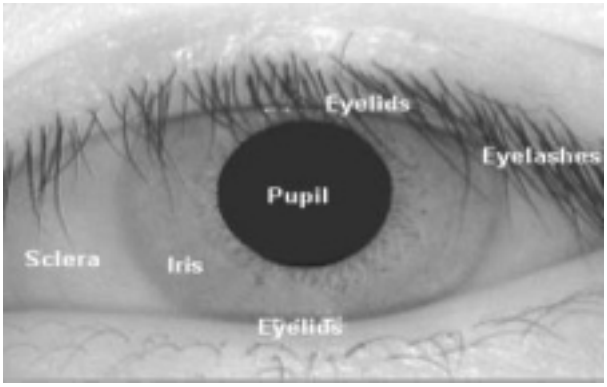

Fig. 1 A front-on view of the human eye

The typical iris recognition system may be giving as shown in Fig.2.

The benefits of using Iris technology include: iris patterns are extremely complex, the iris-recognition systems test for a live eye by checking for the normal continuous fluctuation in pupil size, patterns are individual (even in fraternal or identical twins), patterns are formed by six months after birth, stable after a year, and they remain the same for life, imitation is almost impossible, and patterns are easy to capture and encode.

Iris recognition revealed itself as the most reliable biometrics technology, so it has a large scale of applications include (but not limited to): Financial Services, Health Care, Law Enforcement, Government Applications, Travel and Immigration, E-Commerce and Telephony, Physical Access and Time and Attendance, PC/Network Access, Retail - ATM - Point of Sale, Criminal Identification, Citizen Identification, and Surveillance.

First, an image containing the user's eye is captured by the system, we used an iris image database called CASIA contains 108 classes and each class has 7 iris images, which means that we have 756 iris images. Then, the image is preprocessed to localize the pupil and the iris in the image, so that we can normalize the iris region. Thirdly, features representing the iris

patterns are extracted. Finally decision is made by means of matching.

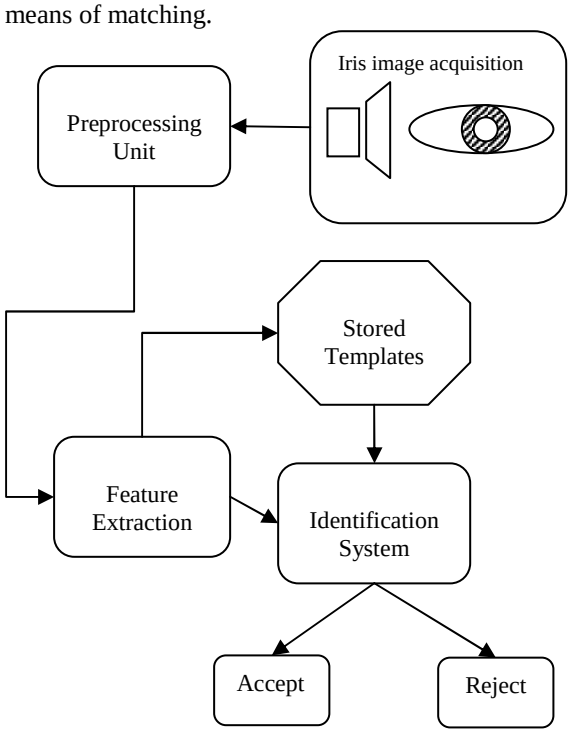

Fig. 2 : A typical iris recognition system

Although MATLAB® is very easy to implement the system simulation; we chose to use JAVA programming language since it provides more object oriented perspectives. In section 2, we discuss the preprocessing stage, section 3 includes the topic of iris extraction to be prepared for the next stage, section 4 shows how iris features are extracted using Gabor filter bank, in section 5 we demonstrate the experimental results conducted using the CASIA iris image database.

2. The preprocessing unit

This unit is one of the important units used in this paper, the aim is to prepare the eye image to the next advance stages. Common preprocessing operations are demonstrated briefly and more details can be found in the literature, while those specialized to iris conversion are discussed with more details.

Image Smoothing and Enhancement

This is consisted of the following processes on each iris image the mean filter, the Gaussian smoothing, and histogram equalization. These operations are commonly used in image processing.

Canny's Edges Detector

the canny edge detector first smoothes the image to eliminate noise and unnecessary details. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (non-maximum suppression). The gradient is now further reduced by hysteresis thresholding. Hysteresis thresholding is used to track along the remaining pixels that have not been suppressed. Hysteresis thresholding uses two thresholds to prevent streaking problem that is very common in edge detection in general. If the magnitude is below the first threshold, it is set to zero (made a non-edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the two thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above the high threshold, the next sections will explain each of these steps [8].

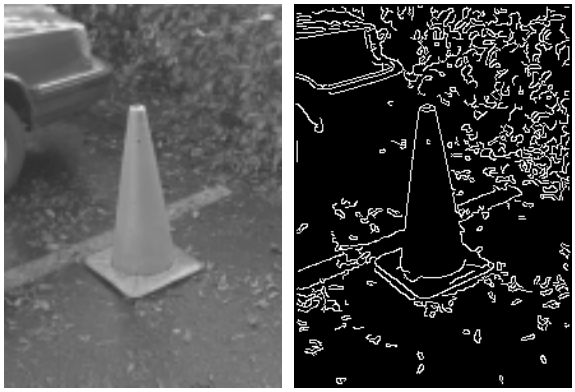

Fig. 3 The image on the left is the original one, and its edge map image, on the right, generated by Canny edge detector

Hough Transform

The Hough transform is a technique which can be used to isolate features of a particular shape within an image. Because it requires that the desired features be specified in some parametric form, the classical Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, etc. The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise. To detect the iris of the image the so called Circular Hough Transform may be used using the following parametric equation:

$$(x - x_0)^2 + (y - y)^2 = r^2 \qquad \textbf{(1)}$$

Where (x_0, y_0) are the coordinates of the center of the circle and r is the radius

The method starts with a search for a pixel that represent an edge point; after such a pixel is found, a locus of potential center points of the circle associated with it can be determined. Such a locus of potential center points forms a circle with the radius r, then (x_0, y_0, r) in the Hough space accumulator is incremented. If each pixel of the image occurs as an element of the circle-center loci the loci of potential

circle centers are constructed for edge map points identified in the original image, the frequency in Hough space accumulator can be determined with. The true center of the circle being sought is represented by the pixel with the highest frequency of occurrence in the circle-center loci, for more information see [5] [6] [10].

Geometric Transformation and Interpolation

Geometric transformations are used for the process of mapping of an annular region to rectangular one. These transformations modify the spatial relationship between pixels in an image. Geometric transformations often called rubber-sheet transformations, because it may be viewed as the process of printing an image on a sheet of rubber and then stretching this sheet according to some predefined set of rules. They are consisted mainly of two parts (1) a spatial transformation, which defines the rearrangement of pixels on the image plan; and (2) gray-level interpolation. Since this transformation is sensitive to the rectangular iris construction, we will discuss it with some details

Suppose that an image with pixel coordinate (x, y) contains an annular region. The annular region has two circles, one of them is an inner circle and the other is an outer circle. The process of mapping the annular region into rectangular one can be accomplished by projecting the original annular region in a Cartesian coordinate system into a doubly dimensionless pseudo polar coordinate system.

The annular region in the new coordinate system (x', y') can be represented in the fixed parameter interval [0, 1]. Similar to this scheme, the annular region can be counterclockwise unwrapped into a rectangular block with a fixed size as follows [11], also see [12]:

$$
\begin{aligned}
&I_{rec}(x', y') = I_{org}(x, y) \\
&x = x_i(\theta) + [x_o(\theta) - x_i(\theta)](y'/M) \\
&y = y_i(\theta) + [y_o(\theta) - y_i(\theta)](y'/M) \\
&\theta = 2\pi(x'/N)
\end{aligned}
\qquad (2)
$$

Where I_{rec} is an $M \times N$ rectangular block, $\{x_i(\theta), y_i(\theta)\}$ and $\{x_o(\theta), y_o(\theta)\}$ are the coordinates of the inner and outer circle's boundary points in the direction θ in the original image I_{org}. Note that y'/M and x'/N in the above equations lies in the interval [0, 1], see Fig. 4.

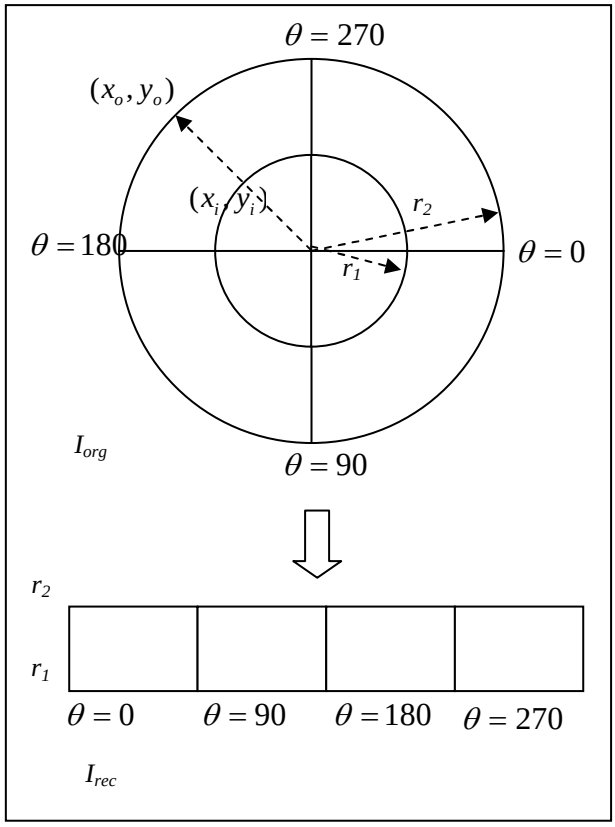

Fig. 4 Mapping of annular region into rectangular one

The method discussed above steps through integer values of the coordinates (x, y) in the original image I_{org} to yield the non-integer output image I_{rec}. Thus using non-integer values for x and y causes a mapping into locations of I_{org} for which no gray levels are defined. Inferring that the gray-level values at those locations is based only on the pixel values at integer coordinate locations. The simplest scheme for gray-level interpolation is based on the nearest neighbor approach, which rounds the values of x and y to the nearest integer. Although the nearest neighbor interpolation is simple to implement, this method often has the drawback of producing undesired artifacts. Smoother result can be obtained using more sophisticated techniques, such as the bilinear interpolation, the bicubic interpolation, and the intensity averaging interpolation presented in this paper. For more information about interpolation see [5] [7]. As shown in Fig. 5, the intensity averaging interpolation can be expressed as:

$$I_{rec}(x, y) = \sum_{i \in \text{all contributed pixels}} A_i I_i \qquad (3),$$

where A_i represents the occupied area from the pixel whose intensity value is I_i.

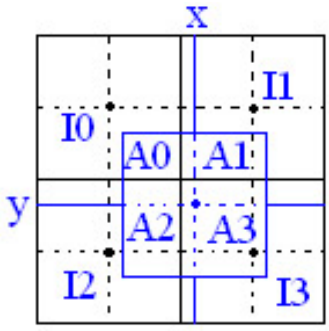

Fig. 5 Intensity averaging interpolation

3. Iris Extraction

The acquired image always contains not only the useful parts (iris) but also some irrelevant parts (e.g. eyelid, pupil …), see Fig. 1. Different eye-to-camera distance may result in different image sized of the same eye. For the purpose of analysis, the original image needs to be preprocessed. The preprocessing is composed of six steps: Pupil and Iris Localization, Eyelids detection, Eyelashes isolation, Extracting region of interest, Estimating and removing the background illumination, Iris texture correction.

Pupil and Iris Localization

Both the inner and the outer boundaries of a typical iris can approximately be taken as circles. However, the two circles are usually not co-centric [13], but they are assumed co-centric at the center of the pupil in this research. We decid to use circular Hough transform, discussed previously, for detecting the iris and pupil circular boundaries [14] [15] [16]. This involves first employing Canny edge detection, discussed previously, to generate an edge map image, only the strongest edges will be detected by controlling the parameters of Canny's edges detector, e.g. size of Gaussain-filter, the upper and the lower thresholds, and sigma of Gaussain-filter. Gradients were biased in the vertical direction for the outer iris/sclera circular boundary, since it appeared as two vertical lines in the eye's image, see Fig. 6 and Fig. 7. Note that the iris/sclera boundary appears as doted edge, thus due to the low contrast between the two regions in the CAISA iris image databse. Vertical and horizontal gradients were used for the inner iris/pupil boundary, see Fig. 8.

The range of radius values to search for have been set manually, depending on the database used. For the CASIA database, values of the iris radius range from 90 to 120 pixels, while the pupil radius ranges from 25 to 75 pixels. In order to make the circle detection process more efficient and accurate, the circular Hough transform of three dimensional space for the iris/pupil boundary is performed first to detect the center and the radius of the pupil, see Fig. 9, then the circular Hough transform of one dimensional space for the iris/sclera boundary have been performed to detect the radius of the iris, since we assume the pupil and the iris are co-centric, see Fig. 10.

By detecting the pupil and the iris, the system will be invariant to translation. The localization technique managed to correctly localize the pupil and the iris region of the CASIA iris image database has a success rate of around 100% and 98% respectively. Others solutions for detecting the iris and the pupil circles are:

Daugman's Integro-differential Operator [3][23].
Active Contour Models as suggested by Ritter et al [25].

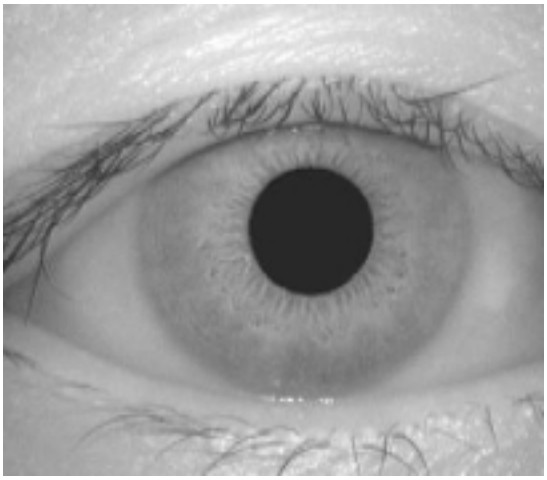

Fig. 6 The original eye image before processing

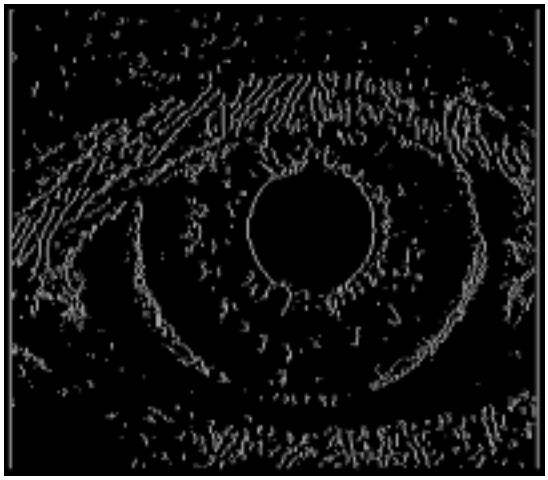

Fig. 7 The edge map image of the eye image shown in Fig. 6 generated by Canny edge detector, where the gradient is biased in vertical direction

Fig. 8 The edge map image of the eye image shown in Fig. 6 generated by Canny edge detector, where the gradient is biased in vertical and horizontal directions

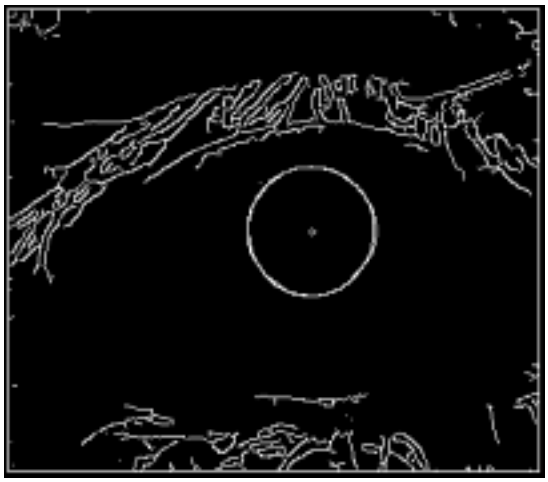

Fig. 9 The circular Hough transform for the iris/pupil boundary is performed first to detect the center and the radius of the pupil

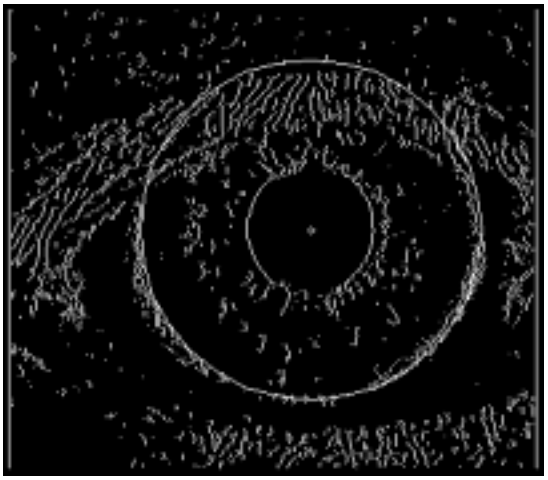

Fig. 10 The circular Hough transform for the iris/sclera boundary is preformed to detect the radius of the iris

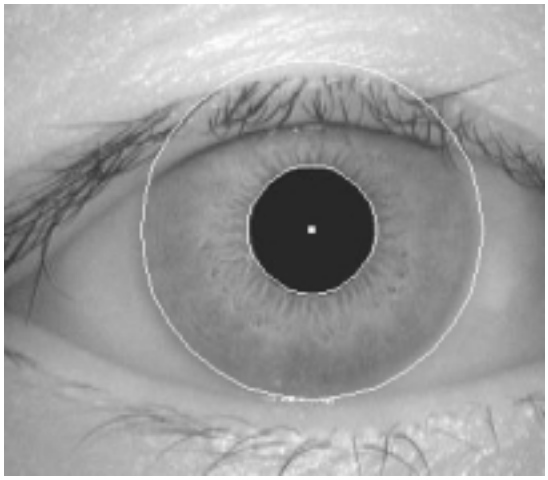

Fig. 11 The final result of the localization process

Eyelids Detection

Eyelids normally occlude the upper and lower parts of the iris region. a simple but effective technique to detect them is introduced herewith, this technique consists of the following steps:

Generate a horizontal edge map image using Canny edge detector and adjust parameters, so that only strongest non-vertical edges appear, see Fig. 12.
Using step 1, the peaks of the upper and the lower eyelids will be appeared as small horizontal line above the center of the pupil, see Fig. 12, so that it can be detected using linear Hough transformation, as in the following:

$$y = mx + c$$

$$m = \tan(\theta), \text{ where } \theta = 0 \qquad (4)$$

$$\rightarrow \mathrm{y} = \mathrm{c}$$

Where the y parameter range from $(pupil\_center.y - iris\_radius)$ to $pupil\_center.y$, and the c parameter range from $(pupil\_center.x - l)\,to\,(pupil\_center.x + l)$, in our experiment l is set to 5.

This experiment shows that if we generate an edge map in which the vertical and horizontal gradients are taken into consideration, the result will be better.

To detect the outskirts of the upper eyelid, search for the following two points in the outer iris ring of width w, see Fig. 12, in this research w is set to 10:

.

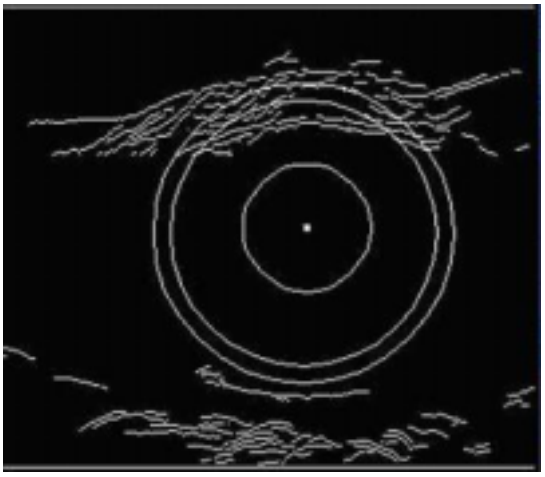

Fig. 12 Horizontal edge map image generated by Canny edge detector, note the outer ring of width 10

Max-Point: an edge point that is found in the outer iris ring, above and on the left of the pupil's center, such that it has the maximum positive distance along the x-axis.
Min-Point: an edge point that is found in the outer iris ring, above and on the right of the pupil' center, such that it has the minimum negative distance along the x-axis.
The distance along the x-axis is calculated as in the following:

$$\text{distance} = \text{pupil\_center}.x - \text{max\_point}.x$$

Construct a polygon from the detected line, max-point and min-point, then Fill it with pure white color to be marked as non-iris region, see Fig. 13.
To detect the outskirts of the lower eyelid, the same concept explained in step three have been employed.

Others solutions for detecting the eyelids are:

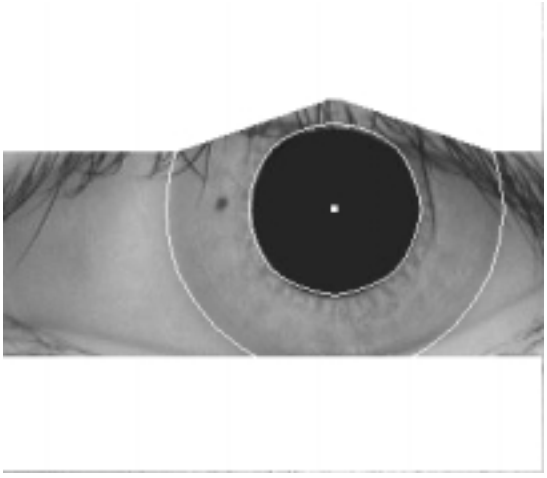

Fig. 14 An iris image after eyelids detection and before eyelashes isolation

.

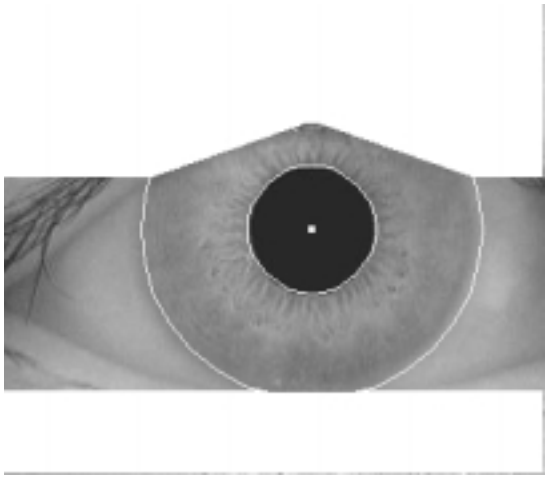

Fig. 13 The final result of eyelids detection, white polygon is placed instead of the eyelids

Daugman's Integro-differential Operator [3][23].
Using parabolic Hough Transform by R. Wildes [14].
Eye Contour Extraction, by Vladimir Vezhnevets and Anna Degtiareva [26].

Eyelashes Isolation

The eyelashes may be occlude the upper part and the lower part of the iris region. For isolating eyelashes in the CASIA database a simple thresholding technique is used to replace eyelashes region by pure white region, since analysis reveals that eyelashes are quite dark when compared with the rest of the eye image [1]. The eyelash detection system implemented for the CASIA database also proved to be successful in isolating most of the eyelashes occurring within the iris region as shown in Fig. 14 and Fig. 15. Kong and Zhang [27] present a method for eyelash detection based on a very complex technique.

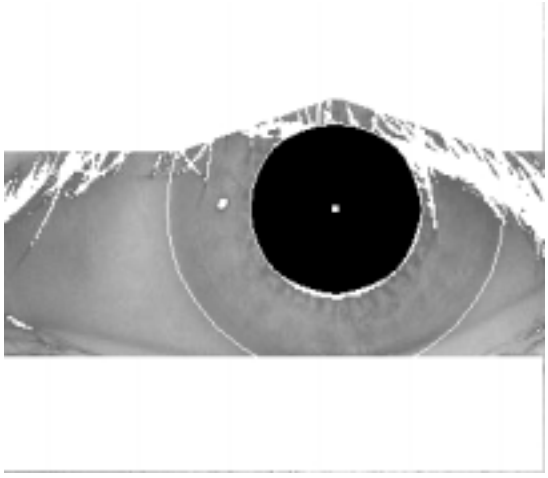

Fig. 15 the iris image shown in Fig. 14 after eyelashes isolation

Extracting Regions of Interest

Irises from different people may be captured in different sizes, and even for the iris from the same person, the size may change because of the variation of the illumination and other factors. We normalize the iris ring to a rectangular block of texture of a fixed size using the geometric transformation and the intensity averaging interpolation techniques discussed previously, so that the system will be invariant to scale. The size of the rectangular block is 60×480. The upper portion of a normalized iris image, corresponding to regions closer to the pupil called region of interest, provides the most useful texture information for recognition as suggested by [11] and [13]; so we need only to extract the region of interest of size 45×480, see Fig. 16.

Fig. 16 The region of interest

Estimating and Removing Background Illumination

The normalized iris image may have non-uniform brightness caused by variant background illumination, all these may affect the subsequent processing in feature extraction and matching. To solve this problem, firstly, we estimate the background illumination by approximating intensity variations across the whole image using the mean filter of size 15 X 15, see Fig. 17. This estimate is further expanded to the same size as the normalized image, see Fig. 18. The estimated background illumination as shown in Fig. 18 is subtracted from the normalized image to compensate for a variety of lighting conditions, see the result in Fig. 19, this technique was suggested by [11] and [13].

Fig. 17 The estimated background illumination

Fig. 18 The scaled background illumination

Fig. 19 The result of removing the background illumination

Texture Correction

The resulted image after removing the estimated background illumination is dark and has low contrast. In order to obtain a more well-distributed texture image, we improve the contrast of the image by means of the histogram equalization [13], discussed previously, only the iris texture is considered in the histogram equalization, this means that eyelids and eyelashes region are skipped. Then white regions which represent either eyelids or eyelashes are replaced by black region, since the intensity value of the black color is 0. The result of improving the contrast is shown below in Fig. 20; we can see that finer iris texture become clearer.

Fig. 20 The final result of the preprocessing stage, this is the iris to be used in feature extraction

4. Features Extraction and Classification

The iris has a particularly interesting structure and provides abundant texture information. So, from the viewpoint of texture analysis, local spatial patterns in an iris mainly involve frequency and orientation information. Gabor filter is a well known technique in texture analysis. It can filter an image at different scales and orientations, and then we can extract the desired features from the filtered image [11] [13].

Gabor Filters

Gabor filters has been used extensively for the analysis of texture in grayscale images. These filters achieved optimal joint localization in space and spatial frequency and can be used to decompose images into components corresponding to different scales and orientations. One approach to texture analysis using Gabor filters is to find specific narrowband filters for each texture class that can be used to distinguish it from other texture classes. An alternate approach is to represent a texture using outputs from a bank of Gabor filters. The outputs of the filterbank can then be passed through a nonlinear operator to extract features. A Gabor filter can be viewed as a sinusoidal plane of particular frequency and orientation, modulated by a Gaussian envelope. Banks of real circularly symmetric Gabor filters have been defined in the spatial domain by, [17] [18]:

$$g_{mn}(x,y) = \frac{1}{2\pi\sigma_m^2}\exp\left\{-\frac{x^2+y^2}{2\sigma_m^2}\right\}$$
$$\text{X } \cos(2\pi u_m(x\cos\theta_n + y\sin\theta_n)) \qquad (5)$$

Where m is the index for the scale u and n is the index for the orientation Θ. The bandwidth B_Θ is taken to be 40 ^{o} in order to maximize coverage of the frequency domain while minimizing the overlap between filters. Filters are placed one octave apart so that

$$u_m = 0.5\, u_m\text{-}1 \qquad (6)$$

The highest center frequency is taken to be

$$u_1 = 0.5 / (1+ \tan(B_\theta / 2)) \qquad (7)$$

so that band pass region of this filter is within the discrete frequency domain. The parameter σ_m can be then expressed as

$$\sigma_m = (2\ln(2))^{0.5} / (2\pi u_m \tan(B_\theta / 2)) \quad (8)$$

Each Gabor filter will be truncated at three times σ_m giving Gabor filters of sizes [9X9], [17X17], and [35X35]. For grayscale images, if ***M*** scales and ***N*** orientations are used, then a set of ***MN*** Gabor filters can be generated. The filterbank is defined using three scales, for each scale filtering is performed at $\boldsymbol{\Theta}$ = 0, 45, 90, 135, so there are total of 12 Gabor filters with different scales and orientations will be generated, for more information see [13] [19] [20] [21]

Feature Vector

Each iris image has a size 45×480 is subdivided into many blocks each has a size 10×10 which yields a total lf 4×48 blocks (after leaving the lower 5 pixels). Each 10×10 block is then filtered by a bank of Gabor filters as described in the previous section using the continuous convolution operator defined by:

$$g_i(x',y') = \frac{1}{s^2}\sum_{x=0}^{W-1}\sum_{y=0}^{H-1} I(x,y)h(x'-x,y'-y), i = 0,...,12 \quad (9)$$

Where ***S*** is the size of the Gabor filter ***h*** (S = 9, 17, 35), ***W*** and ***H*** are the width (480) and the height (45) of the iris image ***I***, and $g_i(x', y')$ is the resultant filtered image. For each iris image, a bank of 12 filtered images will be generated from which iris features are extracted. Statistical features were extracted from each 10 X 10 small block of each filtered image. Only the upper 480 X 40 values of each filtered image were taken into consideration. The total number of small blocks is 192((40/10) (480/10)) for each filtered image. For each small block two features should be calculated. This generates 384 (192 x 2) feature values for each filtered image, and 4608 (384 * 12) feature values for all filtered images. The feature values used in the recognition algorithm are the power ***P*** and the standard deviation $\boldsymbol{\sigma}$ of each filtered block defined as

$$P = BF\sqrt{\sum_x \sum_y g_i^2(x,y)}, i = 1,2,...,12 \quad (10)$$

$$\sigma = BF\sqrt{\frac{1}{B^2-1}\sum_x\sum_y (g_i(x,y) - \bar{G}_i)^2}, i = 1,2,...,12 \quad (11)$$

Where ***x*** and ***y*** belongs to one of the blocks in the filtered image g_i, ***B*** is the size of the block (10), $\overline{G_i}$ is mean of the block, and ***BF*** is the block-factor defined as

$$BF = 1 - (w_p / B^2) \quad (12)$$

Where W_p is the number of white pixels in the current block being processed. Note that ***BF*** is always in the interval [0,1]. The reason of using BF is that we cannot avoid white regions (eyelids and eyelashes) when Gabor filter is convoluted with a normalized iris image. To solve such a problem we propose:

1. Replace white regions (white has intensity value of 255) by black regions (black has intensity value of 0).
2. Use BF to determine how much iris-texture that the current block contains:

If the current block is filled with iris-texture, then BF = 1.

If the current block is filled with white region, then BF = 0.

These feature values are arranged to form a 2D feature vector where features are arrange as shown below

$$V = \begin{bmatrix} [\text{(class name).}] \\ [\text{features that are extracted from } g_1\{(p_1,\sigma_1),\ldots,(p_{192},\sigma_{192})\}] \\ [\text{features that are extracted from } g_2\{(p_1,\sigma_1),\ldots,(p_{192},\sigma_{192})\}] \\ \vdots \\ [\text{features that are extracted from } g_{11}\{(p_1,\sigma_1),\ldots,(p_{192},\sigma_{192})\}] \\ [\text{features that are extracted from } g_{12}\{(p_1,\sigma_1),\ldots,(p_{192},\sigma_{192})\}] \end{bmatrix}$$

Classifier Design

For simplicity, iris matching is based on computing the weighted Euclidean distance (*WED*) between the corresponding feature vectors. *WED* is defined in the following:

$$ED(k) = \sum_{i=1}^{N} \frac{(f_i - f_i^{(k)})^2}{(\delta_i^{(k)})^2} \quad (13)$$

Where f_i is the i^{th} feature of the unknown iris, $f_i^{(k)}$ is the i^{th} feature of iris template, ***k,*** and $\delta_i^{(k)}$ is the standard deviation of the i^{th} feature in iris template ***k,*** and *N* is the total number of features at a single iris image (here it is 384).

Each feature vector contains 12 sub-feature-vectors, and each sub-feature-vector contains two feature values, which are the power ***P*** and the standard deviation $\boldsymbol{\sigma}$. The *WED* in the corresponding feature vectors (the training one and the testing one) is computed between the features values ***P*** and $\boldsymbol{\sigma}$. This generates a total of 2 *WED* values per sub-feature-vector and 24 (12 x 2) values per feature vector. Then the 24 *WED* values are sorted, so that the ***mean*** of the first 12 sorted *WED* values is calculated as defined in the following

$$mean = \frac{1}{12}\sum_{i=1}^{12} WED_i \quad (14)$$

Where WED_i is the i^{th} sorted distance. The unknown iris template is found to match iris template ***k***, when the ***mean*** is a minimum at ***k.*** After feature extraction, an iris image is represented as a feature vector of 4608 feature values. To improve computational efficiency and classification accuracy, there are two popular

methods to reduce the dimensionality of the feature vector which are principal component analysis and Fisher linear discriminant. In this paper, none of them have been implemented, since we are trying to build a prototype for an iris recognition system.

5. Experimental Results

In this section, the performance of the iris recognition system as a whole is examined. Many tests were carried out to confirm that iris recognition can perform accurately as a biometric for recognition of individuals.

Iris Image Database

The Chinese Academy of Sciences - Institute of Automation (CASIA) iris image database version 1.0 includes 756 greyscale iris images from 108 unique eyes or classes [22]. There are 7 different images of each unique eye. Images from each class are taken from two sessions with one month interval between sessions. 3 eye images were captured in the first session, and 4 eye images in the second session. The images were captured especially for iris recognition research using specialized digital optics developed by the National Laboratory of Pattern Recognition, China. The eye images are mainly from persons of Asian people.

Feature-Vectors-Files

As described in the previous section, there are 7 instances (iris images) for each class (unique eye). The extracted feature vectors from the first instance of each class were stored in one file, and the concept have been applied for the remaining 6 instances. This generates a total of 7 feature-vector-files, each one contains feature vectors from one instance as shown in Fig. 21. The reason for such arrangement is to simplify the process of selecting the training and the testing samples. For example, the first the seventh feature-vectors-files can serve as training samples while the remaining files serve as testing samples.

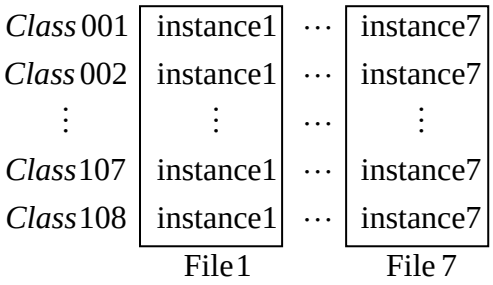

Fig. 21 The feature-vector-files

Results of Testing

Many tests were carried out to confirm that iris recognition can perform accurately as a biometric for recognition of individuals. The proposed system have been evaluated in two stages. In the first stage, the training feature vectors which belong to the same class are considered to be uncorrelated to each other at testing time. In the second stage, the training feature vectors are considered to be correlated to each other, but how?

As mentioned Classifier Design, the *WED* in the corresponding feature vectors (the training one and the testing one) is computed between the features values ***P*** and $\boldsymbol{\sigma}$. This generates a total of 2 *WED* values per sub-feature-vector and 24 (12 x 2) values per feature vector. In the second stage only two feature vectors are used in training and one for testing. The *WED* is computed between the first training sample and the testing sample which generates 24 *WED* values, and then the *WED* is computed between the second training sample and the testing sample which generates 24 *WED* values. After that, the first and the second 24 *WED* values are augmented to generate a 48 *WED* values, then the 48 *WED* values are sorted, so that the ***mean*** of the first 12 sorted *WED* values is calculated to be used in the testing. Next, the ***mean*** of the first 24 sorted *WED* values is calculated to be used in the testing.

The recognition results are summarized in Table 1. Remember that each feature-vectors-file has 108 feature vectors (templates), so numbers in the 2 to 8 column represent how many true matches of 108 are occurred.

Table 1 Recognition results using samples from file in training, others for testing

	Exp1	Exp2	Exp3	Exp4	Exp5	Exp6
File1	TR	TR	TR	82	91	93
File2	88	TR	TR	89	95	101
File3	84	100	TR	91	92	100
File4	85	97	102	99	100	104
File5	83	97	100	97	102	TR
File6	82	95	99	92	TR	TR
File7	80	99	105	TR	TR	TR
Correct /Total	502 /648	488 /540	406 /432	550 /648	480 /540	398 /432
Recognition Accuracy%	77	90	94	85	89	92

Analysis of Results

Due to the large size of the database, we cannot draw a confusion matrix, we pursue the analysis as follows; .As can be seem, Experiment1 uses all features from File1 for training, all other features for testing, the recognition accuracy is 77%. On the other hand, Experiment4 training have been

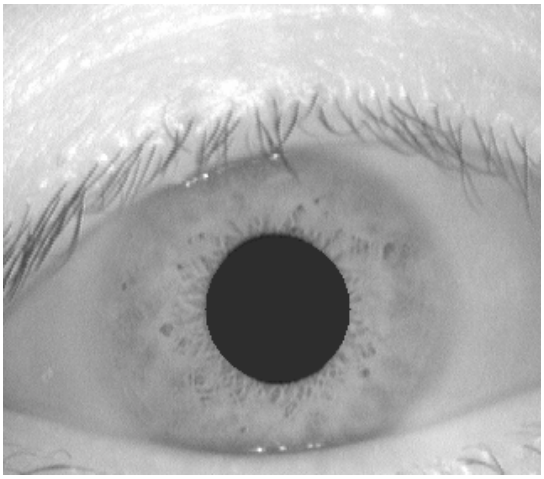

Fig. 22 : CAISA iris image database has a low contrast texture

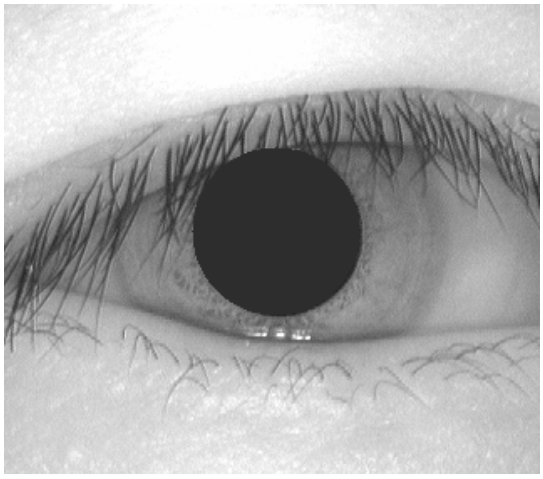

Fig. 23 Heavy occlusion of eyelids and abundant eyelashes

performed using File7 and all other features for testing yielding an 85% of recognition accuracy. In addition, File1 has the lowest correct classification with respect to Experiments 4, 5, and 6 these are (82, 91, 93). This indicates that File1 has a high confusion factor (it contains bad captured irises). This means that a system should have some criteria to determine the degree of a good iris, such as that used as the BF factor (it is never used to reject a training sample in this work), also the used BF factor in this paper is very simple. Using a bad iris texture as a reference may give incorrect classification results. To be more specific, at the time of training (storing feature templates) the system should capture several iris images then automatically choose the best one that has the less eyelashes, less eyelids occlusion, and the highest good iris area. The results demonstrated in Table 1 show that the more samples are used for training, the identification rate is increased slightly. Another issue must be mentioned, that is the classification process affected directly by the quality of the used image database (this is because the CAISA iris image database had been obtained by an infrared scanner) therefore a comparison with another test that uses a database obtained by a visible scanner is unjustified. We list the following drawbacks associated to CASIA database:

Has low contrast and bad texture, see Fig. 22.
Heavy occlusion of eyelids and abundant eyelashes, this means that the iris-texture will be lost, Fig. 23
Variant pupil sizes (according to the background illumination) see Fig. 22. and Fig. 23

Comparison with Existing Systems

The methods proposed by Daugman [3][23], Li Ma, Tieniu Tan, Yunhong Wang, and Dexin Zhang [11], Boles and Boashash [24] are probably the best-known. They characterize local details of the iris based on phase, texture analysis and zero-crossing representation respectively. Table 2 shows the correct identification rates of the proposed system compared with others works as mentioned in [11].

Table 2: Comparison with Existing Systems

Method	Recognition accuracy
Daugman [3][23]	100%
Li Ma and the others[11]	99%
Boles and the others[24]	92%
The system presented here	92%

Conclusions

This paper has presented an iris recognition system, which is tested using the CASIA iris image database. First, an automatic segmentation algorithm is presented, which would localize the iris region from an eye image and isolate eyelids, and eyelashes. Next, the segmented iris region have been normalized to eliminate dimensional inconsistencies between iris regions. This is achieved by unwrapping the annular iris into rectangular block with constant dimensions. Finally, features of the iris were extracted by convolving the normalized iris region with a bank of 2D Gabor filter in order to produce a biometric template. The weighted Euclidean distance is chosen as a matching metric.

Analysis of the developed iris recognition system has revealed a number of conclusions. It can be stated that segmentation is a critical stage of iris recognition, since areas that are wrongly identified as iris regions will corrupt biometric templates resulting in very poor recognition. The segmentation can be the most difficult stage of iris recognition because its success is dependent on the imaging quality of eye images. Experimental results show that the developed system can effectively distinguish different persons by identifying their irises.

The most computation intensive stages include performing the Hough transform, and calculating *WED* distance values between templates to search for the best match. Since the system is implemented in Java programming language, which is an interpreted

language, speed benefits could be made by implementing computationally intensive parts in C or C++ as dynamic link libraries using ***Java Native Interface***. In this paper, speed is not one of the objectives for developing this system, but this would have to be considered if using the system for real-time recognition.

JAVA programming language have been used to implement the system. The Implementation of the system is divided into 5 main packages which are; irisProject.JPGImage, irisProject.Preprocessing,, irisProject.Features-Extraction, risProject.classification, and irisProject.BatchProcessing,
These packages gives the reader a good perspective of the object oriented design of the iris project. Each one of the 5 programmed packages represents a stage in the system.

6. References

[1] Libor Masek, and Peter Kovesi. "Recognition of Human Iris Patterns for Biometric Identification", the University of Western Australia, 2003.

[2] E. Wolff. "Anatomy of the Eye and Orbit", 7th edition, H. K. Lewis & Company, 1976.

[3] J. Daugman. "History and Development of Iris Recognition", from: http://www.cl.cam.ac.uk/users/jgd1000/history.html.

[4] Jarkko Vartiainen. "Iris Recognition Systems and Methods", Lappeenranta University of Technology, Department of Information Technology.

[5] Milan Sonka, Vaclav Hlavac, and Roger Boyle. "Image Processing, Analysis, and Machine Vision", 2nd edition, Cole publishing Company, 1999.

[6] R. Fisher, S. Perkins, A. Walker and E. Wolfart. From: http://www.dai.ed.ac.uk/homes/rbf/HIPR/hiprdemo, 2000.

[7] Rafael E. Gonzalez and Richard E.Woods. "Digital Image Processing", Addison-Wesley Publishing Company, 1992.

[8] Bill Green. "Canny Edge Detection Tutorial", from: http://www.pages.drexel.edu, 2002.

[9] Sohaib Khan. "Canny's Edge Detector: Implementation", from: http://suraj.lums.edu.pk, October 2002.

[10] A web site. From: http://ct.radiology.uiowa.edu/~jiangm/courses/dip.

[11] Li Ma, Tieniu Tan, Yunhong Wang, and Dexin Zhang. "Personal Identification Based on Iris Texture Analysis", IEEE Transactions on Pattern Analysis and machine Intelligence, VOL. 25, NO. 12, December 2003.

[12] Joe F. Thompson and others. "Numerical grid generation foundation and application", online e-book: http://www.erc.msstate.edu/publications/gridbook/index.html.

[13] Li Ma, Yunhong Wang, Tieniu Tan. "Iris Recognition Based on Multichannel Gabor Filtering", National Laboratory of Pattern Recognition, Institute of Automation.

[14] R. Wildes. "Iris Recognition: an Emerging Biometric Technology", Proceedings of the IEEE, Vol. 85, No. 9, 1997.

[15] Witold Zorski,Brain Foxon , Johnathan Blackledge, and Martin Turner. "Fingerprint and Iris Identification Method Based on the Hough Transform1", Cybernetics Faculty, Military University of Technology.

[16] Klaus D. Toennies, Frank Behrens, and Melanie Aurnhammer. "Feasibility of Hough-Transform-based Iris Localisation for Real-Time-Application", Dept. of Computer Science, Otto-von-Guericke Universität Magdeburg, Germany.

[17] A.C. Bovik, M.Clark, and W.S. Geisler. "Multichannl Texture Analysis Using Localized Spatial Filters", IEEE Transactions on Pattern Analysis and machine Intelligence, vol.12, pp. 55-73, 1990.

[18] A.Jain and G. Healey. "A Multiscale Representation Including Opponent Color Features for Texture Recognition", IEEE Transactions on Image Processing, vol. 7, no. 1, pp. 124-128, January 1998.

[19] Sitaram Bhagavathy, Jelena Te¡si'c, and B. S. Manjunath. "On The Rayleigh Nature of Gabor Filter Outputs", Department of Electrical and Computer Engineering, University of California, Santa Barbara.

[20] P. Kruizinga, N. Petkov and S.E. Grigorescu. "Comparison of Texture Features Based on Gabor Filters", Institute of Mathematics and Computing Science, University of Groningen.

[21] Tai Sing Lee. "Image Representation Using 2D Gabor Wavelets", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, No. 10, October 1996.

[22] Chinese Academy of Sciences – Institute of Automation. Database of 756 Greyscale Eye Images. From: http://www.sinobiometrics.com Version 1.0, 2003.

[23] J. Daugman, “Statistical Richness of Visual Phase Information: Update on Recognizing Persons by Iris Patterns”, Int’l J. Computer Vision, vol. 45, no. 1, pp. 25-38, 2001.

[24] W. Boles and B. Boashash, “A Human Identification Technique Using Images of the Iris and Wavelet Transform”, IEEE Transaction on Signal Processing, vol. 46, no. 4, pp. 1185-1188, 1998.

[25] N. Ritter. "Location of the Pupil-Iris Border in Slit-Lamp Images of the Cornea", Proceedings of the International Conference on Image Analysis and Processing, 1999.

[26] Vladimir Vezhnevets and Anna Degtiareva. "Robust and Accurate Eye Contour Extraction", Graphics and Media Laboratory, Faculty of Computational Mathematics and Cybernetics, Moscow State University, Moscow, Russia.

[27] W. Kong, D. Zhang. "Accurate iris segmentation based on novel reflection and eyelash detection model", Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing, Hong Kong.