

Iris Based Human Verification Algorithms

Bhola Ram Meena, Mayank Vatsa, Richa Singh, and Phalguni Gupta
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur
Kanpur – 208016, INDIA

{brmeena, mayankv, richas, pg}@cse.iitk.ac.in

Abstract. In this paper three algorithms for iris verification have been presented. Iris detection algorithms include the normalization and iris extraction steps. Three algorithms for verification process are (a) Algorithm using radial and circular features, (b) Algorithm using Fourier transforms and (c) Algorithm using Circular–Mellin transforms. Proposed algorithms have been tested on CASIA database and some non–infrared Iris images. The experimental results show that the algorithm based on Circular – Mellin Transform gives the best result with an accuracy of 95.45%. Some initial experiments on non–infrared iris images shows that this algorithm can work on such images but it still requires some more attention and this is our future work.

1 Introduction

Most of the biometrics authentication systems are based on signal and image processing based algorithms. Among the various traits, iris recognition has attracted a lot of attention because it has various advantageous factors like greater speed, simplicity and accuracy compared to other biometric traits. Iris recognition relies on the unique patterns of the human iris to identify or verify the identity of an individual.

For iris recognition, an input video stream is taken using Infra-red sensitive CCD camera and the frame grabber. From this video stream eye is localized using various image processing algorithms. Area of interest i.e. iris is then detected from the eye and the features are extracted. These features are encoded into pattern which is stored in the database for enrollment and are matched with the database for authentication.

This paper presents three new algorithms for iris recognition. The first algorithm is based on extracting circular and radial features using edge detection while in the second algorithm features are extracted using Fourier transforms along radial direction. The third algorithm is based on feature extraction with Circular-Mellin operators. These operators are found to be scale invariant and rotational invariant and convolution with them results in shift invariant features.

In Section 2 various steps of the proposed iris verification algorithms are presented. Experimental results of the proposed algorithms are given in Section 3. The experiments have been carried out on the CASIA image database [4] provided by Prof. T. Tan and some iris images taken in non – infrared lighting environment by a CCD Camera. Last section is the conclusion and future work.

2 Iris Verification Algorithm

This section describes various steps of our iris verification algorithm which consists of two sub-algorithms: Iris Detection and Matching Iris Patterns. Iris Detection involves detection of pupil boundary and outer boundary of iris. The other algorithm extracts the feature vector and performs matching.

2.1 Iris Detection

The first step in iris detection (also known as preprocessing) is to detect the pupil. The center of pupil can be used to detect the outer radius of iris patterns. Usually the image captured for iris has many undesired parts like eyelids, pupil etc. This step also involves minimizing noise in iris and making iris scale invariant, i.e. iris patterns independent on size of input image. This can be achieved by converting iris in polar coordinates and normalizing it radially.

Detection of Inner Boundary. Inner boundary of iris is detected by detecting pupil. First the gray scale input image is changed to binary format by using a suitable tight threshold. Assuming that circular area of pupil is the largest black circular part, pupil is detected by searching for largest black circular part in binary image. Points are evenly initialized on image for searching largest black circular area around it and after several iterations we can get the largest circular black area and the boundary of this circular area is the inner boundary (Fig. 1)

Detection of Outer Boundary. The outer boundaries of iris are detected with the help of center of pupil. The binary image is taken and concentric circles of different radii are drawn with respect to center of pupil. For a particular circle the change in intensity between normal pointing toward center and away from center is measured. The radius having highest change in intensity is considered as outer boundary. Fig. 1 shows the change in intensity along normal for a circle.

Iris in Polar Coordinates. As iris patterns lies between two circle concentric at center of pupil. By changing image coordinates to polar coordinates we can remove useless pupillary part. In polar coordinates iris can be seen as rectangular plate. Let (x, y) be any point on input image with respect to center of pupil, which lies between inner and outer boundaries of iris. Let $f(x, y)$ be pixel value of point (x, y) then in polar

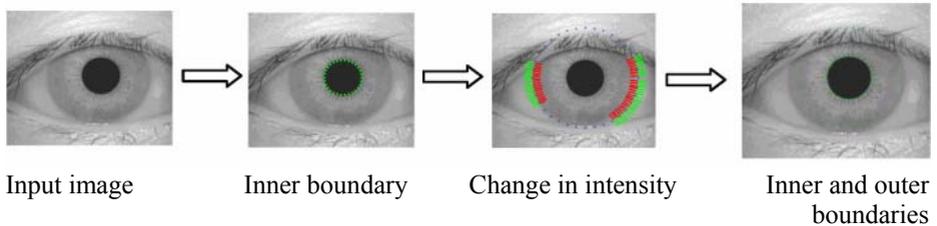


Fig. 1. Iris Detection

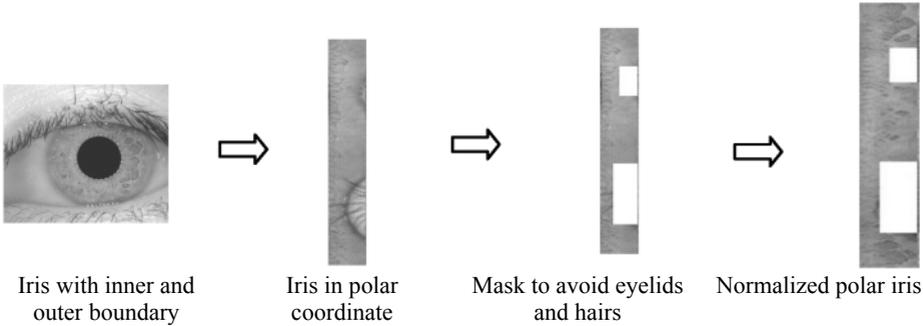


Fig. 2. Getting iris in polar coordinates and reconstruction of iris image

coordinates it will map to the points (r, θ) where

$$r = \sqrt{x^2 + y^2} \quad \text{and} \quad \theta = \tan^{-1}(y/x) \quad \text{for} \quad \theta \in (-\pi, \pi] \tag{1}$$

Figure 2 shows an example of the iris in polar coordinates. There are some errors in the iris image due to presence of eyelids. We can get rid of these eyelids and hairs by using mask on the polar image. We used rectangular masks to remove noise due to eye lids. Assuming iris boundary as circle we found that upper eye lids come in $60^\circ - 120^\circ$ part of upper iris region and lower eye lids come in $255^\circ - 285^\circ$ of lower iris region. We have consider the 75% gap between pupil and iris boundary as noise in upper iris region and 50% gap in lower region as noise.

Iris Normalization. Iris normalization is done to make iris image in polar coordinates independent of size of the input image. Polar iris image is normalized by fixing number of points along radius. The constant number of points along radius of iris is known as normalization constant (N). For normalization all r calculated in Eq. (1) are mapped between radius 0 to N.

$$r_{norm} = (r - r_{min}) \frac{N}{r_{max} - r_{min}} \quad \text{where} \quad r_{max} = \text{Maximum}(r) \quad \text{and} \quad r_{min} = \text{Minimum}(r) \tag{2}$$

It may happen that more than one point on polar image map to same value of r_{norm} and θ then pixel value of the point $[r_{norm}, \theta]$ is assigned by the minimum pixel value of these points. On the other hand it may happen that for a particular r_{norm} and θ no point in polar image maps to it then pixel value of this point is taken as minimum pixel value of its neighbor normalized polar points. Fig. 2 shows the example of iris normalization.

2.2 Algorithms for Verification

Each of the proposed algorithms for verification consists of two steps: (a) Iris code generation and (b) Matching Iris Codes. First step is used to extract features from the normalized iris image given in polar coordinates and to generate an Iris code. Second step deals with matching of the Iris code generated for query image with the other available in database.

Three algorithms have been presented and analyzed. The first algorithm is based on extracting circular and radial features using edge detection while in the second one, features are extracted using Fourier transforms along radial direction. Feature extraction in the third algorithm is done using Circular-Mellin transform.

Verification Based on Circular and Radial Features. This algorithm is based on edge detection. Edges are detected in input image using canny edge detector [1]. After edge detection image is changed to binary format in which white pixels are present on edges and black pixels elsewhere. The number of white pixels in radial direction and on circle of different radius gives important information about iris patterns. Normalized polar iris image will contain only white and black pixels as it is obtained from above edge detected input image.

Extracting Radial Features. In iris image value of feature at particular angle is the number of white pixels along the radial direction. If we consider

$$S_{r,\theta} = \begin{cases} 1 & \text{iris_polar_image}[r][\theta] = \text{WHITE} \\ 0 & \text{iris_polar_image}[r][\theta] = \text{BLACK} \end{cases} \quad (3)$$

then feature at angle θ will be

$$F_{\theta} = \sum_{r=1}^N S_{r,\theta} \quad (4)$$

Extracting circular features. In iris image, value of feature for particular radius is considered as sum of white pixels along the circle of that radius and keeping the meaning of $S_{r,\theta}$ same as in Eq. (3). The feature of particular radius r will be given as $F_r = \sum_{\theta=0}^{360} S_{r,\theta}$. Figure 3 shows the graphical representation of radial and circular features of the iris. Iris code is generated as vector of radial and circular features.

Matching. Each Iris code can be considered as signal and then the matching of two Iris codes become the problem of comparing two signals. Absolute difference between two Iris codes is used to compare two Iris codes. Let us define

$$\Delta F_{\theta} = \text{Minimum} \left\{ \sum_{\theta=0}^{360} |F_{\theta+\phi}^1 - F_{\theta}^2| \right\} \quad \text{for } -10 \leq \phi \leq 10 \quad (5)$$

$$\Delta F_r = \sum_{r=1}^N |F_r^1 - F_r^2| \quad (6)$$

Equation (5) differs from Eq. (6) because radial features can be rotated if input image is rotated. Assuming that there is no rotational error of more than 10° , then the final difference is taken as minimum of all differences considering the rotational error.

If $\alpha \Delta F_{\theta} + \beta \Delta F_r \leq \text{Threshold}$ then both iris codes are of same person where α and β are weighted parameters. The reason for using different weight for radial and circular features is that circular features are scale and rotational invariant while radial features are not rotational invariant.

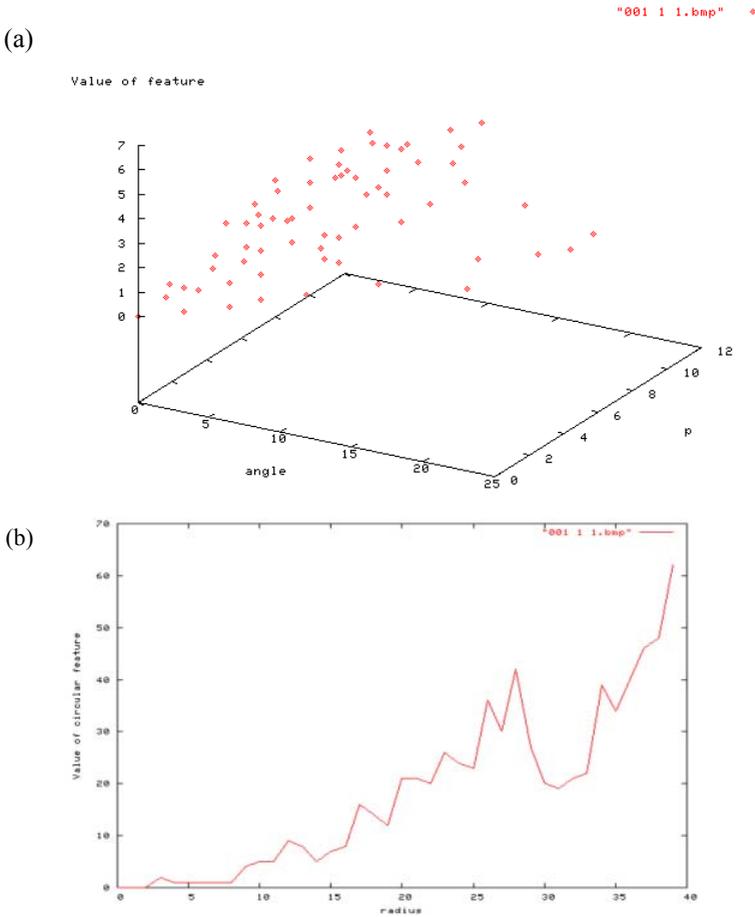


Fig. 3. (a) radial features of iris, (b) circular features of iris

Verification Based on Fourier Transforms Along Radial Directions. This algorithm is also based on edge detection. At first, edges are detected in input image which changes it in binary format. One minor change is done in normalized polar iris image that white pixel has the value 1 instead of 255. The reason for using Fourier transforms on radial direction instead on angular direction is that rotational error can be coped at matching time but not the radial error introduce due to extraction of iris from eye image. Fourier coefficients extract the features along radial direction. Radial features are more important because most of the iris images are noise free.

Generating Two Dimensional Iris Code. Let $I(r, \theta)$ be modified normalized polar iris image. Fourier transforms are applied in radial directions for getting features using Eq. (7)

$$Z = \int_r I(r, \theta) e^{-jpr} dr \tag{7}$$

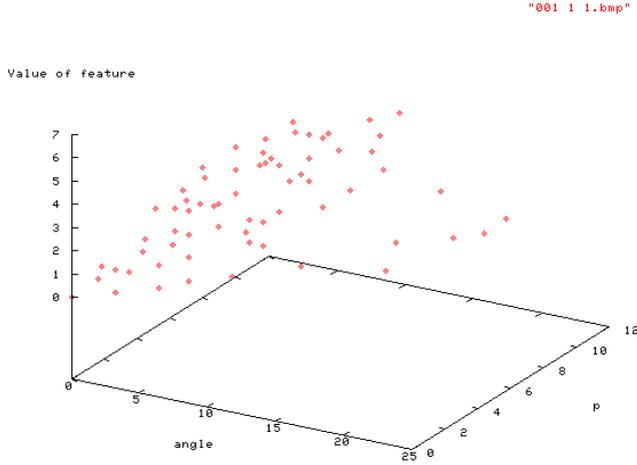


Fig. 4. 2 D Feature Vector

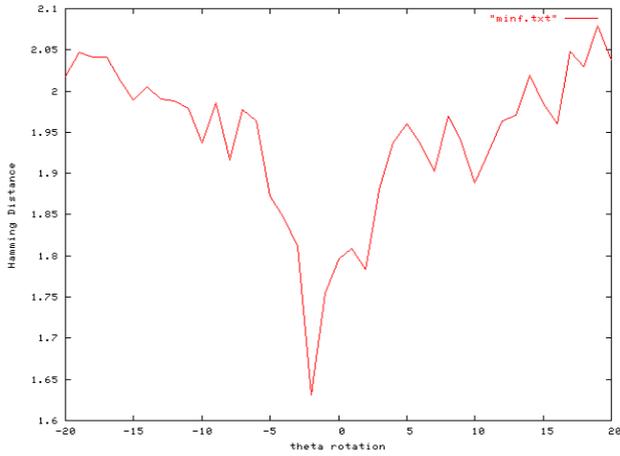


Fig. 5. Hamming distance for different rotational error

The value of Iris code for particular θ_0 and p will be integral part of $|Z|$. For $0 \leq \theta_0 \leq 360$ and $1 \leq p \leq P$ we can create a two dimensional feature vector as shown in Fig. 4.

Matching. Hamming distance between two iris codes is used to determine whether they belong to same person or not. Let A and B be two iris codes to be matched. Hamming distance [2] is calculated as follows:

$$HD = \text{Min} \left\{ \sum_{i=0}^M \sum_{j=0}^N (A(i + \phi, j) - B(i, j)) \right\} \quad \text{for} \quad -10 \leq \phi \leq 10 \quad (8)$$

where M is size of angular part generally 360 and N is value of P as in Eq. (7) taken at time of creating 2-dimensional iris code. If $HD \leq threshold$ then the given two iris belong to the same person. Fig. 5. shows the graph for selecting the threshold value for hamming distance for different rotational error.

Verification Using Circular-Mellin Operators. The Circular Mellin operator [3] is given by

$$h(\lambda, \theta) = e^{-\lambda} e^{j(2\pi P\lambda + Q\theta)} \quad j = \sqrt{-1} \tag{9}$$

where constants P and Q controls the weights given to radial and circular features of image. Circular - Mellin Operators are scale and rotational invariant. Convolution of iris image with it makes it shift-invariant. There is no use of scale invariance of Mellin operator as polar iris images are normalized to be scale invariant. Features in iris image are extracted based on phase of its convolution with Mellin operators. The polar image of iris given by $P(r, \theta)$ is convolved with Circular - Mellin operator.

$$Z = \int_r \int_\theta P(r, \theta) \frac{r_0}{r^2} e^{j(2\pi p \log(r/r_0) + q(\theta - \theta_0))} d\theta dr \tag{10}$$

where $Z = Re(Z) + j Im(Z)$ and p, q are parameters controlling behavior of filter. Iris code is generated based on phase of above Z . Iris code at point (r_0, θ_0) is taken as

$$iriscode[r_0, \theta_0] = \begin{cases} 1 & Re(Z) * Im(Z) \geq 0 \\ 0 & Re(Z) * Im(Z) < 0 \end{cases} \tag{11}$$

Fig. 6. shows the iris code generation from polar image.

Matching. The noise in iris also introduces the noise in iris code. A new method for comparing two iris patterns has been presented. As iris code generated by our algorithm is two dimensional array of bits. Instead of comparing it bit by bit, window-wise comparison is done. In every corresponding window in both the Iris codes the number of 1's present is counted. If the difference in number of 1's in corresponding windows is greater than a threshold then difference between them is increased by one. Let us consider

$$D_{i,j} = \begin{cases} 1 & (IC^1(i, j) \otimes K) - (IC^2(i, j) \otimes K) \geq thresh \\ 0 & (IC^1(i, j) \otimes K) - (IC^2(i, j) \otimes K) < thresh \end{cases} \tag{12}$$

where K is kernel given by $K = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ and \otimes stands for convolution. IC stands for

iris code. Now if M and N are dimensions of iris code then the total difference is calculated by

$$Difference = \sum_{i=1}^M \sum_{j=1}^N D_{i,j} \tag{13}$$

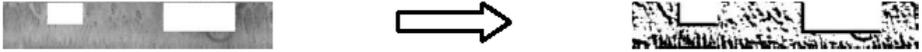


Fig. 6. Generation of iris code.

Dynamically the value of the threshold has been selected. It varies person to person. For calculating local threshold, distance between Iris codes of same person has been calculated using above method and finally local threshold is computed as

$$\text{Local Threshold} = \text{Max. Difference} + 0.5 * (\text{Max. Difference} - \text{Avg. Difference}) \quad (14)$$

3 Results

We have tested these above mentioned algorithms on the CASIA image database [4] and some iris images taken in non – infrared lighting environment by a CCD Camera.

The accuracy of iris detection algorithm is 100%. Table 1 shows the experimental results for verification using the three algorithms. Among the three algorithms, Circular Mellin Transform based algorithm gives the best results with an overall accuracy of 95.45%. Fig. 7.(a) shows results for the Circular – Mellin transform on CASIA Database and Fig. 7 (b) on the non-infrared images.

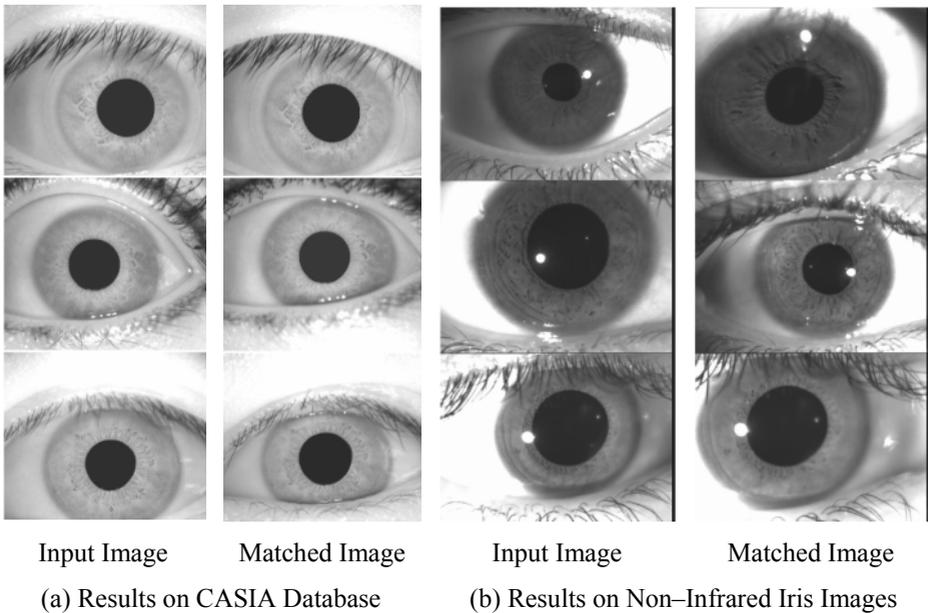


Fig. 7. Experimental Results of Circular – Mellin Transform

Table 1. Results of the Three Algorithms

Algorithm	FRR	FAR
Radial and circular features	1.89 %	23.26 %
Fourier Transforms	6.77 %	21.05 %
Circular – Mellin Transforms	3.52 %	5.58 %

4 Conclusion

In this paper new algorithms for iris verification have been presented. We have shown three algorithms for verification process namely 1 - Radial and circular features, 2 - Fourier Transforms and 3 Circular – Mellin Transforms. The experiments carried out on our experiments on CASIA database and some non – infrared Iris images show that the algorithm based on Circular – Mellin Transform gives the best results in terms of false reject accuracy and false accepts accuracy and hence an overall accuracy of 95.45%. Some initial experiments on non–infrared iris images shows that this algorithm can work on such images but it still requires some more attention.

Acknowledgement

Authors wish to acknowledge Prof. T. Tan for providing CASIA database and Mr. S. Sukumar for providing non – infrared iris images. This work has been done as a part of the project sponsored by the Department of Communication and Information Technology, INDIA.

Reference

- [1] Canny J.: A computational approach to edge detection, IEEE transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, (1986) 679 - 698.
- [2] Daugman, J.: High confidence visual recognition of persons by a test of statistical independence, IEEE transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 11, (1993) 1148–1161.
- [3] Gopalan Ravichandran, Mohan M. Trivedi: Texture Segmentation using Circular-Mellin Operators. Proceedings of ICIP Vol. 2 (1994) 635-639.
- [4] <http://nlpr-web.ia.ac.cn/english/irds/irisdatabase.htm>.