Abstract

Human iris is one of the most reliable biometric because of its uniqueness, stability and non-invasive nature. Biometrics based human authentication systems are becoming more important as government & corporations worldwide deploy them in such schemes as access & border control, time & attendance record, driving license registration and national ID card schemes. For this various preprocessing steps are carried out on the iris image which also includes segmentation. Normalization deals with polar to rectangular conversion. The edges are detected using canny edge detector. Features are extracted using ridge energy direction algorithm. It uses two directional filters i.e. horizontal and vertical oriented. The final template is generated by comparing the two templates and considering the predominant edge. This final template is match with the stored one using hamming distance and the match ID is displayed.

1. Introduction

Biometric solutions address the security issues associated with traditional method of human recognition based on identity card, secret password or personal identification number (PIN) etc. These methods face sever problems such as ID may be forged, keys may be lost and password may be forgotten. Biometric measures based on physiological or behavioural characteristics are unique to an individual and has ability to distinguish between authorized person and an imposter. The physiological characteristics include finger print, retina, palm prints, face, DNA, hand geometry and iris recognition while the behavioural characteristics include handwriting, signature, gesture, keystrokes and gait [1].

The behavioural characteristics fail in many cases as the characteristics can easily be learnt and changed by practice. Some of the techniques based on physiological characteristics such as finger prints, face recognition and hand geometry also fail when used over a long time as they may change due to ageing or cuts and burns. Among all the biometric techniques iris recognition has drawn a lot of interest in Pattern Recognition and Machine Learning research. As it is more unique, stable, universal and do not depend upon genetics [1], [2].

The iris formation starts in the third month of gestation period and is largely complete by the eighth month and then it does not change after two to three years. The forming of iris depends on the initial environment of the embryo and hence the iris pattern does not correlate with genetic determination. Even the left and the right irises of the same person are unique. It is non-invasive and almost impossible to modify the iris structure by surgery. Iris is the only internal organ which can be seen outside the body. Hence it does not require any contact with the sensing device. The iris verification can be divided into four stages: (i) Data Acquisition, (ii) Preprocessing (includes segmentation) (iii) Feature Extraction (encoding) and (iv) Matching [3], [4], [5].

2. Methodology

The first and foremost step is to collect the iris images. On these images various preprocessing steps are carried out. It includes conversion of color image to gray scale, histogram equalization and segmentation. Polar to rectangular conversion is applied and on this rectangular template RED algorithm is applied which generates the template. These templates are match with the stored one using hamming distance and the match ID is displayed. The flow of process is shown below:
2.1. Capture Image

Generally iris images are captured using 3CCD camera working at near infrared (NIR) light. Here we have collected different color images of iris from net on which the RED algorithm is applied.

2.2. Preprocessing

2.2.1. Gray Images. The color image is converted into gray shade using the formula:

$$Y = 0.3\text{RED} + 0.59\text{GREEN} + 0.11\text{Blue}$$  \hspace{1cm} (1)

The brightness or luminance of color image is converted into shade of gray \cite{6}, \cite{7}.

2.2.2. Histogram Equalization. Here the contrast of the images is uniformly distributed to enhance the quality of image. The histogram of a segmented image \(H[n]\) is then computed. Since the segmented image contains primarily zero pixel values, and the pupil itself has very low values, the histogram is modified to remove the effects of these pixels. This modification is described as: \cite{8}.

\[
H_i[n] = \begin{cases} 
0, & n < 20 \\
H_i[n], & 20 \leq n \leq 230 \\
255, & n > 230 
\end{cases} \quad (2)
\]

In histogram equalization the number of pixel of same intensity is stored in one array which is commonly called as "bin". For an 8-bit gray scale image, the size of histogram bin is 256, because the range of the intensity of 8-bit image is from 0 to 255. Histogram is a useful tool to analyze the brightness and contrast of an image. Histogram equalization works best on an over or under exposed image, which has narrow contrast range. In histogram equalization, the input pixel intensity, \(x\) is transformed to new intensity value, \(x'\) by \(T\). The transform function, \(T\) is the product of a cumulative histogram and a scale factor. The scale factor is needed to fit the new intensity value within the range of the intensity values, for: \cite{9}.

\[
x' = T(x) = \sum_{i=0}^{\text{max. intensity}} \frac{n_i}{N} \quad (3)
\]

where: \(n_i\) is the number of pixel at intensity \(i\), \(N\) is the total number of pixel in the image (0-255).

2.2.3. Segmentation. In this the pupil is separated using canny edge detector. It detects the boundary of pupil and iris using gradient change concept. Based on the heuristic that there will be only two important gradients in the region (pupil–iris, iris–sclera) and pupil pixels will be the darkest, iris pixels will be intermediate and sclera pixels will be whitest, this way we can look for the second gradient and take it as iris estimated radius. The left and right boundaries of the iris are found by selecting the largest gradient change to the left and right of the pupil. Wildes has adopted a two-step method to localize iris: edge map detection followed by edge-based segmentation. This iris segmentation approach is based on the fact that the pupil is typically darker than the iris and the iris is darker than the sclera. Starting from the pupil center \((c_1, c_2)\) two regions to look for jumps in gray scale level are defined as: \cite{2}

\[
W_l = W_1, W_2 \quad (4)
\]
\[
W_i = W_1, W_2 \quad (5)
\]

Where: \(W_l\) and \(W_i\) are rectangle based on the two coordinates in the original image. As shown in Fig. 2, the region represents the pupil-iris and iris-sclera gradient change \cite{2}.
Figure 2: Is the original image with the projected right region (in white), as we can see Zone A represents pupil area with a dramatic change of gradient (pupil – iris gradient), Zone B represents the clearer pixels in the iris pattern Zone C represents the gradient change between iris and sclera followed by much whiter pixels of the sclera in Zone D.

In order to implement the canny edge detector algorithm, a series of steps must be followed. The first step is to filter out any noise in the original image before trying to locate and detect any edges. The Gaussian filter can be computed using a simple mask; it is used exclusively in the canny algorithm. Once a suitable mask has been calculated, the Gaussian smoothing can be performed using standard convolution methods. A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise.

The above gradient is then approximated using the formula:

$$|G| = |G_x| + |G_y|$$  \hspace{1cm} (6)

The direction of the edge is computed using the gradient in the ‘x’ and ‘y’ directions. The formula for finding the edge direction is as follows:

$$\theta = \tan^{-1} \frac{G_y}{G_x}$$  \hspace{1cm} (7)

Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image. So if the pixels of a 5x5 image are aligned as follows: [10], [11]

$$\begin{bmatrix}
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & a & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times 
\end{bmatrix}$$

Figure 3: Matrix for detecting edge

Then, it can be seen by looking at pixel “a”, there are only four possible directions when describing the surrounding pixels - 0°, 45°, 90°, or 135°. So now the edge orientation has to be resolved into one of these four directions depending on which direction it is closest to.

Therefore, any edge direction falling within the yellow range is set to 0°, falling in the green range is set to 45°, falling in the blue range is set to 90°. And finally, any edge direction falling within the red range is set to 135°.

After this non-maximum suppression is applied. Non-maximum suppression is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge. This will give a thin line in the output image.

Finally, hysteresis is used as a means of eliminating streaking. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold. If a single threshold, T1 is applied to an image, and an edge has an average strength equal to T1, then due to noise, there will be instances where the edge dips below the threshold. Equally it will also extend above the threshold making an edge look like a dashed line. To avoid this, hysteresis uses 2 thresholds, a high and a low. Any pixel in the image that has a value greater than T1 is presumed to be an edge pixel, and is marked immediately. Then, any pixels that are connected to this edge pixel and that have a value greater than T2 are also selected as edge pixels [10], [11].

The steps required for canny detector are as follows:

- Smooth by Gaussian:
  $$S = G_\sigma \ast I$$  \hspace{1cm} (8)
  $$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}$$  \hspace{1cm} (9)

- Compute x and y derivatives:
  $$\nabla S = \begin{bmatrix}
\frac{\partial}{\partial x} S & \frac{\partial}{\partial y} S
\end{bmatrix}^T = [S_x, S_y]^T$$  \hspace{1cm} (10)

- Compute gradient magnitude and orientation
  $$|\nabla S| = \sqrt{S_x^2 + S_y^2}$$  \hspace{1cm} (11)
\[ \theta = \tan^{-1} \left( \frac{S_y}{S_x} \right) \] (12)

- Canny edge operator:
\[ S = G_\sigma \ast I \] (13)
\[ \nabla G_\sigma = \left[ \frac{\partial G_\sigma}{\partial x} \frac{\partial G_\sigma}{\partial y} \right] \] (14)
\[ \nabla S = \left[ \frac{\partial G_\sigma}{\partial x} \frac{\partial G_\sigma}{\partial y} \right] \ast I \] (15)

- Non-maximum suppression: we wish to mark points along the curve where the magnitude is biggest and suppress the pixels in “Gradient Magnitude Image” which are not maximum.
\[ M(x, y) = \begin{cases} 
\nabla S(x, y) & \text{if } |\nabla S(x, y)| > |\nabla S(x', y')| \\
& \& |\nabla S(x, y)| > |\nabla S(x'', y'')| \\
0 & \text{otherwise}
\end{cases} \] (16)

Where \((x', y')\) and \((x'', y'')\) are the neighbours of \((x, y)\) along the direction normal to an edge.
After this hysteresis thresholding is done. This includes following stages: [30],
- If the gradient at a pixel is above ‘High’, declare it an ‘edge pixel’
- If the gradient at a pixel is below ‘Low’, declare it a ‘non-edge pixel’
- If the gradient at a pixel is between ‘Low’ and ‘High’ then declare it an ‘edge pixel’ if and only if it is connected to an ‘edge pixel’ directly or via pixels between ‘Low’ and ‘High’

2.2.4. Polar to rectangular conversion. After separating the pupil the polar to rectangular conversion is applied this generates the rectangular template as shown below:

![Figure 5: Polar to rectangular co-ordinate conversion](image)

Converting pixel from an iris image to rectangular template was performed using the common polar to rectangular coordinate transformation this process is often called as unwrapping or normalization. A pair of dimensional real co-ordinates \((r, \theta)\) where \(r\) lies in the unit interval \([0, 1]\) and \(\theta\) is the usual angular quantity that is cyclic over \([0, 2\pi]\). This model was given by Daugman also called as rubber sheet model. The rubber sheet model takes into account pupil dilation and size inconsistencies in order to produce a normalized representation with constant dimension. In this way the iris region is modelled as a flexible rubber sheet anchored at the iris boundary with the pupil center as the reference point [5].

The dilation and constriction of the elastic mesh work on iris when the pupil changes size. The homogeneous rubber sheet model assigns to each point on the iris, regardless of its size and papillary dilation, a pair of real co-ordinates \((r, \theta)\) where is on the unit interval \([0, 1]\) and ‘\(\theta\)’ is angle \([0, 2\pi]\). The remapping of the iris image \(I(x, y)\) from raw Cartesian coordinates \((x, y)\) to the dimensionless non-concentric polar coordinate system \((r, \theta)\) can be represented as: [5], [12], [13].

\[ I(r, \theta) \to I(r, \theta) \] (17)

Where \(x\) \((r, \theta)\) and \(y\) \((r, \theta)\) are defined as linear combinations of both the set of pupillary boundary points \((x_s(\theta), y_s(\theta))\) and the set of limbus boundary points along the outer perimeter of the iris \((x_p(\theta), y_p(\theta))\) as follows:

\[ x(r, \theta) = (1 - r)x_s(\theta) + rx_p(\theta) \] (18)
\[ y(r, \theta) = (1 - r)y_s(\theta) + ry_p(\theta) \] (19)

Since the radial coordinate ranges from the inner boundary to its outer boundary as a unit interval, it inherently corrects for the elastic pattern deformation in the iris when the pupil changes in size [12], [13].

\[ r = \sqrt{x^2 + y^2} \] (20)

where:
\[ x = r \cos \theta \] (21)
\[ y = r \sin \theta \] (22)

This converts the polar image to rectangular on which desired algorithm is applied for feature extraction.

2.3. Feature Extraction

The Ridge Energy Direction (RED) algorithm used for iris recognition is currently being developed at the U.S. Naval Academy by Dr. Robert Ives et.al. Feature extraction is based on the prominent direction of the ridges that appear on the image the polar coordinates are converted into rectangular co-ordinates and transformed into an energy image. We refer this feature extraction as the Ridge Energy Direction (RED) algorithm [14].

To perform the filtering the input data is further passed through a periodic array taking in 81 \((9 \times 9)\) values at a time. More specifically, the result is computed by first multiplying each filter value by the corresponding input data value. Then a summation is performed, and the result is stored in a memory location that corresponds to the centroid of the filter. This process repeats for each pixel in the input data.
stepping right, column-by-column, and down, row-by-row, until the filtering is completed. The step is demonstrated in Fig. 6 [15]. Finally, the template is generated by comparing the results of two different directional filters (horizontal and vertical) and writing a single bit that represents the filter with the highest output at the equivalent location. The output of each filter is compared and for each pixel, ‘1’ is assigned for strong vertical content or ‘0’ for strong horizontal content. These bits are concatenated to form a bit vector unique to the “iris signal” that conveys the identifiable information. This final step is demonstrated in Fig. 7 and Fig. 8 [16], [17].

Figure 6: 9×9 filters computing the filtering of the top left portion of hypothetical input image data

In Fig. 7, each coefficient of the filter is multiplied by the corresponding image data within the scope of the filter where some of the data is repeated from the opposite side. These filter co-efficient and input data is repeated from the opposite side. These filter coefficient and input data products make up a partial result, the sum of which generates a local result corresponding to the centroid of the filter [18], [19].

Figure 7: Example input passed into two directional filters

2.4. Template Matching

The template can now be compared with the stored template using Hamming distance (HD) as the measure of closeness.

\[
\text{HD} = \frac{\| (\text{template A} \bigotimes \text{template B}) \cap \text{mask A} \cap \text{mask B} \|}{\| \text{mask A} \cap \text{mask B} \|} \tag{23}
\]

Where \( \bigotimes \) symbol indicates the binary exclusive-or operator to detect disagreement between the bits that represent the directions in the two templates, \( \cap \) is the binary AND function \( \bullet \) is a summation, and mask A and mask B are associated binary masks for each template. The denominator ensures that only required bits are included in calculation [18], [19].

The very famous and highlighted person of Iris based Image processing is John Daugman, who conducted his tests on a very large number of iris patterns (up to 3000000 iris images) and deduced than the maximum Hamming distance that exists between two irises belonging to the same person is 0.32. The test is conducted based on the following way, initially binary feature vectors are passed to a mathematical function, secondly after getting the numeric value it is compared with the Hamming Distance between the things. Finally the decision is made up, with the following results.

- If \( \text{HD} \leq 0.32 \) decides that it is same person
- If \( \text{HD} > 0.32 \) decides that it is different person

3. Result

Stage 1: Original Image: The original image is colored one as displayed below:

Figure 9: Color image
Stage 2: Gray Image: The above color image is converted into gray shade image as shown in Fig. 10.

Stage 3: Enhance image: On the above image histogram equalization is carried out as shown below:

Stage 4: Edge Detection: Apply canny edge detector to the above image to detect the boundary of pupil, iris and edges as displayed in Fig. 12.

Stage 5: Start and end point of image: Find the start and end point of the image and obtain the center of the image as displayed below:

Stage 6: Center of pupil: Obtain the center of pupil using the above center point of image as trace below:

Stage 7: Center and radius of pupil: Display the center and radius of pupil as shown in Fig. 15.

Stage 8: Area of interest: Obtain area of interest from which features are to be extracted as below:

Stage 9: Binary image: Convert the above image into binary image as follows:

Stage 10: Region of interest: Separate area of interest from which features are to extracted is shown below:
Stage 11: Polar to rectangular conversion: Convert the above image into rectangular template as displayed below:

![Figure 19: Polar to rectangular conversion](image)

Stage 12: Horizontal and vertical template: Apply the RED algorithm on the above image and extract the features using horizontal and vertical filters as displayed below:

![Figure 20: Horizontal and vertical template](image)

Stage 13: ID: Display the match ID of the person. Here ‘ind’ indicates ID number.

\[ \text{ind} = 1, \text{ind} = 5 \]  

4. Conclusion

The RED algorithm is used to reduce the effect of illumination. The algorithm is tested on different database along with this different colored images are applied whose results are shown in this paper. The algorithm works well in noise free condition. False reject is equal to 0% as it does not reject the image which is present in the database. Hence accuracy is equal to 100% as it is able to detect the correct ID for the images which are present in the database.

5. References


