Iris Recognition Using Wavelets and Integro Differential Method

MS COMPUTER SOFTWARE ENGINEERING

By

Qazi Sami ul Haq

Advisor

Brig. Dr. Muhammad Younus Javed

Department of Computer Engineering
College of Electrical & Mechanical Engineering
National University of Sciences and Technology (NUST)
2008

DEDICATION

Iris Recognition Using Integro Differential Method And Wavelets

Abstract

This research paper deals with the implementation of iris recognition using Integro differential method and wavelets on iris images. The proposed system contains three parts i.e. preprocessing, feature extraction and matching. The preprocessing part further contains pupil localization, iris localization and normalization processes. First the centre coordinates and pupil is calculated. In the next phase iris area is calculated using Integro differential method and ellipse as the contour model i.e. the parameter form of ellipse is used. Then the iris area is converted to normalized template. After that feature extraction process takes over in which the rows of input normalized template are taken as signals and then these are convolved with the gabor filter or wavelets. The output of this process is encoded in the form of feature template. Then this feature template of the input iris image is stored in the database. When the iris identification is required then the input image is compared with the feature templates that are stored in database using hamming distance which gives the ratio that describes the no of pixels that are different in both templates.

The proposed algorithm was tested on CASIA database. The empirical results provide the accuracy of 96% with time delay of 0.066134 sec per image. The comparison of the proposed technique with other well known techniques is provided in the thesis both with respect to time and performance in the form of graphs. It is evident from the comparison that the proposed technique performs better not only with respect to accuracy but time also. The robustness and time efficiency of the proposed algorithm makes it perfect candidate for real time applications.

Acknowledgements

Most of all I m grateful to ALLAH tallah for his beneficence and mercy, then I am highly obliged to my supervisor Brig. Dr. Muhammad Younus Javed for his motivation, guidance, suggestions, and support anytime in the study and in my research work and giving out his precious and valuable time. I like to thank my MS Thesis committee for their willingness to serve on my MS Thesis

Table of Contents

List of Figures

Chapter 1: Introduction

1. Biometric Technology

A biometric system provides automatic recognition of an individual based on some sort of unique feature or characteristic possessed by the individual. Biometric systems have been developed based on fingerprints, facial features, voice, hand geometry, handwriting, the retina and the one presented in this thesis, the iris.

Biometric systems work by first capturing a sample of the feature, such as recording a digital sound signal for voice recognition, or taking a digital color image for face recognition. The sample is then transformed using some sort of mathematical function into a biometric template. The biometric template will provide a normalized, efficient and highly discriminating representation of the feature, which can then be objectively compared with other templates in order to determine identity. Most biometric systems allow two modes of operation. An enrolment mode for adding templates to a database, and an identification mode, where a template is created for an individual and then a match is searched for in the database of pre-enrolled templates.

A good biometric is characterized by use of a feature that is; highly unique – so that the chance of any two people having the same characteristic will be minimal, stable – so that the feature does not change over time, and be easily captured – in order to provide convenience to the user, and prevent misrepresentation of the feature.

1.1 Biometric Types

Commonly implemented or studied biometric modalities include fingerprint, face, iris, voice, signature and hand geometry. Many other modalities are in various stages of development and assessment. There is not one biometric modality that is best for all implementation. Many factors must be taken into account when implementing a biometric device including location security risks, task (identification or verification), expected number of users, user circumstances, existing data, etc. It is also important to note that biometric modalities are in varying stages of maturity. Some of the major modalities include

1.1.1 Iris Recognition

Iris recognition is the process of recognizing a person by analyzing the random pattern of the iris and by far it is the best biometric technique available. The iris is a muscle within the eye that regulates the pupil, controlling the amount of light that enters eye. It is colored portion of the eye with coloring based on the amount melatonin pigment within the muscle.

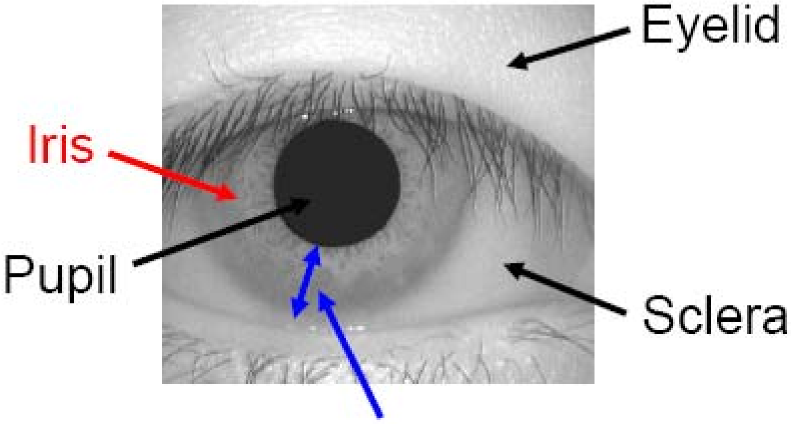


Figure 1.1: An eye image

Although the coloration and the structure of the iris are genetically linked, the details of the pattern are not. The iris develops during parental growth through a process of tight forming and folding of the tissue membrane. Prior the birth degeneration occurs, resulting in the pupil opening and the random, unique patterns of the iris. Although genetically identical, an individual’s irises are unique and structurally distinct, which allows it to be used for recognition purposes. To obtain a good image of the iris, recognition system typically illuminates the iris with near-infrared light, which can be observed by most cameras yet is not detectable by, nor can it cause injury to human eye.

1.1.2 Face Recognition

Humans often use faces to recognize individuals and advancements in computing capability over the past few decades now enable similar recognitions automatically. Early face recognition algorithms used simple geometric models, but the face recognition process has now matured into a science of sophisticated mathematical representations and matching processes. Major advancements and initiatives in the past ten fifteen years have propelled face recognition technology into the spot light. Multiple approaches exist since several years for 2D face recognition.

1.1.3 Finger Print Recognition

Recognition through fingerprints is a well-known biometrics technique. Because of their uniqueness and consistency over time, finger prints have been used for identification for over a century, more recently becoming automated (i.e. a biometric) due to advancements in computing capabilities. Fingerprint identification is popular because of the inherent ease in acquisition, the numerous sources (ten fingers) available for collection, and there established use and collections by law enforcement and immigration. Fingerprints have an uneven surface of ridges and valleys that form a unique pattern for each individual. For most applications primary interest is in the ridges pattern on the top joint of the finger.

1.1.4 Palm Print Recognition

Palm print recognition inherently implements many of the same matching characteristics that have allowed finger print recognition to be well reputed. Similar to finger, palm biometrics are represented by the information presented in the fiction ridge impression. This information combines ridge flow, ridge characteristics, and ridge structure of the raised portion of the epidermis, The data represented by these fiction ridge impressions allow a determination that corresponding area of fiction ridge impression either originated by the same source or could not have been made by the same source, As finger prints, palm prints also have uniqueness and permanence. However palm recognition has been slower in becoming automated due to some restraints in computing capabilities and live-scan technologies.

1.1.5 Hand/Finger Geometry

One of the first successful commercial biometric products was a hand geometry system. The systems are widely implemented for their ease of use, public acceptance and integration capabilities. One of the shortcomings of the hand geometry characteristics is that it is not highly unique. Hence limiting the application of hand geometry. Typically a user enters Personal Identification Number, PIN code to claim an identity, and then places his/her hand on the system, which takes picture of the hand, using mirrors the picture shows the view of the hand from the top and side. Measurements are then taken on the digits of hand and compared to those collected at enrollment.

1.1.6 Speaker Recognition

Uses an individual speech, a feature influenced by both the physical structure of an individual’s vocal tract and the behavioral characteristics of an individual for recognition purposes.

1.1.7 Dynamic Signatures

Dynamic signatures measure the speed and pressure one uses when signing his or her name (not what the signatures look like).

1.1.8 Key strokes Dynamics

Measures the typing pattern of an individual

1.1.9 Gait / Body Recognition

It measures how someone appears as he or she walks. As in face recognition, this technique is one that humans intuitively use to recognize someone.

1.1.10 Facial Thermographs

It measures how heat dissipates off the face of an individual.

1.2 Iris Recognition

The iris is an externally visible, yet protected organ whose unique epigenetic pattern remains stable throughout adult life [2]. These characteristics make it very attractive for use as a biometric for identifying individuals. Image processing techniques can be employed to extract the unique iris pattern from a digitized image of the eye, and encode it into a biometric template, which can be stored in a database. This biometric template contains an objective mathematical representation of the unique information stored in the iris, and allows comparisons to be made between templates. When a subject wishes to be identified by iris recognition system, their eye is first photographed, and then a template created for their iris region. This template is then compared with the other templates stored in a database until either a matching template is found and the subject is identified, or no match is found and the subject remains unidentified.

Although prototype systems had been proposed earlier, it was not until the early nineties that Cambridge researcher, John Daugman, implemented a working automated iris recognition system. The Daugman system is patented and the rights are now owned by the company Iridian Technologies. Even though the Daugman system is the most successful and most well known, many other systems have been developed. The most notable include the systems of Wildes et al., Boles and Boashash, Lim et al., and Noh et al. [10]. The algorithms by Lim et al. are used in the iris recognition system developed by the Evermedia and Senex companies. Also, the Noh et al. algorithm is used in the 'IRIS2000' system, sold by IriTech. These are, apart from the Daugman system, the only other known commercial implementations.

The Daugman system has been tested under numerous studies, all reporting a zero failure rate. The Daugman system is claimed to be able to perfectly identify an individual, given millions of possibilities. The prototype system by Wildes et al. also reports flawless performance with 520 iris images, and the Lim et al. system attains a recognition rate of 98.4% with a database of around 6,000 eye images.

Compared with other biometric technologies, such as face, speech and finger recognition, iris recognition can easily be considered as the most reliable form of biometric technology. However, there have been no independent trials of the technology, and source code for systems is not available. Also, there is a lack of publicly available datasets for testing and research, and the test results published have usually been produced using carefully imaged irises under favorable conditions.

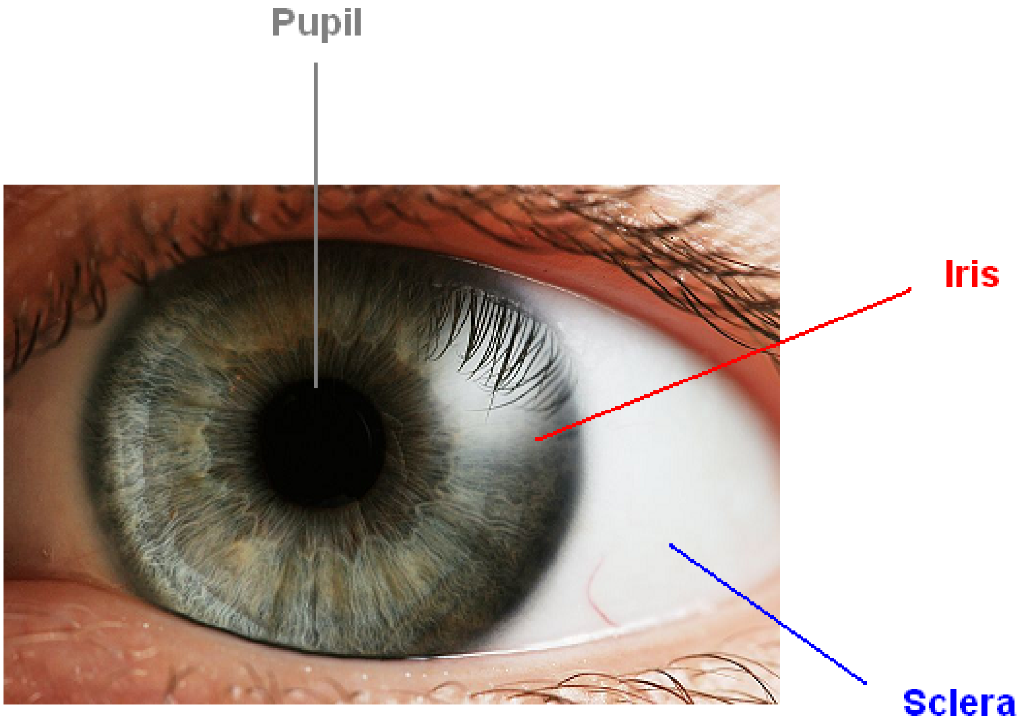


Figure 1.2: The human iris

The iris is a thin circular diaphragm, which lies between the cornea and the lens of the human eye. A front-on view of the iris is shown in Figure 1.1. The iris is perforated close to its centre by a circular aperture known as the pupil. The function of the iris is to control the amount of light entering through the pupil, and this is done by the sphincter and the dilator muscles, which adjust the size of the pupil. The average diameter of the iris is 12 mm, and the pupil size can vary from 10% to 80% of the iris diameter.

The iris consists of a number of layers; the lowest is the epithelium layer, which contains dense pigmentation cells. The stromal layer lies above the epithelium layer, and contains blood vessels, pigment cells and the two iris muscles. The density of stromal pigmentation determines the color of the iris. The externally visible surface of the multi-layered iris contains two zones, which often differ in color. An outer ciliary zone and an inner pupillary zone, and these two zones are divided by the collarette – which appears as a zigzag pattern. Formation of the iris begins during the third month of embryonic life. The unique pattern on the surface of the iris is formed during the first year of life, and pigmentation of the stroma takes place for the first few years. Formation of the unique patterns of the iris is random and not related to any genetic factors. The only characteristic that is dependent on genetics is the pigmentation of the iris, which determines its color. Due to the epigenetic nature of iris patterns, the two eyes of an individual contain completely independent iris patterns, and identical twins possess uncorrelated iris patterns. For further details on the anatomy of the human eye consult the book by Wolff .

1.3 Why is Iris Recognition possible?

1.3.1 History

Efforts to devise reliable mechanical means for biometric personal identification have a long and colorful history. However, the idea of using iris patterns for personal identification was originally proposed in 1936 by ophthalmologist Frank Burch, MD. In the 1980's the idea appeared in James Bond movies, but it remained science fiction. It was not until 1987, two American ophthalmologists, Leonard Flom and Aran Safir patented Burch's concept but they were unable to develop such a process. So Instead they turned to John Daugman, who was teaching at Harvard University and now at Cambridge University, to develop actual algorithms for iris recognition. These algorithms, which Daugman developed in 1994, are the basis for all current iris recognition systems.

1.3.2 Properties of an Iris

The critical attributes for any biometrics are: the number of degree-of-freedom of variation in the chosen index across the human population, since this determines uniqueness; its immutability over time and its immunity to intervention; and the computational prospects for efficiently encoding and reliably recognizing the identifying pattern. In the entire human population, no two irises are alike in their mathematical detail, even among identical (monozygotic) twins. The probability that two irises could produce exactly the same Iris Code is approximately 1 in 10_{78}. (The population of the earth is around 10_{10} .)

The possibility that the iris of the eye might be used as a kind of optical fingerprint for personal identification was suggested originally by ophthalmologists , who noted from clinical experience that every iris had a highly detailed and unique texture, which remained unchanged in clinical photographs spanning decades. The iris is composed of elastic connective tissue, the trabecular meshwork, whose prenatal morphogenesis[1] is completed during the 8th month of gestation. It consists of pectinate ligaments adhering into a tangled mesh revealing striations, ciliary processes, crypts, rings, furrows, a corona, sometimes freckles, vasculature, and other features. During the first year of life a blanket of chromatophore cells usually changes the color of the iris, but the available clinical evidence indicates that the trabecular pattern itself is stable throughout the lifespan.

Properties that enhance its suitability for use in automatic identification include: its inherent isolation and protection from the external environment, being an internal organ of the eye, behind the cornea and the aqueous humor; the impossibility of surgically modifying it without unacceptable risk to vision and its physiological response to light, which provides a natural test against artifice.

A Property the iris shares with fingerprints is the random morphogenesis of its minutiae. Because there is no genetic penetrance in the expression of this organ beyond its anatomical form, physiology, color and general appearance, the iris texture itself is stochastic or possibly chaotic. Since its detailed morphogenesis depends on initial conditions in the embryonic mesoderm from which it develops, the phenotypic expression even of two irises with the same genetic genotype (as in identical twins, or the pair possessed by one individual) have uncorrelated minutiae. In these respects the uniqueness of every iris parallels the uniqueness of every fingerprints, common genotype or not. But the iris enjoys further practical advantages over fingerprints and other biometrics for purposes of automatic recognition, including the ease of registering its image at some distance from the Subject without physical contact unobtrusively and perhaps inconspicuously and its intrinsic polar geometry, which imparts a natural coordinate system and an origin of coordinates.

1.4 Objective

The objective will be to implement an iris recognition system using integro diffenrential operator and wavelets. The development tool used will be MATLAB®, and emphasis will be only on the software for performing recognition, and not hardware for capturing an eye image. A rapid application development (RAD) approach will be employed in order to produce results quickly. MATLAB® provides an excellent RAD environment, with its image processing toolbox, and high level programming methodology. The system is to be composed of a number of sub-systems, which correspond to each stage of iris recognition. These stages are segmentation – locating the iris region in an eye image, normalization – creating a dimensionally consistent representation of the iris region, and feature encoding – creating a template containing only the most discriminating features of the iris. The input to the system will be an eye image, and the output will be an iris template, which will provide a mathematical representation of the iris region.

Chapter 2: Segmentation

2.1 Introduction

The first stage of iris recognition is to isolate the actual iris region in a digital eye image. The iris region, shown in Figure 1.1, can be approximated by two circles, one for the iris/sclera boundary and another, interior to the first, for the iris/pupil boundary. The eyelids and eyelashes normally occlude the upper and lower parts of the iris region. Also, specular reflections can occur within the iris region corrupting the iris pattern. A technique is required to isolate and exclude these artifacts as well as locating the circular iris region.

The success of segmentation depends on the imaging quality of eye images. Images in the CASIA iris database do not contain specular reflections due to the use of near infra-red light for illumination. Also, persons with darkly pigmented irises will present very low contrast between the pupil and iris region if imaged under natural light, making segmentation more difficult. The segmentation stage is critical to the success of an iris recognition system, since data that is falsely represented as iris pattern data will corrupt the biometric templates generated, resulting in poor recognition rates.

2.1.1 Integro-differential Operator

In order to localize an iris, Daugman proposed the Integro-differential operator . The operator assumes that pupil and limbus are circular contours and performs as a circular edge detector. Detecting the upper and lower eyelids are also performed using the Integro-differential operator by adjusting the contour search from circular to a designed arcuate. The Integro-differential method is defined as

$$\max(r, x_0, y_0) \left| G_\sigma(r) * \frac{\partial}{\partial r} \int_{(r,x_0,y_0)} \frac{I(x,y)}{2\pi r} ds \right|. \tag{2.1}$$

The operator pixel-wise searches throughout the raw input image, $I(x; y)$, and obtains the blurred partial derivative of the integral over normalized circular contours in different radii. The pupil and limbus boundaries are expected to maximize the contour integral derivative, where the intensity values over the circular borders would make a sudden change.

G¾(r) is a smoothing function controlled by ¾ that smoothes the image intensity for a more precise search.

2.1.2 Hough Transform

The Hough transform is a standard computer vision algorithm that can be used to determine the parameters of simple geometric objects, such as lines and circles, present in an image. The circular Hough transform can be employed to deduce the radius and centre coordinates of the pupil and iris regions. An automatic segmentation algorithm based on the circular Hough transform is employed by Wildes et al., Kong and Zhang, Tisse et al. , and Ma et al. . Firstly, an edge map is generated by calculating the first derivatives of intensity values in an eye image and then thresholding the result. From the edge map, votes are cast in Hough space for the parameters of circles passing through each edge point. These parameters are the centre coordinates xc and yc, and the radius r, which are able to define any circle according to the equation

$$x_c^2 + y_c^2 - r^2 = 0$$

A maximum point in the Hough space will correspond to the radius and centre coordinates of the circle best defined by the edge points. Wildes et al. and Kong and

Zhang also make use of the parabolic Hough transform to detect the eyelids, approximating the upper and lower eyelids with parabolic arcs, which are represented as;

$$(-(x-h_j)\sin\theta_j+(y-k_j)\cos\theta_j)^2=a_j((x-h_j)\cos\theta_j+(y-k_j)\sin\theta_j)$$

In performing the preceding edge detection step, Wildes et al. bias the derivatives in the horizontal direction for detecting the eyelids, and in the vertical direction for detecting the outer circular boundary of the iris, this is illustrated in eq 2.1. The motivation for this is that the eyelids are usually horizontally aligned, and also the eyelid edge map will corrupt the circular iris boundary edge map if using all gradient data. Taking only the vertical gradients for locating the iris boundary will reduce influence of the eyelids when performing circular Hough transform, and not all of the edge pixels defining the circle are required for successful localization. Not only does this make circle localization more accurate, it also makes it more efficient, since there are less edge points to cast votes in the Hough space.

There are a number of problems with the Hough transform method. First of all, it requires threshold values to be chosen for edge detection, and this may result in critical edge points being removed, resulting in failure to detect circles/arcs. Secondly, the Hough transform is computationally intensive due to its 'brute-force' approach, and thus may not be suitable for real time applications.

2.1.3 Active Contour Models

Ritter et al. make use of active contour models for localizing the pupil in eye images. Active contours respond to pre-set internal and external forces by deforming internally or moving across an image until equilibrium is reached. The contour contains a number of vertices, whose positions are changed by two opposing forces, an internal force, which is dependent on the desired characteristics, and an external force, which is dependent on the image. Each vertex is moved between time t and t + 1 by

$$v_i(t+1)=v_i(t)+F_i(t)+G_i(t)$$

where Fi is the internal force, Gi is the external force and vi is the position of vertex i.
For localization of the pupil region, the internal forces are calibrated so that the contour forms a globally expanding discrete circle. The external forces are usually found using the edge information. In order to improve accuracy Ritter et al. use the variance image, rather than the edge image.
A point interior to the pupil is located from a variance image and then a discrete circular active contour (DCAC) is created with this point as its centre. The DCAC is then moved under the influence of internal and external forces until it reaches equilibrium, and the pupil is localized.

2.1.4 Other Segmentation Methods

Other researchers use methods similar to the described segmentation methods. For instance, the iris localization proposed by Tisse et al. is a combination of the Integro-differential and the Hough transform. The Hough transform is used for a quick guess of the pupil center and then the Integro-differential is used to accurately locate pupil and limbus using a smaller search space.
Lim et al. localize pupil and limbus by providing an edge map of the intensity values of the image. The center of pupil is then chosen using a bisection method that passes perpendicular lines from every two points on the edge map. The center point is then obtained by voting the point that has the largest number of line crossovers. The pupil and limbus boundaries are then selected by increasing the radius of a virtual circle with the selected center point and choosing the two radii that have the maximum number of edge crosses by the virtual circle as the pupil and limbus radii.

2.2 Histogram

The histogram in the context of image processing is the operation by which the occurrences of each intensity value in the image is shown. Normally, the histogram is a graph showing the number of pixels in an image at each different intensity value found in that image. For an 8-bit grayscale image there are 256 different possible intensities, and so the histogram will graphically display 256 numbers showing the distribution of pixels amongst those grayscale values

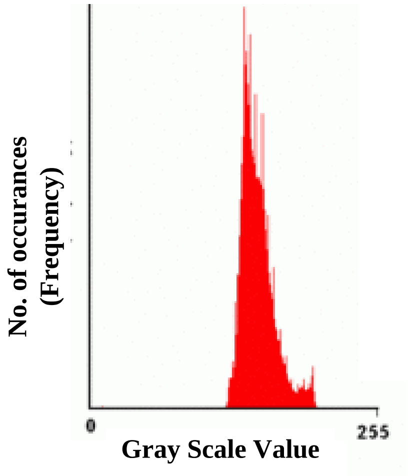


Figure 2.1: Image Histogram

2.2.1 What is Histogram Equalization

Histogram equalization is the technique by which the dynamic range of the histogram of an image is increased. Histogram equalization assigns the intensity values of pixels in the input image such that the output image contains a uniform distribution of intensities. It improves contrast and the goal of histogram equalization is to obtain a uniform histogram. This technique can be used on a whole image or just on a part of an image.

Histogram equalization redistributes intensity distributions. If the histogram of any image has many peaks and valleys, it will still have peaks and valley after equalization, but peaks and valley will be shifted. Because of this, "spreading" is a better term than "flattening" to describe histogram equalization. In histogram equalization, each pixel is assigned a new intensity value based on its previous intensity level.

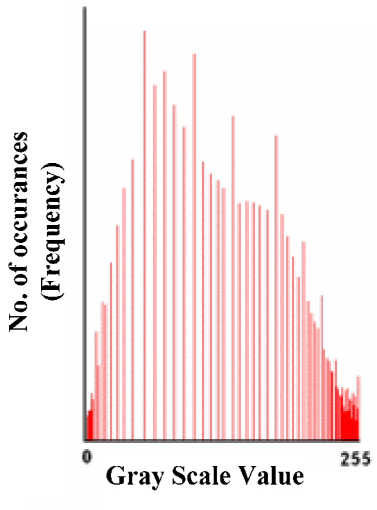


Figure 2.2: Image Equalized Histogram

2.2.2 General Working

The histogram equalization is operated on an image in three steps:

1). Histogram Formation

2). New Intensity Values calculation for each Intensity Levels

3). Replace the previous Intensity values with the new intensity values

$$Q_i = \sum_{j=0}^{i} N_j \quad \genfrac{}{}{0pt}{}{\text{Max. Intensity Level}}{\text{No. Of Pixels}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \text{(Eq. 2.2)}$$

Consider a discrete grayscale image, and let n_i be the number of occurrences of gray level i. The probability of an occurrence of a pixel of level i in the image is

$$P\left(x_i\right)=\frac{n_i}{n}, i \in 0,\ldots,L-1 \quad \text{……………….. (Eq. 2.3)}$$

L being the total number of gray levels in the image, *n* being the total number of pixels in the image, and *p* being in fact the image's histogram, normalized to 0..1.

Let us also define *c* as the cumulative probability function corresponding to *p*, defined by:

$$c(i)=\sum_{j=0}^{i} p(x_j) \quad \text{……………….. (Eq. 2.4)}$$

c is the image's accumulated normalized histogram.

The method is useful in images with backgrounds and foregrounds that are both bright or both dark. In particular, the method can lead to better views of bone structure in x-ray images, and to better detail in photographs that are over or under-exposed. A key advantage of the method is that it is a fairly straightforward technique and an invertible operator. If the histogram equalization function is known, then the original histogram can be recovered. The calculation is not computationally intensive.

This technique is used in image comparison processes (because it is effective in detail enhancement) and in the correction of non-linear effects introduced by, say a digitizer or display system.

2.3 Mean Filter

Mean filtering is a simple, intuitive and easy to implement method of *smoothing* images, *i.e.* reducing the amount of intensity variation between one pixel and the next. It is often used to reduce noise in images.

Discrete-time systems if modeled well can take a given input and process it to generate a desired set of output sequences. A good example of a discrete-time system is something called a moving average system, also known as a FIR averaging filter [19]. The goal of a moving average system is to smooth irregularities and random variations in a data set or signal. Figure 2.3 shows a situation where a moving average filter can help remove noise (agn[n]) that has been added to the desired signal (sig[n]).

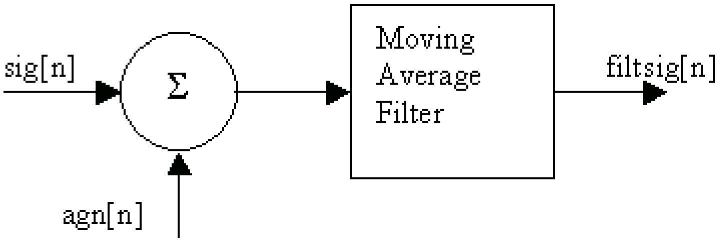


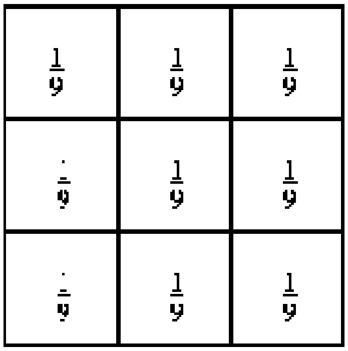
Figure 2.3: **Corrupted Signal through a Moving Average Filter**

A moving average filter can be modeled mathematically as shown in Equation (2.5). Looking at Equation (2.5), you can tell that the upper bound M in the summation can be any number of values. However, the value you select determines the order of the filter, which is the number of input samples you want your filter to take in for averaging

$$\mathbf{y}[\boldsymbol{n}]= \frac{\mathbf{1}}{\mathrm{M}} \sum_{k=0}^{M-1} \boldsymbol{x}\left[\boldsymbol{n}-\boldsymbol{k}\right] \quad \ldots\ldots\ldots\ldots\ldots\ldots.. \text{(Eq. 2.5)}$$

2.3.1 How It Works

The idea of mean filtering is simply to replace each pixel value in an image with the mean (`average') value of its neighbors, including itself. This has the effect of eliminating pixel values, which are unrepresentative of their surroundings. Mean filtering is usually thought of as a convolution filter. Like other convolutions it is based around a kernel, which represents the shape and size of the neighborhood to be sampled when calculating the mean. Often a 3×3 square kernel is used, as shown in Figure, although larger kernels (*e.g.* 5×5 squares) can be used for more severe smoothing. (Note that a small kernel can be applied more than once in order to produce a similar - but not identical - effect as a single pass with a large kernel.)



$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Figure 2.4: 3×3 averaging kernel often used in mean filtering

Computing the straightforward convolution of an image with this kernel carries out the mean filtering process. Mean filtering is most commonly used as a simple method for reducing noise in an image. In my research technique I have smoothened/average filtered the images using 5 * 5 kernel to achieve more fine results.

2.4 Interpolation

In the mathematical subfield of numerical analysis, **interpolation** is a method of constructing new data points from a discrete set of known data points.

In engineering and science one often has a number of data points, as obtained by sampling or experiment, and tries to construct a function which closely fits those data points. This is called curve fitting. Interpolation is a specific case of curve fitting, in which the function must go exactly through the data points [20].

Given a sequence of *n distinct* numbers x_k called **nodes** and for each x_k a second number y_k, we are looking for a function f so that

$$f(x_k) = y_k, k = 1, \ldots., n \quad \ldots\ldots\ldots\ldots\ldots\ldots.. \text{(Eq. 2.6)}$$

A pair x_k,y_k is called a **data point** and f is called the **interpolant** for the data points.

When the numbers y_k are given by a known function, we sometimes write f_k.

A different problem which is closely related to interpolation is the approximation of a complicated function by a simple function. Suppose we know the function but it is too complex to evaluate efficiently. Then we could pick a few known data points from the complicated function, creating a lookup table, and try to interpolate those data points to construct a simpler function. Of course when using the simple function to calculate new data points we usually do not receive the same result as when using the original function, but depending on the problem domain and the interpolation method used the gain in simplicity might offset the error. Output pixel values are calculated from a weighted average of pixels in the nearest 4-by-4 neighborhood

2.4.1 Bicubic Interpolation

In numerical analysis, a branch of mathematics, bicubic interpolation is one of the most common interpolation methods in two dimensions. Bicubic Interpolation is mainly used for image resizing process [20]. With this method, the value f(x, y) of a function f at a point (x, y) is computed as a weighted average of the nearest sixteen pixels in a rectangular grid (a 4x4 array). Here, two cubic interpolation polynomials, one for each plane direction, are used. The bicubic interpolation is calculated as follows:

$$a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2 + a_{21}x^2y + a_{12}xy^2 + a_{22}x^2y^2 + a_{30}x^3 + a_{03}y^3 + a_{31}x^3y + a_{13}xy^3 + a_{32}x^3y^2 + a_{23}x^{2} \quad \text{………………..} \quad (\text{Eq. } 2.7)$$

Or, in more compact form:

$$\sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}x^i y^j \quad \text{………………..} \quad (\text{Eq. } 2.8)$$

Bicubic interpolation is more sophisticated and produces smoother edges than bilinear interpolation. Here, a new pixel is a bicubic function using 16 pixels in the nearest 4 x 4 neighborhood of the pixel in the original image. This is the method most commonly used by image editing, in many digital cameras for resampling images.

2.5 Implementation

The following step are performed in image preprocessing

1. Iris Localization
2. Iris Normalization
3. Contrast Enhancement

1. Iris Localization: First stage of iris recognition is to isolate the iris region. It can be approximated by two circles. One is for iris-pupil and second one is iris-sclera. First of all the pupil is extracted along with its centre coordinates. Following steps are performed for this purpose:
   - First of all the sobel filter is applied on the input image to detect the edges.
   - Then thresholding is applied to make the pupil appear alone.
   - Median filter is applied to remove salt and pepper noise.
   - Find the summation of black pixels in each row
   - Find the first row x1 that has the maximum no of black pixels.
   - Find the last row x2 that has the maximum no of black pixels.
   - Get the x-component of pupil center coordinates using Xc=(x1+x2)/2
   - Apply same procedure on the columns to get y-component of pupil center.
   - Get the pupil radius by subtracting first black row of pupil from the pupil center.

   Following steps are performed to calculate the iris.

   - Gaussian low-pass smoothing filter is applied for smoothing the image which attenuates the effect of high level frequencies i.e. to reduce the effect of sharp reflections etc.

- A Circle is used as model for outer boundary that has the radius greater than the radius of the pupil.
- Twenty five small increments are used in diameter of the circle.
- The outer boundary is detected as the circle on the image for which there will be sudden change in luminance summed around its perimeter.
- Detect the iris boundary by finding the ellipse that maximizes the derivative.

2.6 Results

The automatic segmentation model proved to be successful. The CASIA database provided good segmentation, since those eye images had been taken specifically for iris recognition research and boundaries of iris pupil and sclera were clearly distinguished.

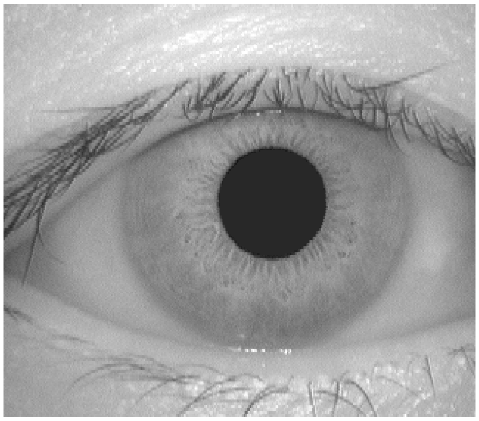


Figure 2.5: An Iris Image

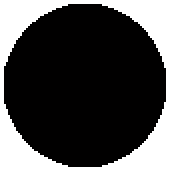


Figure 2.6: Output of intermediate step

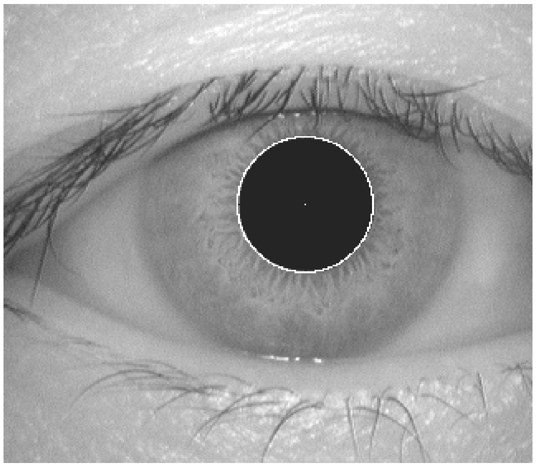


Figure 2.7:Output of the pupil localization step

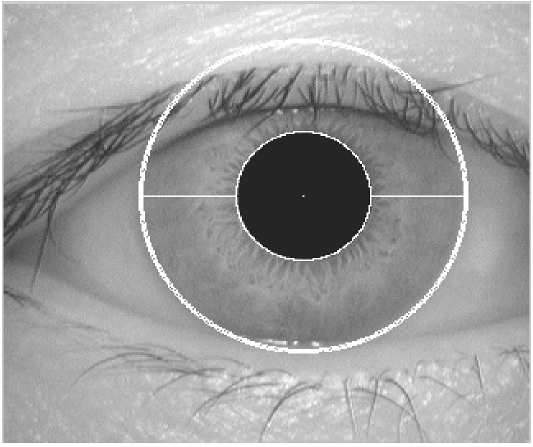


Figure 2.8: Output of iris localization step

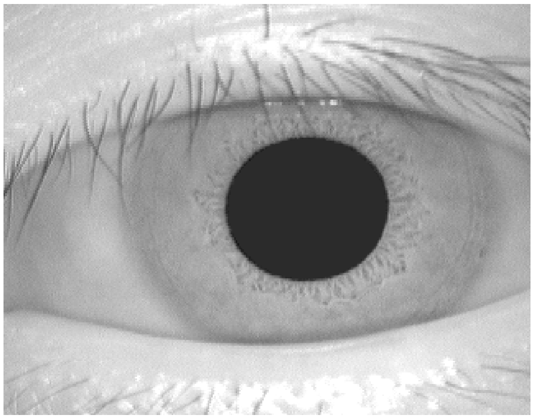


Figure 2.9: An Iris Image

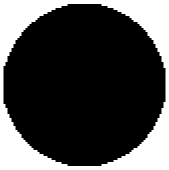


Figure 2.10: Output of intermediate step

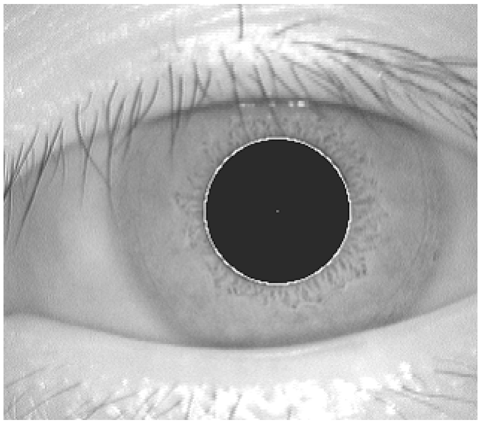


Figure 2.11: Output of the pupil localization step

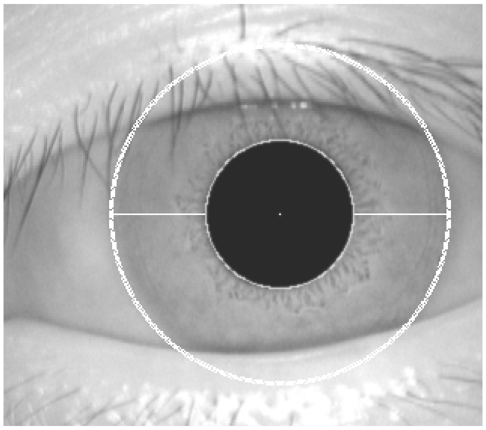


Figure 2.12: Output of iris localization step

Chapter 3: Normalization

3.1 Introduction

Once the iris region is successfully segmented from an eye image, the next stage is to transform the iris region so that it has fixed dimensions in order to allow comparisons. The dimensional inconsistencies between eye images are mainly due to the stretching of the iris caused by pupil dilation from varying levels of illumination. Other sources of inconsistency include, varying imaging distance, rotation of the camera, head tilt, and rotation of the eye within the eye socket. The normalization process will produce iris regions, which have the same constant dimensions, so that two photographs of the same iris under different conditions will have characteristic features at the same spatial location.

Another point of note is that the pupil region is not always concentric within the iris region, and is usually slightly nasal [2]. This must be taken into account if trying to normalize the ‘doughnut’ shaped iris region to have constant radius.

3.1.1 Daugman’s Rubber Sheet Model

The homogenous rubber sheet model devised by Daugman [1] remaps each point within the iris region to a pair of polar coordinates (r,θ) where r is on the interval [0,1] and θ is angle [0,2π].

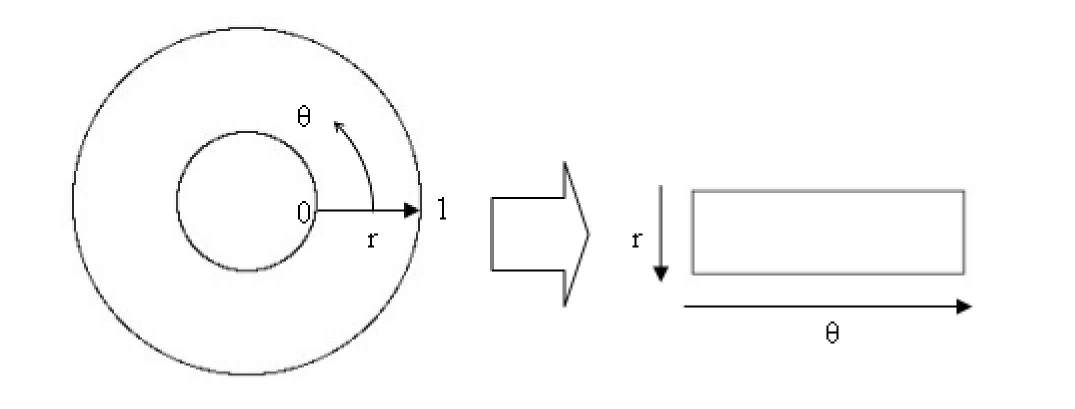


Figure 3.1: Daugman’s rubber sheet model

The remapping of the iris region from *(x,y)* Cartesian coordinates to the normalized non-concentric polar representation is modeled as

$$x(\rho,\theta) = (1-\rho) \times x_p(\theta) + \rho \times x_i(\theta),$$
$$y(\rho,\theta) = (1-\rho) \times y_p(\theta) + \rho \times y_i(\theta),$$

where

$$x_p(\theta) = x_{p0}(\theta) + r_p \times \cos(\theta),$$
$$y_p(\theta) = y_{p0}(\theta) + r_p \times \sin(\theta),$$
$$x_i(\theta) = x_{i0}(\theta) + r_i \times \cos(\theta),$$
$$y_i(\theta) = y_{i0}(\theta) + r_i \times \sin(\theta).$$

The process is inherently dimensionless in the angular direction. In the radial direction, the texture is assumed to change linearly, which is known as the rubber sheet model. The rubber sheet model linearly maps the iris texture in the radial direction from pupil border to limbus border into the interval [0 1] and creates a dimensionless transformation in the radial direction as well.

3.1.2 Image Registration

The Wildes et al. system employs an image registration technique, which geometrically warps a newly acquired image, into alignment with a selected database image [4]. When choosing a mapping function to transform the original coordinates, the image intensity values of the new image are made to be close to those of corresponding points in the reference image. The mapping function must be chosen so as to minimize

$$\int_x \int_y (I_d(x,y) - I_a(x-u, y-v))^2 \, dxdy$$

while being constrained to capture a similarity transformation of image coordinates (x,y) to (x',y'), that is

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} - sR(\phi) \begin{pmatrix} x \\ y \end{pmatrix}$$

with s a scaling factor and)(φR a matrix representing rotation by φ. In implementation, given a pair of iris images I_a and I_d, the warping parameters s and φ are recovered via an iterative minimization procedure [4].

3.1.3 Virtual Circles

In the Boles [8] system, iris images are first scaled to have constant diameter so that when comparing two images, one is considered as the reference image. This works differently to the other techniques, since normalization is not performed until attempting to match two iris regions, rather than performing normalization and saving the result for later comparisons. Once the two irises have the same dimensions, features are extracted from the iris region by storing the intensity values along virtual concentric circles, with origin at the centre of the pupil. A normalization resolution is selected, so that the number of data points extracted from each iris is the same. This is essentially the same as Daugman's rubber sheet model; however scaling is at match time, and is relative to the comparing iris region, rather than scaling to some constant dimensions. Also, it is not mentioned by Boles, how rotational invariance is obtained.

3.1.4 Non-linear Normalization Model

The unwrapping method proposed by Daugman assumes that iris patterns are linearly distributed in the radial direction, which allows the mapping procedure into the interval [0 1]. The technique relies on two main factors:

1. The image acquisition process adjusts the pupil size to a proper radius range by adjusting the illumination.
2. The feature extraction process is locally applied to many different positions of the iris texture, which would compensate the local nonlinear variations.

The proposed non-linear normalization method proposed by Yuan and Shi [33], considers a nonlinear behavior of iris patterns due to changes of pupil size. In order to unwrap an iris region properly, a non-linear model and a linear normalization model are combined. The non-linear method, which is first applied to an iris image, is based on three assumptions:

1. The pupil margin and iris root (which correspond to the inner and outer boundaries

of the iris) are concentric circles.

2. The margin of the pupil does not rotate significantly during pupil size changes.

3. The pupil shape does not change and remain circular when pupil size changes.

The non-linear model is defined by virtual arcs, which are named "fibers" following Wyatts work, that connect a point on the pupil border to a point on the limbus. The polar angle traversed by the arcs between these two points is ¦=2. The virtual arcs are defined based on normalized pupil sizes to a fixed value using a pre-defined ¸$_{ref}$, which is obtained by the mean of all ¸ values defined as ¸ = r=R in the iris database. The r and R represent the radius of pupil and limbus respectively. The reference annular zone with ¸$_{ref}$ is then linearly mapped into a fixed-size rectangle zone of m£ n by equally sampling m points in each virtual concentric sampling circle with a fixed radial resolution.

It is concluded by the authors of the presented approach that the non-linear model still simplifies the real physiological mechanism of iris deformation and some more assumptions and approximations are required to support the model. The model is also believed to explicitly show the non-linear behavior of iris textures due to the improvements obtained in the experiments.

3.1.5 Other Normalization Methods

Lim et al. uses a method very similar to the pseudo polar transform of Daugman. In this method, after finding the center of pupil and the inner and outer boundaries of iris, the texture is transformed into polar coordinates with a fixed resolution. In the radial direction, the texture is normalized from the inner boundary to the outer boundary into 60 pixels which is fixed throughout all iris images. The angular resolution is also fixed to a 0.8 degree over the 360 degree which produces 450 pixels in the angular direction.

Boles' normalization technique is also similar to Daugman's method with the difference that it is performed at the time of matching. The method is based on the diameter of the two matching irises. The ratio of the diameters are calculated and the diameter of irises are adjusted to have the same diameters. The number of samples is also fixed and it is set to a power-of-two integer in order to be suitable for the dyadic wavelet transform.

In addition, there has been some research on the pseudo polar transform in order to optimize its performance. The work presented by Joung et al. discusses the different possibilities of iris transformation. The research focuses on the fact that pupil and limbus are not always concentric and presents a method to improve the unwrapping process.

3.2 Implementation

The following steps are performed for the normalization process.

- The radial resolution is set to 64 and angular resolution is set to 256 pixels.
- For every pixel in the iris, an equivalent position is found out on polar axes. The normalized image was then interpolated into the size of the original image, by using the interp2 function.
- Only the lower half of the iris region is used.
- Upper region is usually covered by eye lashes.
- The lower half is enough for the further processing i.e. feature extraction and matching etc.
- The normalized iris region usually has low contrast.
- Improve the contrast by using histogram equalization.
- We divide the area between pi to 2 pi in 256 increasing equal parts to calculate the theta.
- Then we get the radial resolution by defining 64 rows.
- Then a single row with pre defined no of angular resolution is multiplied with radius of iris. So we get the radius of the iris in single row in each col.
- Then the pupil radius is subtracted from above step matrix.
- Then it is multiplied by the 64 rows and single col. Thus getting 64*256.Each row contains the same magnitude.
- Then the area of iris is divided into increasing steps between 0 and 1 having dim of 64*256.Then it is multiplied with the result obtained from step 5.hence we get the correct magnitude of each row in iris area i.e. starting from zero,xx,yy,.....,r-1,radius of iris on last row.
- Then cos and sin matrix of theta is calculated.
- Then x=r cos theta and y = r sin theta is calculated to get the cartesian coordinates.

- Then these are mapped on the image by x0=xp + xo and y0=yp-y0 using spatial coordinates instead of pixel coords. In spatial coords x represent cols going to right and y represent rows going to bottom.
- After that interpolation is used to get the image values of x0,y0 as x0 and y0 are not exact whole coords but in fractions.

3.3 Results

The normalization process proved to be successful and some results are shown in

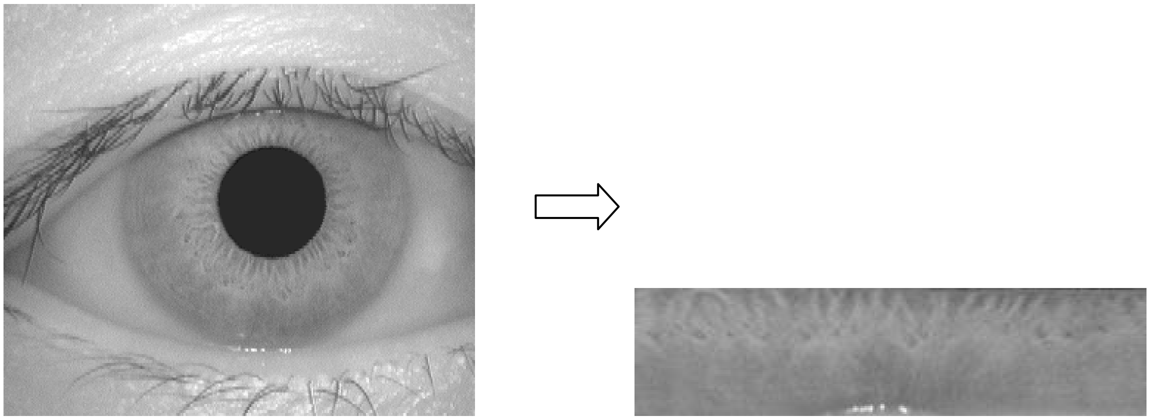


Figure 3.2: An Iris Image and normalized form

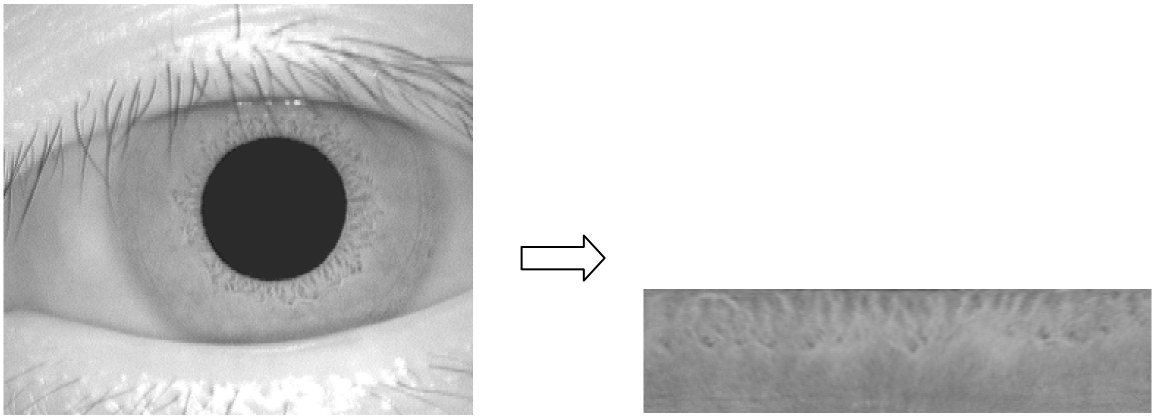


Figure 3.3: An Iris Image and normalized form

Chapter 4: Feature Encoding and Matching

4.1 Introduction

In order to provide accurate recognition of individuals, the most discriminating information present in an iris pattern must be extracted. Only the significant features of the iris must be encoded so that comparisons between templates can be made. Most iris recognition systems make use of a band pass decomposition of the iris image to create a biometric template.

The template that is generated in the feature encoding process will also need a corresponding matching metric, which gives a measure of similarity between two iris templates. This metric should give one range of values when comparing templates generated from the same eye, known as intra-class comparisons, and another range of values when comparing templates created from different irises, known as inter-class comparisons. These two cases should give distinct and separate values, so that a decision can be made with high confidence as to whether two templates are from the same iris, or from two different irises.

4.2 Wavelet

The term wavelet means a small wave. The smallness refers to the condition that this (window) function is of finite length (compactly supported). The wave refers to the condition that this function is oscillatory. The term mother implies that the functions with different region of support that are used in the transformation process are derived from one main function, or the mother wavelet. In other words, the mother wavelet is a prototype for generating the other window functions.

The term translation is used in the same sense as it was used in the STFT; it is related to the location of the window, as the window is shifted through the signal. This term, obviously, corresponds to time information in the transform domain. However, we do not have a frequency parameter, as we had before for the STFT. Instead, we have scale parameter which is defined as $1/frequency$. The term frequency is reserved for the STFT. Scale is described in more detail in the next section.

4.2.1 Scale

The parameter scale in the wavelet analysis is similar to the scale used in maps. As in the case of maps, high scales correspond to a non-detailed global view (of the signal), and low scales correspond to a detailed view. Similarly, in terms of frequency, low frequencies (high scales) correspond to a global information of a signal (that usually spans the entire signal), whereas high frequencies (low scales) correspond to a detailed information of a hidden pattern in the signal (that usually lasts a relatively short time). Cosine signals corresponding to various scales are given as examples in the following figure .

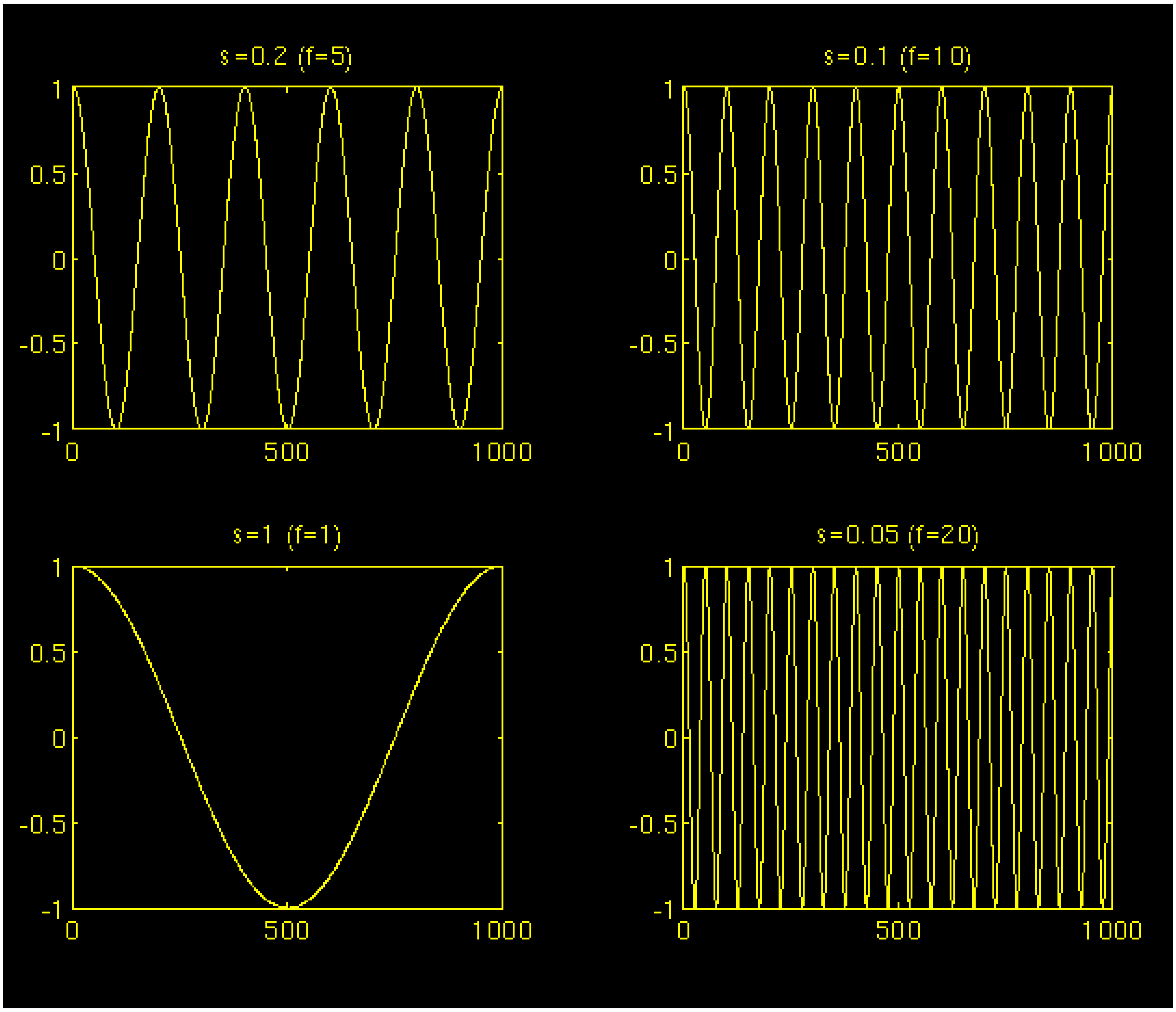


Figure 4.1: Cosine signals corresponding to various scales

Fortunately in practical applications, low scales (high frequencies) do not last for the entire duration of the signal, unlike those shown in the figure, but they usually appear from time to time as short bursts, or spikes. High scales (low frequencies) usually last for the entire duration of the signal.

Scaling, as a mathematical operation, either dilates or compresses a signal. Larger scales correspond to dilated (or stretched out) signals and small scales correspond to compressed signals. All of the signals given in the figure are derived from the same cosine signal, i.e., they are dilated or compressed versions of the same function. In the above figure, s=0.05 is the smallest scale, and s=1 is the largest scale.

In terms of mathematical functions, if f(t) is a given function f(st) corresponds to a contracted (compressed) version of f(t) if s > 1 and to an expanded (dilated) version of f(t) if s < 1 .

However, in the definition of the wavelet transform, the scaling term is used in the denominator, and therefore, the opposite of the above statements holds, i.e., scales s > 1 dilates the signals whereas scales s < 1 , compresses the signal.

4.2.2 Wavelet Encoding

Wavelets can be used to decompose the data in the iris region into components that appear at different resolutions. Wavelets have the advantage over traditional Fourier transform in that the frequency data is localised, allowing features which occur at the same position and resolution to be matched up. A number of wavelet filters, also called a bank of wavelets, is applied to the 2D iris region, one for each resolution with each wavelet a scaled version of some basis function. The output of applying the wavelets is then encoded in order to provide a compact and discriminating representation of the iris pattern.

4.2.3 Gabor Filters

Gabor filters are able to provide optimum conjoint representation of a signal in space and spatial frequency. A Gabor filter is constructed by modulating a sine/cosine wave with a Gaussian. This is able to provide the optimum conjoint localization in both space and frequency, since a sine wave is perfectly localized in frequency, but not localized in space. Modulation of the sine with a Gaussian provides localization in space, though with loss of localization in frequency. Decomposition of a signal is accomplished using a quadrature pair of Gabor filters, with a real part specified by a cosine modulated by a Gaussian, and an imaginary part specified by a sine modulated by a Gaussian. The real and imaginary filters are also known as the even symmetric and odd symmetric components respectively.

The centre frequency of the filter is specified by the frequency of the sine/cosine wave, and the bandwidth of the filter is specified by the width of the Gaussian.

Daugman makes uses of a 2D version of Gabor filters [1] in order to encode iris pattern data. A 2D Gabor filter over the an image domain *(x,y)* is represented as

$$G(x,y) = e^{-\pi[(x-x_0)^2/\alpha^2+(y-y_0)^2/\beta^2]} e^{-2\pi i[u_0(x-x_0)+v_0(y-y_0)]}$$

Where (xo,yo) specify position in the image, (α,β) specify the effective width and length, and (uo, vo) specify modulation, which has spatial frequency .The odd symmetric and even symmetric 2D Gabor filters are shown in Figure 4.2.

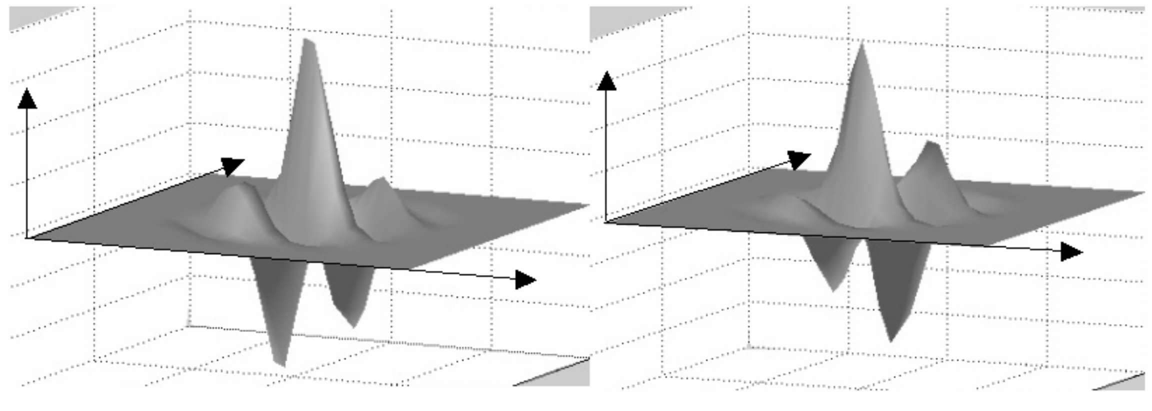


Figure 4.2: A quadrature pair of 2D Gabor filters left) real component or even symmetric filter characterised by a cosine modulated by a Gaussian right) imaginary component or odd symmetric filter characterised by a sine modulated by a Gaussian.

Daugman demodulates the output of the Gabor filters in order to compress the data. This is done by quantizing the phase information into four levels, for each possible quadrant in the complex plane. It has been shown by Oppenheim and Lim [23] that phase information, rather than amplitude information provides the most significant information within an image. Taking only the phase will allow encoding of discriminating information in the iris, while discarding redundant information such as illumination, which is represented by the amplitude component.

These four levels are represented using two bits of data, so each pixel in the normalized iris pattern corresponds to two bits of data in the iris template. A total of 2,048 bits are calculated for the template, and an equal number of masking bits are generated in order to mask out corrupted regions within the iris. This creates a compact 256-byte template, which allows for efficient storage and comparison of irises. The Daugman system makes use of polar coordinates for normalization, therefore in polar form the filters are given as

$$H(r,\theta) = e^{-i\omega(\theta-\theta_0)} e^{-(r-r_0)^2/\alpha^2} e^{-i(\theta-\theta_0)^2/\beta^2}$$

where *(α,β)* are the same as in Equation 4.1 and *(r_0, θ_0)* specify the centre frequency of the filter.

The demodulation and phase Quantisation process can be represented as

$$h_{\{\mathrm{Re},\mathrm{Im}\}} = \mathrm{sgn}_{\{\mathrm{Re},\mathrm{Im}\}} \int_\rho \int_\phi I(\rho,\phi) e^{-i\omega(\theta_0-\phi)} e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_0-\phi)^2/\beta^2} \rho d\rho d\phi$$

where h{Re, Im} can be regarded as a complex valued bit whose real and imaginary components are dependent on the sign of the 2D integral, and I(p,φ) is the raw iris image in a dimensionless polar coordinate system.

4.2.4 Log-Gabor Filters

A disadvantage of the Gabor filter is that the even symmetric filter will have a DC component whenever the bandwidth is larger than one octave [20]. However, zero DC component can be obtained for any bandwidth by using a Gabor filter which is Gaussian on a logarithmic scale, this is known as the Log-Gabor filter. The frequency response of a Log-Gabor filter is given as;

$$G(f) = \exp\left(\frac{-(\log(f/f_0))^2}{2(\log(\sigma/f_0))^2} \right)$$

where f_0 represents the centre frequency, and *σ* gives the bandwidth of the filter. Details of the Log-Gabor filter are examined by Field .

4.2.5 Zero-crossings of the 1D wavelet

Boles and Boashash [8] make use of 1D wavelets for encoding iris pattern data. The mother wavelet is defined as the second derivative of a smoothing function θ(x).

$$\psi(x) = \frac{d^2\theta(x)}{dx^2}$$

The zero crossings of dyadic scales of these filters are then used to encode features.
The wavelet transform of a signal *f(x)* at scale *s* and position *x* is given by

$$W_s f(x) = f * \left(s^2 \frac{d^2\theta(x)}{dx^2} \right)(x)$$
$$= s^2 \frac{d^2}{dx^2}(f * \theta_s)(x)$$

where

$$\theta_s = (1/s)\theta(x/s)$$

$W_s f(x)$ is proportional to the second derivative of *f(x)* smoothed by $\theta_s(x)$, and the zero crossings of the transform correspond to points of inflection in $f*\theta_s(x)$. The motivation for this technique is that zero-crossings correspond to significant features with the iris region.

4.2.6 Haar Wavelet

Lim et al. [9] also use the wavelet transform to extract features from the iris region. Both the Gabor transform and the Haar wavlet are considered as the mother wavelet. From multi-dimensionally filtering, a feature vector with 87 dimensions is computed. Since each dimension has a real value ranging from -1.0 to +1.0, the feature vector is sign quantised so that any positive value is represented by 1, and negative value as 0. This results in a compact biometric template consisting of only 87 bits.

Lim et al. compare the use of Gabor transform and Haar wavelet transform, and show that the recognition rate of Haar wavelet transform is slightly better than Gabor transform by 0.9%.

4.2.7 Laplacian of Gaussian Filters

In order to encode features, the Wildes et al. system decomposes the iris region by application of Laplacian of Gaussian filters to the iris region image. The filters are given as

$$\nabla G = -\frac{1}{\pi\sigma^4}\left(1-\frac{\rho^2}{2\sigma^2}\right)e^{-\rho^2/2\sigma^2}$$

where σ is the standard deviation of the Gaussian and ρ is the radial distance of a point from the centre of the filter.

The filtered image is represented as a Laplacian pyramid which is able to compress the data, so that only significant data remains. Details of Laplacian Pyramids are presented by Burt and Adelson [24]. A Laplacian pyramid is constructed with four different resolution levels in order to generate a compact iris template.

4.2.8 Hamming distance

The Hamming distance gives a measure of how many bits are the same between two bit patterns. Using the Hamming distance of two bit patterns, a decision can be made as to whether the two patterns were generated from different irises or from the same one.

In comparing the bit patterns *X* and *Y*, the Hamming distance, *HD*, is defined as the sum of disagreeing bits (sum of the exclusive-OR between *X* and *Y*) over *N*, the total number of bits in the bit pattern.

$$HD = \frac{1}{N}\sum_{j=1}^{N} X_j (XOR) Y_j$$

Since an individual iris region contains features with high degrees of freedom, each iris region will produce a bit-pattern which is independent to that produced by another iris, on the other hand, two iris codes produced from the same iris will be highly correlated.

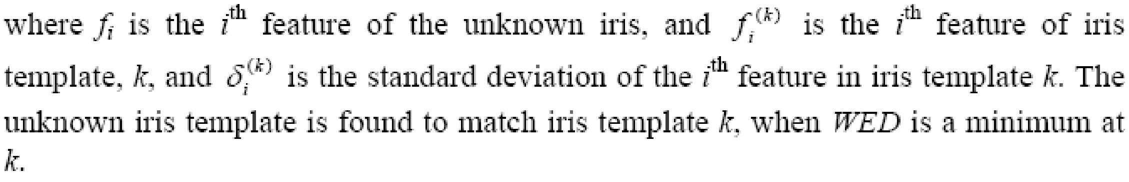
If two bits patterns are completely independent, such as iris templates generated from different irises, the Hamming distance between the two patterns should equal 0.5. This occurs because independence implies the two bit patterns will be totally random, so there is 0.5 chance of setting any bit to 1, and vice versa. Therefore, half of the bits will agree and half will disagree between the two patterns. If two patterns are derived from the same iris, the Hamming distance between them will be close to 0.0, since they are highly correlated and the bits should agree between the two iris codes.

The Hamming distance is the matching metric employed by Daugman, and calculation of the Hamming distance is taken only with bits that are generated from the actual iris region.

4.2.9 Weighted Euclidean Distance

The weighted Euclidean distance (WED) can be used to compare two templates, especially if the template is composed of integer values. The weighting Euclidean distance gives a measure of how similar a collection of values are between two templates. This metric is employed by Zhu et al. [11] and is specified as

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



where f_i is the i^{th} feature of the unknown iris, and $f_i^{(k)}$ 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. The unknown iris template is found to match iris template k, when WED is a minimum at k.

4.3 Implementation

Following steps are performed for feature extraction and matching

- Final process is generation of iris code.
- Most discriminating information in iris is extracted.
- The phase information in the pattern is used only because the phase angles are assigned regardless of the contrast in the image.
- Amplitude information is not used because it depends on extraneous factors
- Extraction of phase information is done using using 2D Gabor wavelets.
- Normalized iris image is divided into one-dimensional signals i.e. one for each row of the image.
- These signals are convolved with 1D Log Gabor Wavelets
- Using the output of Gabor convolve, the iris code is formed by assigning 2 elements for each pixel of the image.
- Each element contains a value 1 or 0 depending on the sign + or – of the real and imaginary part respectively.
- Comparison of the bit patterns generated is done to check if the two irises belong to the same person.
- Calculation of Hamming Distance (HD) is done for this comparison.
- Hamming Distance is a fractional measure of the number of bits disagreeing between two binary patterns.

$$HD = \frac{1}{N - \sum_{k=1}^{N} Xn_k (OR) Yn_k} \sum_{j=1}^{N} X_j (XOR) Y_j (AND) Xn'_j (AND) Yn'_j$$

Where, Xj and Yj are the two iriscodes, Xnj and Ynj are the corresponding noisy mask bits and N is the number of bits in each template.

- Take two feature templates as input.
- Convert them into the logical templates.
- Perform AND operation between the XOR result and noise mask.
- Count the no of bits that are 1 in the result of XOR operation.
- Divide the result by the total no of bits.
- If the result is less than a certain threshold then consider the two images as a match otherwise treat them as different.

4.4 Results

Feature Encoding and Matching process proved to be successful and some results are shown below

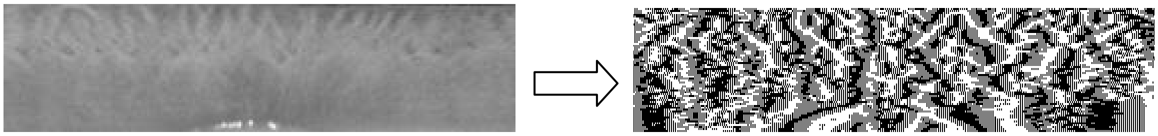


Figure 4.3: Normalized template and feature template

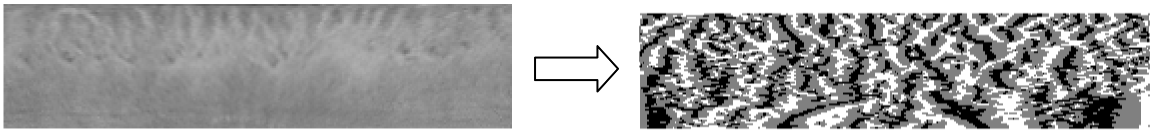


Figure 4.4: Normalized template and feature template

Chapter 5: Results/Comparisons & Discussion

In the previously explained chapters the procedure of implementation is discussed thoroughly. Now in this chapter the results of the technique are described. Database used is CASIA which is the most widely used iris database for research purposes in the world. For each eye, no of images are captured in two sessions, where some samples are collected in the first session and other in the second session.

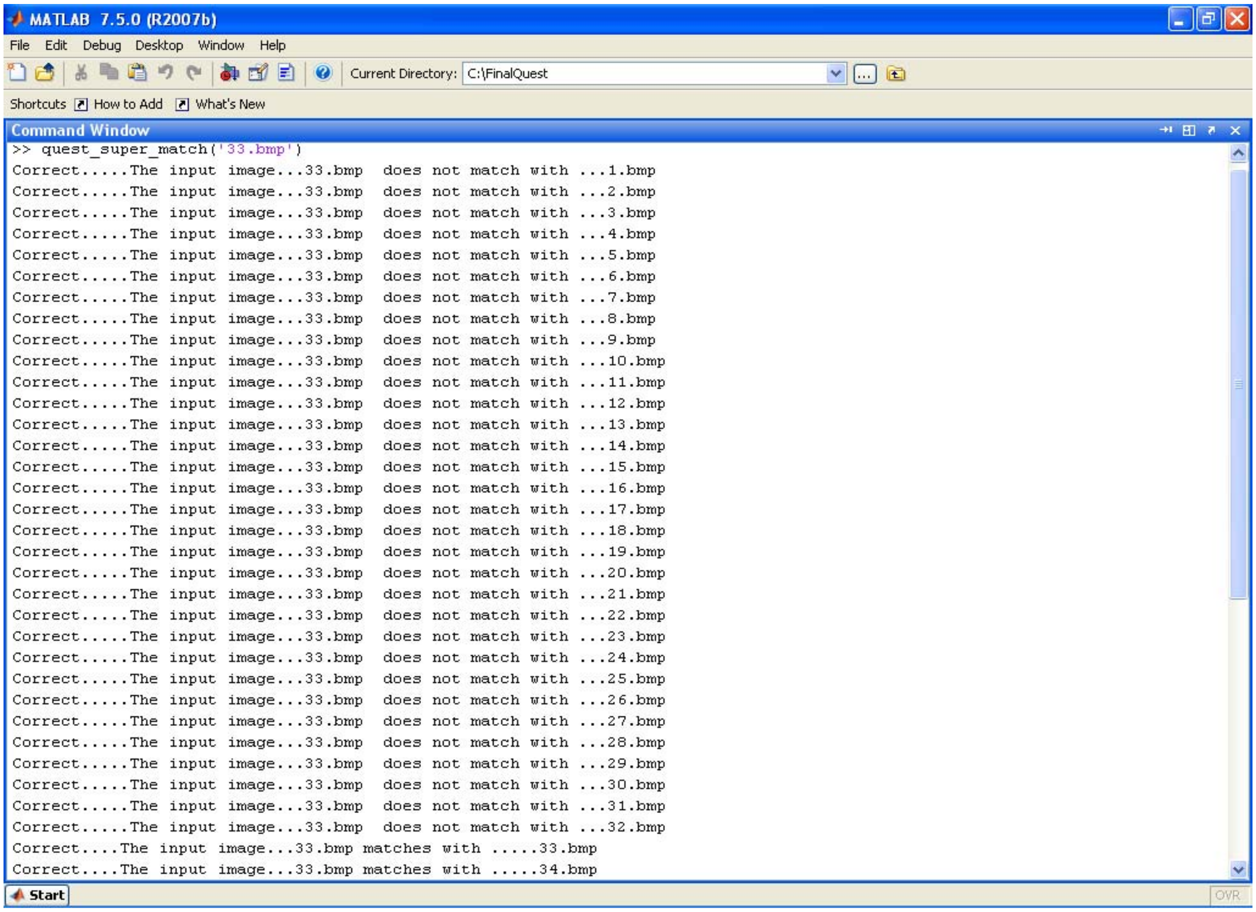
5.1 Test Case1:

Implemented algorithm takes 53 templates from 8 different eyes as input. Feature templates are generated from these templates and are stored in database. After testing I deduce the following results.

Number of Images	%age Recognition Testing images
50	**96.22**

Table 5.1: **Percentage Recognition for 50 iris templates**

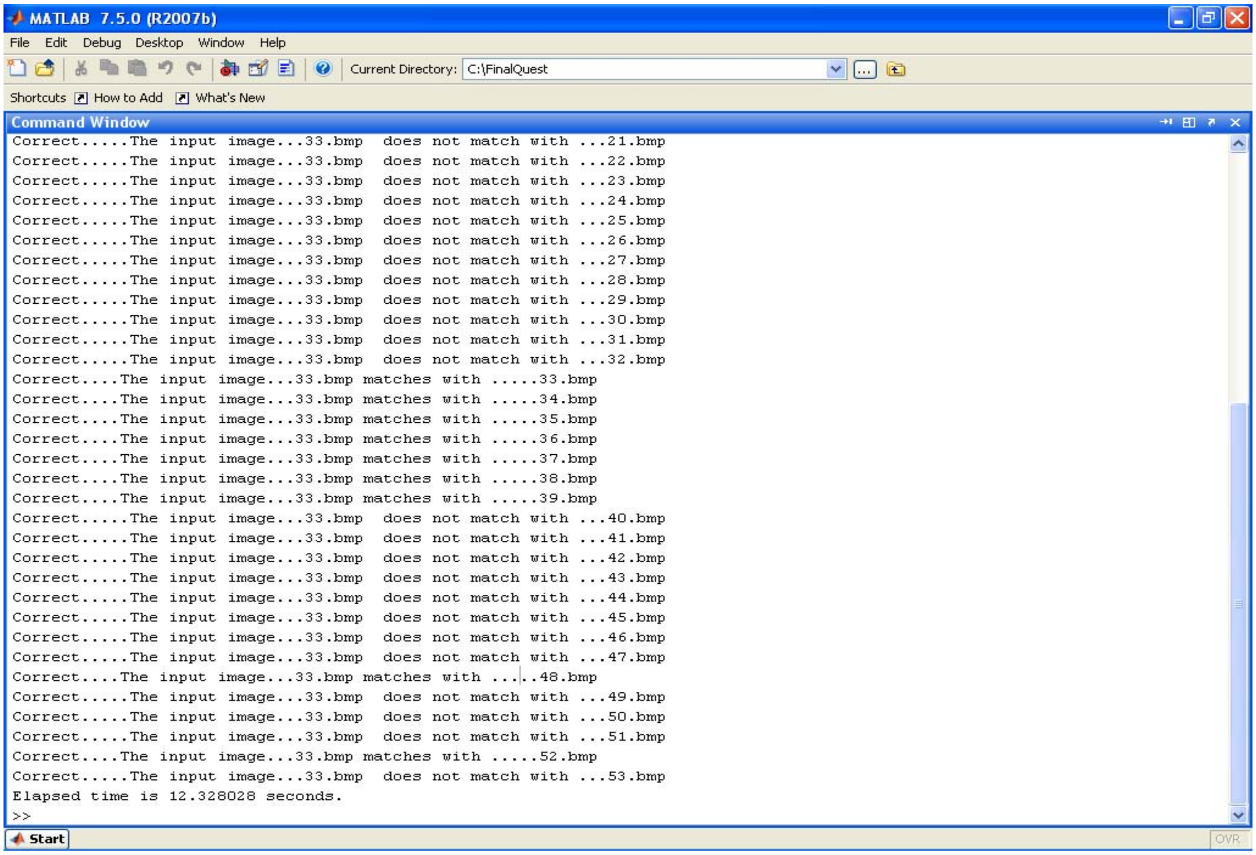
The following figure 5.1,5.2 displays the testing results in which the input image is compared with all the templates in the database one by one.


MATLAB 7.5.0 (R2007b)
File Edit Debug Desktop Window Help
Shortcuts How to Add What's New
Command Window
Start
OVR

Current Directory: C:\FinalQuest

```
>> quest_super_match('33.bmp')
Correct.....The input image...33.bmp  does not match with ...1.bmp
Correct.....The input image...33.bmp  does not match with ...2.bmp
Correct.....The input image...33.bmp  does not match with ...3.bmp
Correct.....The input image...33.bmp  does not match with ...4.bmp
Correct.....The input image...33.bmp  does not match with ...5.bmp
Correct.....The input image...33.bmp  does not match with ...6.bmp
Correct.....The input image...33.bmp  does not match with ...7.bmp
Correct.....The input image...33.bmp  does not match with ...8.bmp
Correct.....The input image...33.bmp  does not match with ...9.bmp
Correct.....The input image...33.bmp  does not match with ...10.bmp
Correct.....The input image...33.bmp  does not match with ...11.bmp
Correct.....The input image...33.bmp  does not match with ...12.bmp
Correct.....The input image...33.bmp  does not match with ...13.bmp
Correct.....The input image...33.bmp  does not match with ...14.bmp
Correct.....The input image...33.bmp  does not match with ...15.bmp
Correct.....The input image...33.bmp  does not match with ...16.bmp
Correct.....The input image...33.bmp  does not match with ...17.bmp
Correct.....The input image...33.bmp  does not match with ...18.bmp
Correct.....The input image...33.bmp  does not match with ...19.bmp
Correct.....The input image...33.bmp  does not match with ...20.bmp
Correct.....The input image...33.bmp  does not match with ...21.bmp
Correct.....The input image...33.bmp  does not match with ...22.bmp
Correct.....The input image...33.bmp  does not match with ...23.bmp
Correct.....The input image...33.bmp  does not match with ...24.bmp
Correct.....The input image...33.bmp  does not match with ...25.bmp
Correct.....The input image...33.bmp  does not match with ...26.bmp
Correct.....The input image...33.bmp  does not match with ...27.bmp
Correct.....The input image...33.bmp  does not match with ...28.bmp
Correct.....The input image...33.bmp  does not match with ...29.bmp
Correct.....The input image...33.bmp  does not match with ...30.bmp
Correct.....The input image...33.bmp  does not match with ...31.bmp
Correct.....The input image...33.bmp  does not match with ...32.bmp
Correct....The input image...33.bmp matches with .....33.bmp
Correct....The input image...33.bmp matches with .....34.bmp
```

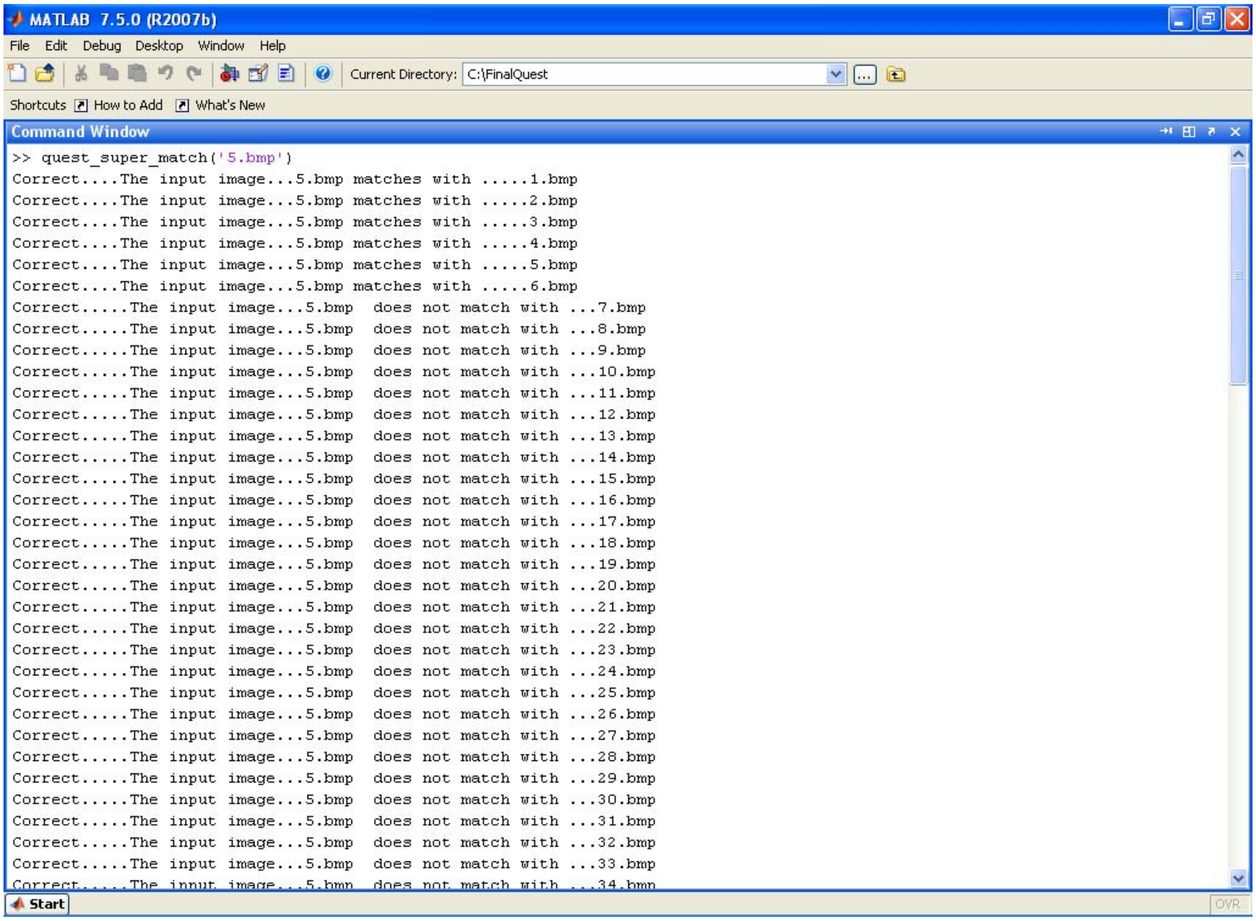
Figure 5.1: Testing of 50 subjects


MATLAB 7.5.0 (R2007b)
File Edit Debug Desktop Window Help
Current Directory: C:\FinalQuest
Shortcuts How to Add What's New
Command Window
Start
OVR

```
Correct.....The input image...33.bmp  does not match with ...21.bmp
Correct.....The input image...33.bmp  does not match with ...22.bmp
Correct.....The input image...33.bmp  does not match with ...23.bmp
Correct.....The input image...33.bmp  does not match with ...24.bmp
Correct.....The input image...33.bmp  does not match with ...25.bmp
Correct.....The input image...33.bmp  does not match with ...26.bmp
Correct.....The input image...33.bmp  does not match with ...27.bmp
Correct.....The input image...33.bmp  does not match with ...28.bmp
Correct.....The input image...33.bmp  does not match with ...29.bmp
Correct.....The input image...33.bmp  does not match with ...30.bmp
Correct.....The input image...33.bmp  does not match with ...31.bmp
Correct.....The input image...33.bmp  does not match with ...32.bmp
Correct....The input image...33.bmp matches with .....33.bmp
Correct....The input image...33.bmp matches with .....34.bmp
Correct....The input image...33.bmp matches with .....35.bmp
Correct....The input image...33.bmp matches with .....36.bmp
Correct....The input image...33.bmp matches with .....37.bmp
Correct....The input image...33.bmp matches with .....38.bmp
Correct....The input image...33.bmp matches with .....39.bmp
Correct.....The input image...33.bmp  does not match with ...40.bmp
Correct.....The input image...33.bmp  does not match with ...41.bmp
Correct.....The input image...33.bmp  does not match with ...42.bmp
Correct.....The input image...33.bmp  does not match with ...43.bmp
Correct.....The input image...33.bmp  does not match with ...44.bmp
Correct.....The input image...33.bmp  does not match with ...45.bmp
Correct.....The input image...33.bmp  does not match with ...46.bmp
Correct.....The input image...33.bmp  does not match with ...47.bmp
Correct....The input image...33.bmp matches with .....48.bmp
Correct.....The input image...33.bmp  does not match with ...49.bmp
Correct.....The input image...33.bmp  does not match with ...50.bmp
Correct.....The input image...33.bmp  does not match with ...51.bmp
Correct....The input image...33.bmp matches with .....52.bmp
Correct.....The input image...33.bmp  does not match with ...53.bmp
Elapsed time is 12.328028 seconds.
>>
```

Figure 5.2: Results of comparison with 50 subjects.

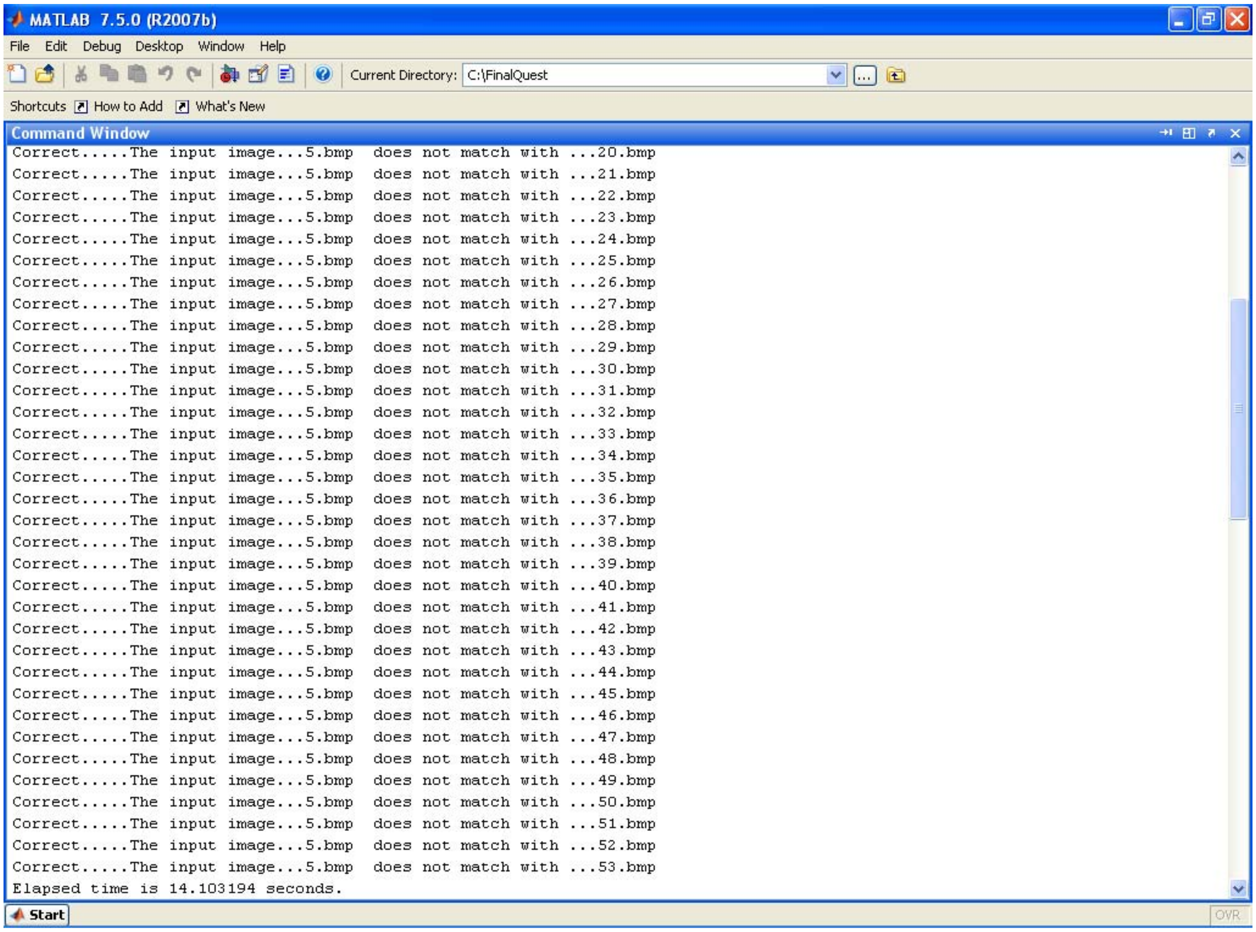
The following figure 5.3,5.4 displays the testing results of another input image that is compared with all the templates in the database one by one.



Current Directory: C:\FinalQuest

```
>> quest_super_match('5.bmp')
Correct....The input image...5.bmp matches with .....1.bmp
Correct....The input image...5.bmp matches with .....2.bmp
Correct....The input image...5.bmp matches with .....3.bmp
Correct....The input image...5.bmp matches with .....4.bmp
Correct....The input image...5.bmp matches with .....5.bmp
Correct....The input image...5.bmp matches with .....6.bmp
Correct.....The input image...5.bmp  does not match with ...7.bmp
Correct.....The input image...5.bmp  does not match with ...8.bmp
Correct.....The input image...5.bmp  does not match with ...9.bmp
Correct.....The input image...5.bmp  does not match with ...10.bmp
Correct.....The input image...5.bmp  does not match with ...11.bmp
Correct.....The input image...5.bmp  does not match with ...12.bmp
Correct.....The input image...5.bmp  does not match with ...13.bmp
Correct.....The input image...5.bmp  does not match with ...14.bmp
Correct.....The input image...5.bmp  does not match with ...15.bmp
Correct.....The input image...5.bmp  does not match with ...16.bmp
Correct.....The input image...5.bmp  does not match with ...17.bmp
Correct.....The input image...5.bmp  does not match with ...18.bmp
Correct.....The input image...5.bmp  does not match with ...19.bmp
Correct.....The input image...5.bmp  does not match with ...20.bmp
Correct.....The input image...5.bmp  does not match with ...21.bmp
Correct.....The input image...5.bmp  does not match with ...22.bmp
Correct.....The input image...5.bmp  does not match with ...23.bmp
Correct.....The input image...5.bmp  does not match with ...24.bmp
Correct.....The input image...5.bmp  does not match with ...25.bmp
Correct.....The input image...5.bmp  does not match with ...26.bmp
Correct.....The input image...5.bmp  does not match with ...27.bmp
Correct.....The input image...5.bmp  does not match with ...28.bmp
Correct.....The input image...5.bmp  does not match with ...29.bmp
Correct.....The input image...5.bmp  does not match with ...30.bmp
Correct.....The input image...5.bmp  does not match with ...31.bmp
Correct.....The input image...5.bmp  does not match with ...32.bmp
Correct.....The input image...5.bmp  does not match with ...33.bmp
Correct.....The input image...5.bmp  does not match with ...34.bmp
```

Figure 5.3: Results of testing of another image.


MATLAB 7.5.0 (R2007b)
File Edit Debug Desktop Window Help
Current Directory: C:\FinalQuest
Shortcuts How to Add What's New
Command Window
Start
OVR

```
Correct.....The input image...5.bmp  does not match with ...20.bmp
Correct.....The input image...5.bmp  does not match with ...21.bmp
Correct.....The input image...5.bmp  does not match with ...22.bmp
Correct.....The input image...5.bmp  does not match with ...23.bmp
Correct.....The input image...5.bmp  does not match with ...24.bmp
Correct.....The input image...5.bmp  does not match with ...25.bmp
Correct.....The input image...5.bmp  does not match with ...26.bmp
Correct.....The input image...5.bmp  does not match with ...27.bmp
Correct.....The input image...5.bmp  does not match with ...28.bmp
Correct.....The input image...5.bmp  does not match with ...29.bmp
Correct.....The input image...5.bmp  does not match with ...30.bmp
Correct.....The input image...5.bmp  does not match with ...31.bmp
Correct.....The input image...5.bmp  does not match with ...32.bmp
Correct.....The input image...5.bmp  does not match with ...33.bmp
Correct.....The input image...5.bmp  does not match with ...34.bmp
Correct.....The input image...5.bmp  does not match with ...35.bmp
Correct.....The input image...5.bmp  does not match with ...36.bmp
Correct.....The input image...5.bmp  does not match with ...37.bmp
Correct.....The input image...5.bmp  does not match with ...38.bmp
Correct.....The input image...5.bmp  does not match with ...39.bmp
Correct.....The input image...5.bmp  does not match with ...40.bmp
Correct.....The input image...5.bmp  does not match with ...41.bmp
Correct.....The input image...5.bmp  does not match with ...42.bmp
Correct.....The input image...5.bmp  does not match with ...43.bmp
Correct.....The input image...5.bmp  does not match with ...44.bmp
Correct.....The input image...5.bmp  does not match with ...45.bmp
Correct.....The input image...5.bmp  does not match with ...46.bmp
Correct.....The input image...5.bmp  does not match with ...47.bmp
Correct.....The input image...5.bmp  does not match with ...48.bmp
Correct.....The input image...5.bmp  does not match with ...49.bmp
Correct.....The input image...5.bmp  does not match with ...50.bmp
Correct.....The input image...5.bmp  does not match with ...51.bmp
Correct.....The input image...5.bmp  does not match with ...52.bmp
Correct.....The input image...5.bmp  does not match with ...53.bmp
Elapsed time is 14.103194 seconds.
```

Figure 5.4: Results of comparison with 50 subjects.

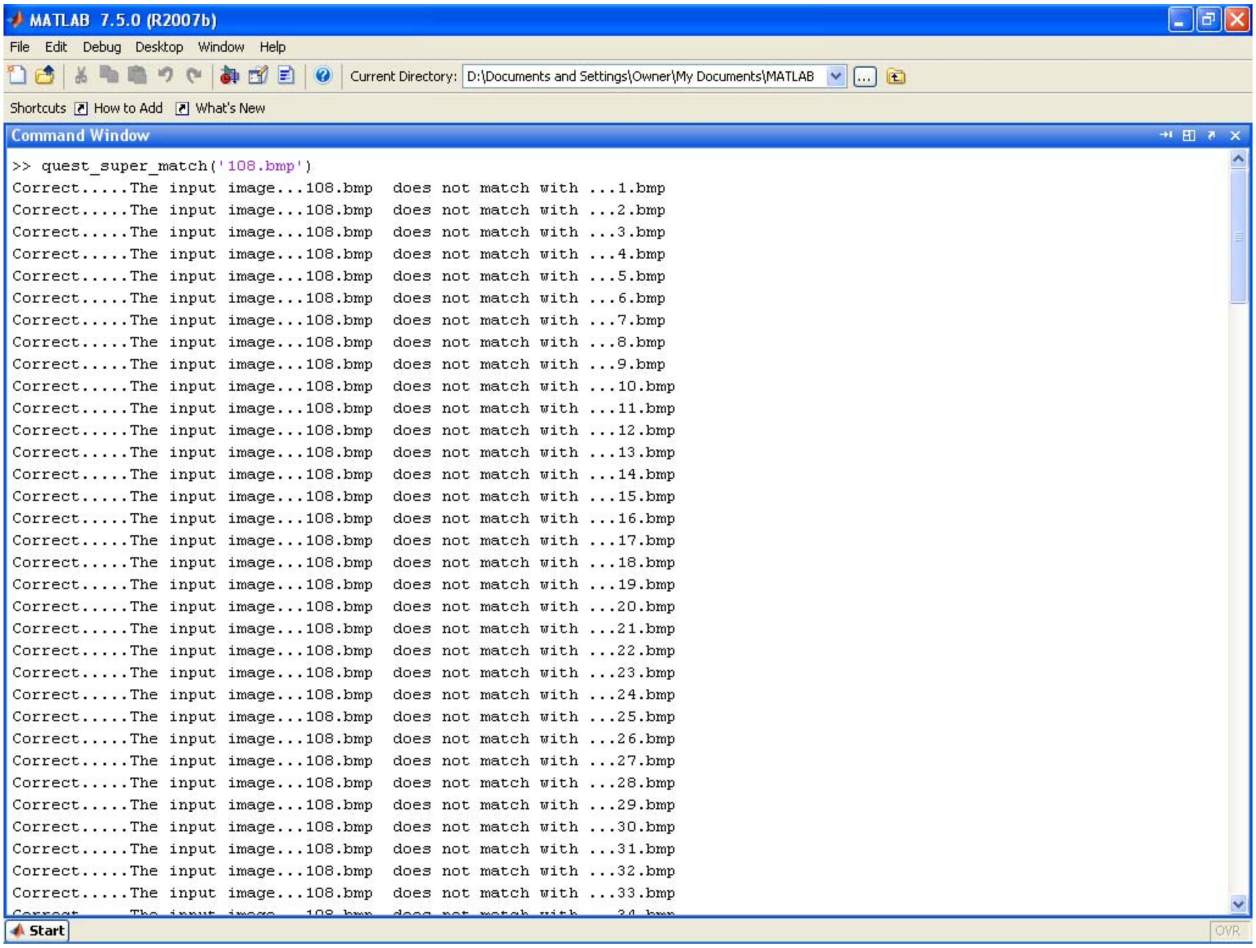
5.2 Test Case 2:

Implemented algorithm takes 200 templates from 30 different eyes as input. Feature templates are generated from these templates and are stored in database. After testing I deduce the following results.

Number of Images	%age Recognition Testing images
200	**95**

Table 5.2: **Percentage Recognition for 200 iris templates**

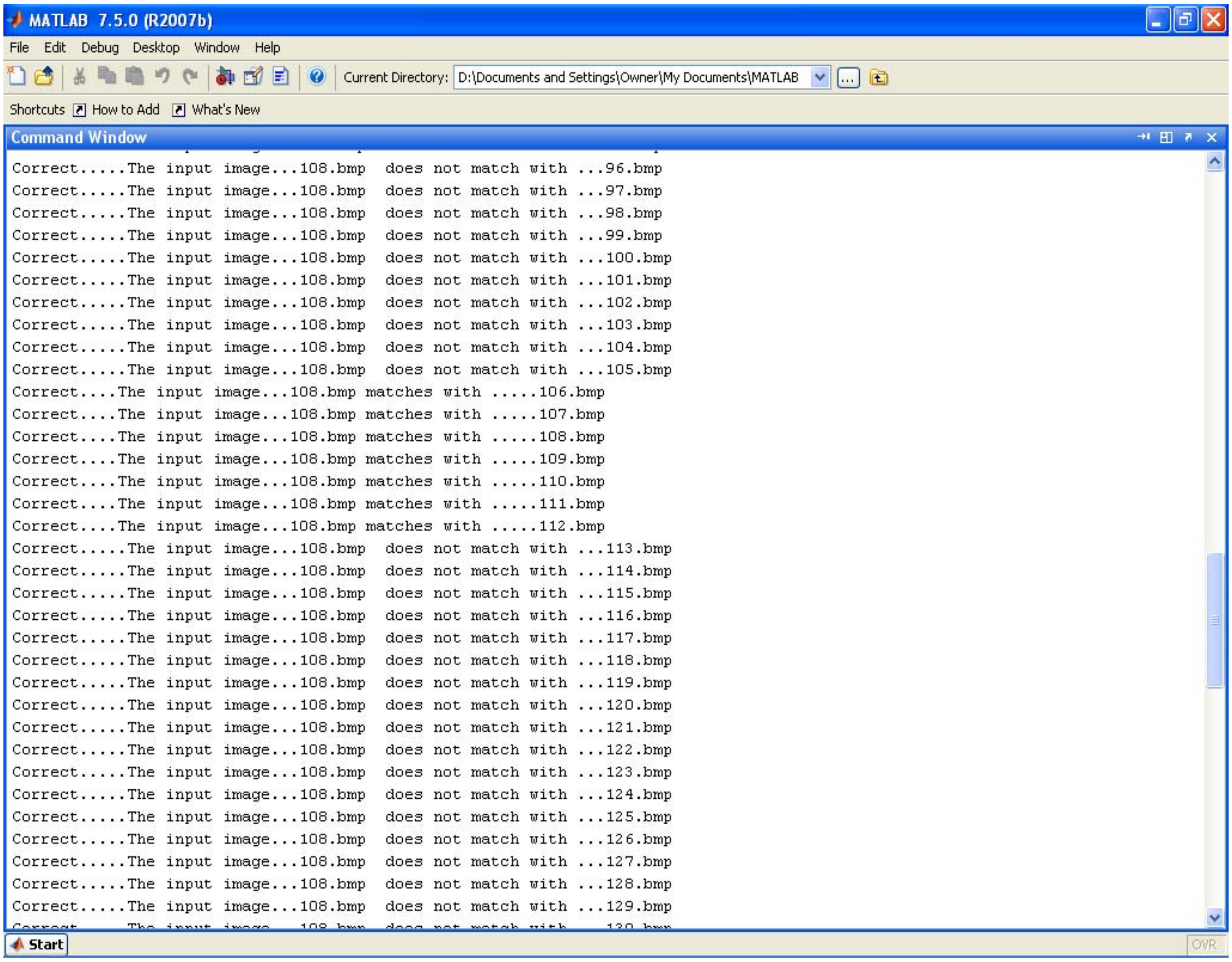
The following figure 5.5, 5.6 displays the testing results of a input image that is compared with all the templates in the database one by one.


MATLAB 7.5.0 (R2007b)
File Edit Debug Desktop Window Help
Shortcuts How to Add What's New
Command Window
Start
OVR

Current Directory: D:\Documents and Settings\Owner\My Documents\MATLAB

```
>> quest_super_match('108.bmp')
Correct.....The input image...108.bmp  does not match with ...1.bmp
Correct.....The input image...108.bmp  does not match with ...2.bmp
Correct.....The input image...108.bmp  does not match with ...3.bmp
Correct.....The input image...108.bmp  does not match with ...4.bmp
Correct.....The input image...108.bmp  does not match with ...5.bmp
Correct.....The input image...108.bmp  does not match with ...6.bmp
Correct.....The input image...108.bmp  does not match with ...7.bmp
Correct.....The input image...108.bmp  does not match with ...8.bmp
Correct.....The input image...108.bmp  does not match with ...9.bmp
Correct.....The input image...108.bmp  does not match with ...10.bmp
Correct.....The input image...108.bmp  does not match with ...11.bmp
Correct.....The input image...108.bmp  does not match with ...12.bmp
Correct.....The input image...108.bmp  does not match with ...13.bmp
Correct.....The input image...108.bmp  does not match with ...14.bmp
Correct.....The input image...108.bmp  does not match with ...15.bmp
Correct.....The input image...108.bmp  does not match with ...16.bmp
Correct.....The input image...108.bmp  does not match with ...17.bmp
Correct.....The input image...108.bmp  does not match with ...18.bmp
Correct.....The input image...108.bmp  does not match with ...19.bmp
Correct.....The input image...108.bmp  does not match with ...20.bmp
Correct.....The input image...108.bmp  does not match with ...21.bmp
Correct.....The input image...108.bmp  does not match with ...22.bmp
Correct.....The input image...108.bmp  does not match with ...23.bmp
Correct.....The input image...108.bmp  does not match with ...24.bmp
Correct.....The input image...108.bmp  does not match with ...25.bmp
Correct.....The input image...108.bmp  does not match with ...26.bmp
Correct.....The input image...108.bmp  does not match with ...27.bmp
Correct.....The input image...108.bmp  does not match with ...28.bmp
Correct.....The input image...108.bmp  does not match with ...29.bmp
Correct.....The input image...108.bmp  does not match with ...30.bmp
Correct.....The input image...108.bmp  does not match with ...31.bmp
Correct.....The input image...108.bmp  does not match with ...32.bmp
Correct.....The input image...108.bmp  does not match with ...33.bmp
Correct.....The input image...108.bmp  does not match with ...34.bmp
```

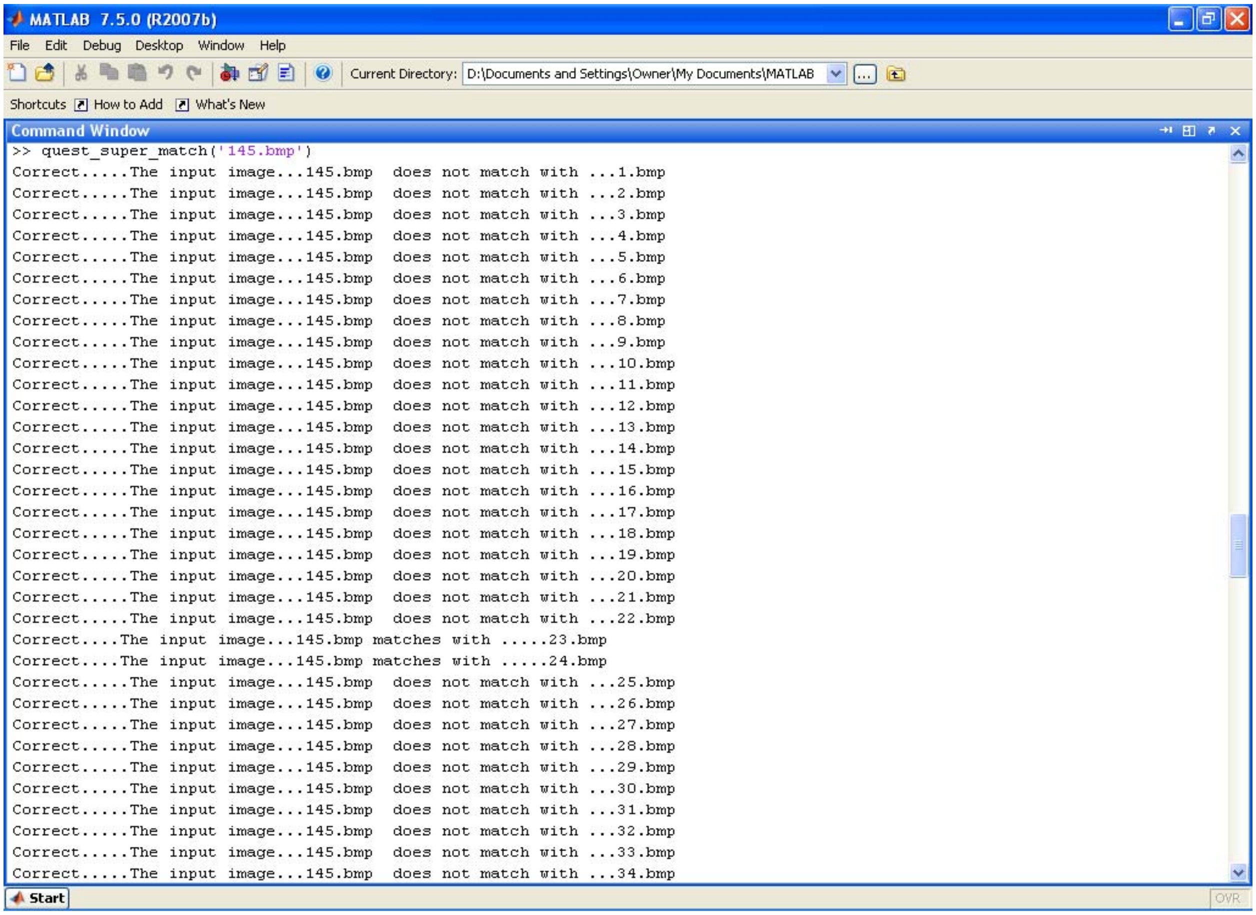
Figure 5.5: Results of testing of an image with 200 subjects.


MATLAB 7.5.0 (R2007b)
File Edit Debug Desktop Window Help
Current Directory: D:\Documents and Settings\Owner\My Documents\MATLAB
Shortcuts How to Add What's New
Command Window
Start
OVR

```
Correct.....The input image...108.bmp  does not match with ...96.bmp
Correct.....The input image...108.bmp  does not match with ...97.bmp
Correct.....The input image...108.bmp  does not match with ...98.bmp
Correct.....The input image...108.bmp  does not match with ...99.bmp
Correct.....The input image...108.bmp  does not match with ...100.bmp
Correct.....The input image...108.bmp  does not match with ...101.bmp
Correct.....The input image...108.bmp  does not match with ...102.bmp
Correct.....The input image...108.bmp  does not match with ...103.bmp
Correct.....The input image...108.bmp  does not match with ...104.bmp
Correct.....The input image...108.bmp  does not match with ...105.bmp
Correct....The input image...108.bmp matches with .....106.bmp
Correct....The input image...108.bmp matches with .....107.bmp
Correct....The input image...108.bmp matches with .....108.bmp
Correct....The input image...108.bmp matches with .....109.bmp
Correct....The input image...108.bmp matches with .....110.bmp
Correct....The input image...108.bmp matches with .....111.bmp
Correct....The input image...108.bmp matches with .....112.bmp
Correct.....The input image...108.bmp  does not match with ...113.bmp
Correct.....The input image...108.bmp  does not match with ...114.bmp
Correct.....The input image...108.bmp  does not match with ...115.bmp
Correct.....The input image...108.bmp  does not match with ...116.bmp
Correct.....The input image...108.bmp  does not match with ...117.bmp
Correct.....The input image...108.bmp  does not match with ...118.bmp
Correct.....The input image...108.bmp  does not match with ...119.bmp
Correct.....The input image...108.bmp  does not match with ...120.bmp
Correct.....The input image...108.bmp  does not match with ...121.bmp
Correct.....The input image...108.bmp  does not match with ...122.bmp
Correct.....The input image...108.bmp  does not match with ...123.bmp
Correct.....The input image...108.bmp  does not match with ...124.bmp
Correct.....The input image...108.bmp  does not match with ...125.bmp
Correct.....The input image...108.bmp  does not match with ...126.bmp
Correct.....The input image...108.bmp  does not match with ...127.bmp
Correct.....The input image...108.bmp  does not match with ...128.bmp
Correct.....The input image...108.bmp  does not match with ...129.bmp
Correct.....The input image...108.bmp  does not match with ...130.bmp
```

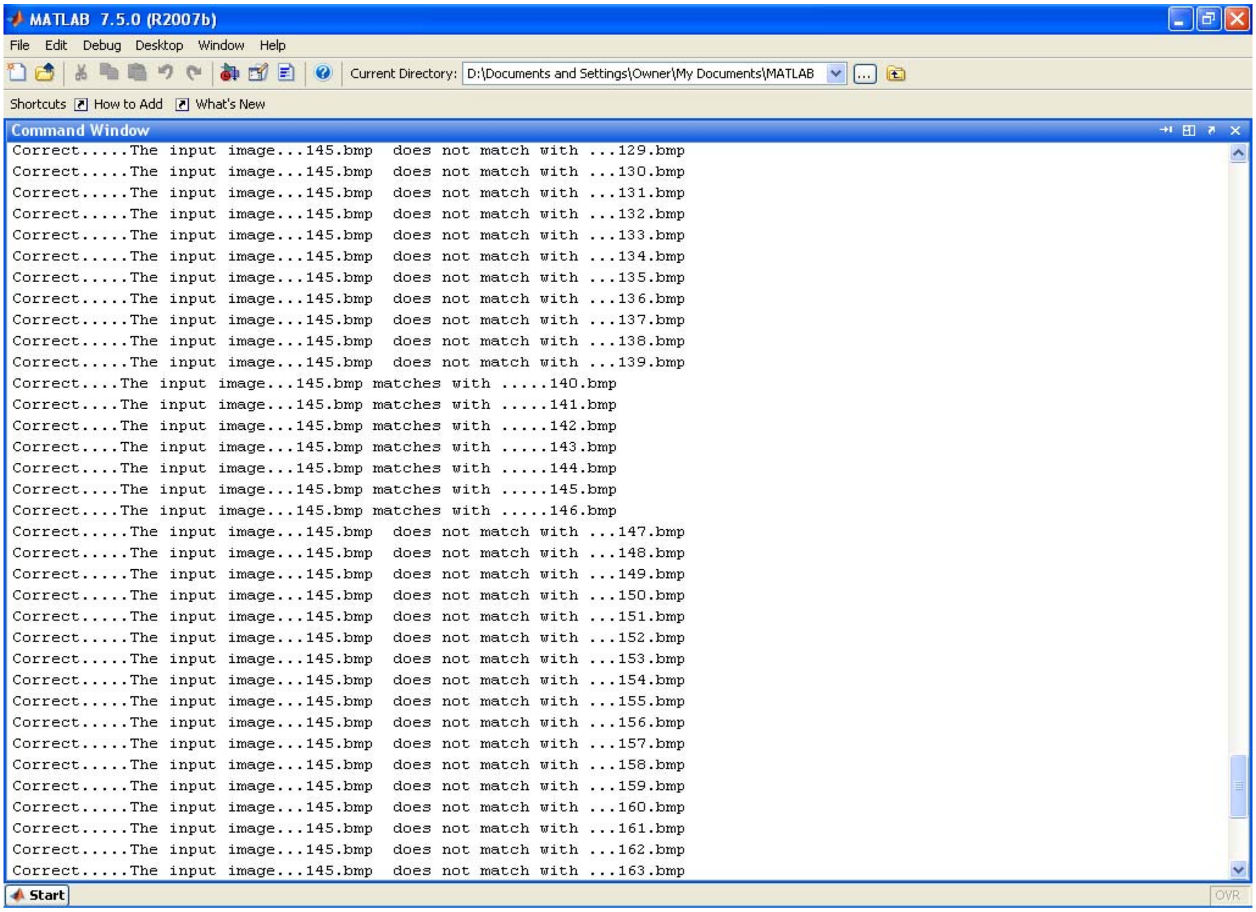
Figure 5.6: Results of testing of an image with 200 subjects.

The following figure 5.7, 5.8 displays the testing results of another input image that is compared with all the templates in the database one by one.


MATLAB 7.5.0 (R2007b)
File Edit Debug Desktop Window Help
Current Directory: D:\Documents and Settings\Owner\My Documents\MATLAB
Shortcuts How to Add What's New
Command Window
Start
OVR

```
>> quest_super_match('145.bmp')
Correct.....The input image...145.bmp  does not match with ...1.bmp
Correct.....The input image...145.bmp  does not match with ...2.bmp
Correct.....The input image...145.bmp  does not match with ...3.bmp
Correct.....The input image...145.bmp  does not match with ...4.bmp
Correct.....The input image...145.bmp  does not match with ...5.bmp
Correct.....The input image...145.bmp  does not match with ...6.bmp
Correct.....The input image...145.bmp  does not match with ...7.bmp
Correct.....The input image...145.bmp  does not match with ...8.bmp
Correct.....The input image...145.bmp  does not match with ...9.bmp
Correct.....The input image...145.bmp  does not match with ...10.bmp
Correct.....The input image...145.bmp  does not match with ...11.bmp
Correct.....The input image...145.bmp  does not match with ...12.bmp
Correct.....The input image...145.bmp  does not match with ...13.bmp
Correct.....The input image...145.bmp  does not match with ...14.bmp
Correct.....The input image...145.bmp  does not match with ...15.bmp
Correct.....The input image...145.bmp  does not match with ...16.bmp
Correct.....The input image...145.bmp  does not match with ...17.bmp
Correct.....The input image...145.bmp  does not match with ...18.bmp
Correct.....The input image...145.bmp  does not match with ...19.bmp
Correct.....The input image...145.bmp  does not match with ...20.bmp
Correct.....The input image...145.bmp  does not match with ...21.bmp
Correct.....The input image...145.bmp  does not match with ...22.bmp
Correct....The input image...145.bmp matches with .....23.bmp
Correct....The input image...145.bmp matches with .....24.bmp
Correct.....The input image...145.bmp  does not match with ...25.bmp
Correct.....The input image...145.bmp  does not match with ...26.bmp
Correct.....The input image...145.bmp  does not match with ...27.bmp
Correct.....The input image...145.bmp  does not match with ...28.bmp
Correct.....The input image...145.bmp  does not match with ...29.bmp
Correct.....The input image...145.bmp  does not match with ...30.bmp
Correct.....The input image...145.bmp  does not match with ...31.bmp
Correct.....The input image...145.bmp  does not match with ...32.bmp
Correct.....The input image...145.bmp  does not match with ...33.bmp
Correct.....The input image...145.bmp  does not match with ...34.bmp
```

Figure 5.7: Results of testing of another image with 200 subjects.


MATLAB 7.5.0 (R2007b)
File Edit Debug Desktop Window Help
Current Directory: D:\Documents and Settings\Owner\My Documents\MATLAB
Shortcuts How to Add What's New
Command Window
Start
OVR

```
Correct.....The input image...145.bmp  does not match with ...129.bmp
Correct.....The input image...145.bmp  does not match with ...130.bmp
Correct.....The input image...145.bmp  does not match with ...131.bmp
Correct.....The input image...145.bmp  does not match with ...132.bmp
Correct.....The input image...145.bmp  does not match with ...133.bmp
Correct.....The input image...145.bmp  does not match with ...134.bmp
Correct.....The input image...145.bmp  does not match with ...135.bmp
Correct.....The input image...145.bmp  does not match with ...136.bmp
Correct.....The input image...145.bmp  does not match with ...137.bmp
Correct.....The input image...145.bmp  does not match with ...138.bmp
Correct.....The input image...145.bmp  does not match with ...139.bmp
Correct....The input image...145.bmp matches with .....140.bmp
Correct....The input image...145.bmp matches with .....141.bmp
Correct....The input image...145.bmp matches with .....142.bmp
Correct....The input image...145.bmp matches with .....143.bmp
Correct....The input image...145.bmp matches with .....144.bmp
Correct....The input image...145.bmp matches with .....145.bmp
Correct....The input image...145.bmp matches with .....146.bmp
Correct.....The input image...145.bmp  does not match with ...147.bmp
Correct.....The input image...145.bmp  does not match with ...148.bmp
Correct.....The input image...145.bmp  does not match with ...149.bmp
Correct.....The input image...145.bmp  does not match with ...150.bmp
Correct.....The input image...145.bmp  does not match with ...151.bmp
Correct.....The input image...145.bmp  does not match with ...152.bmp
Correct.....The input image...145.bmp  does not match with ...153.bmp
Correct.....The input image...145.bmp  does not match with ...154.bmp
Correct.....The input image...145.bmp  does not match with ...155.bmp
Correct.....The input image...145.bmp  does not match with ...156.bmp
Correct.....The input image...145.bmp  does not match with ...157.bmp
Correct.....The input image...145.bmp  does not match with ...158.bmp
Correct.....The input image...145.bmp  does not match with ...159.bmp
Correct.....The input image...145.bmp  does not match with ...160.bmp
Correct.....The input image...145.bmp  does not match with ...161.bmp
Correct.....The input image...145.bmp  does not match with ...162.bmp
Correct.....The input image...145.bmp  does not match with ...163.bmp
```

Figure 5.8: Results of testing of an image with 200 subjects.

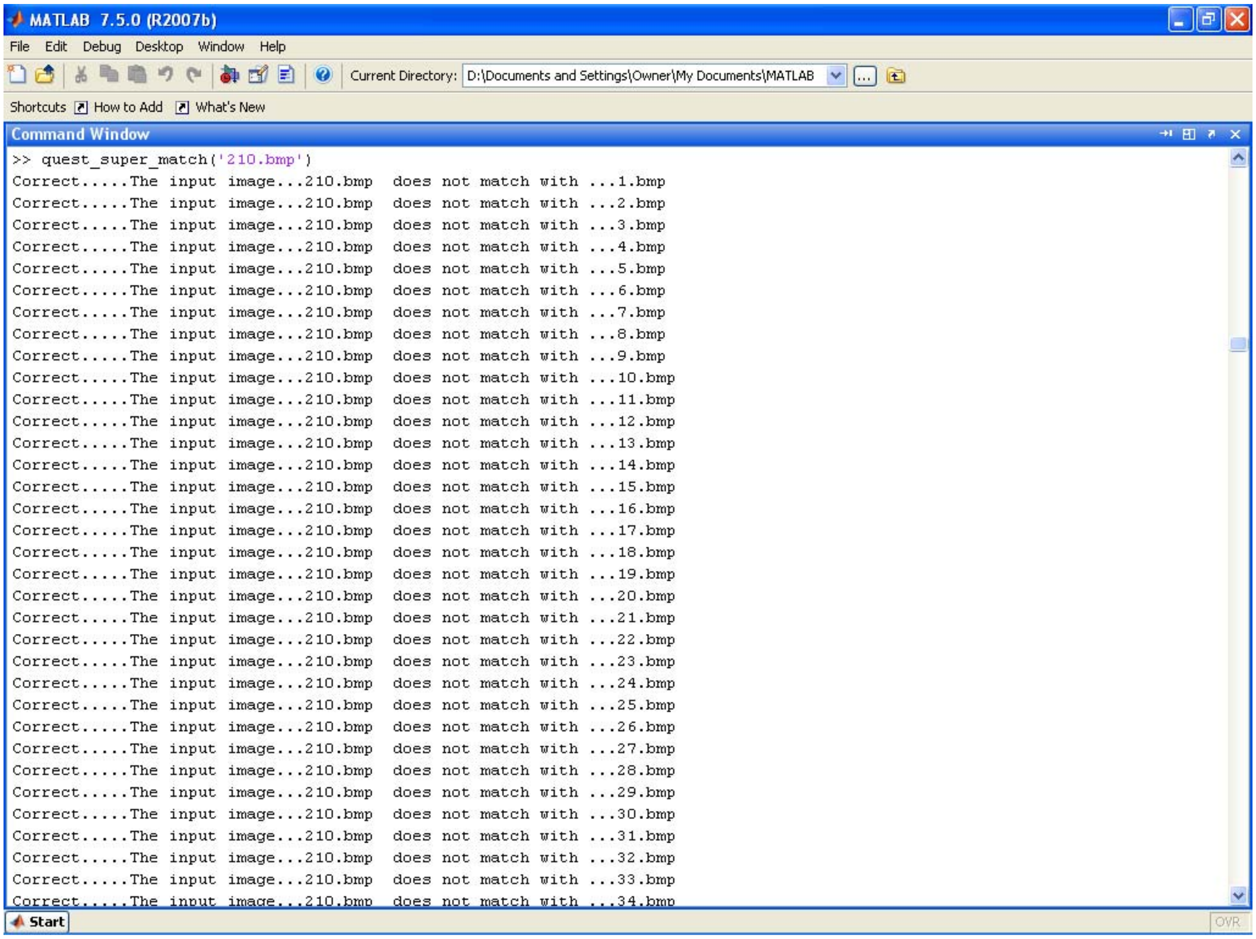
5.3 Test Case 3:

Implemented algorithm takes 400 templates from 66 different eyes as input. Feature templates are generated from these templates and are stored in database. After testing I deduce the following results.

Number of Images	%age Recognition Testing images
400	**96**

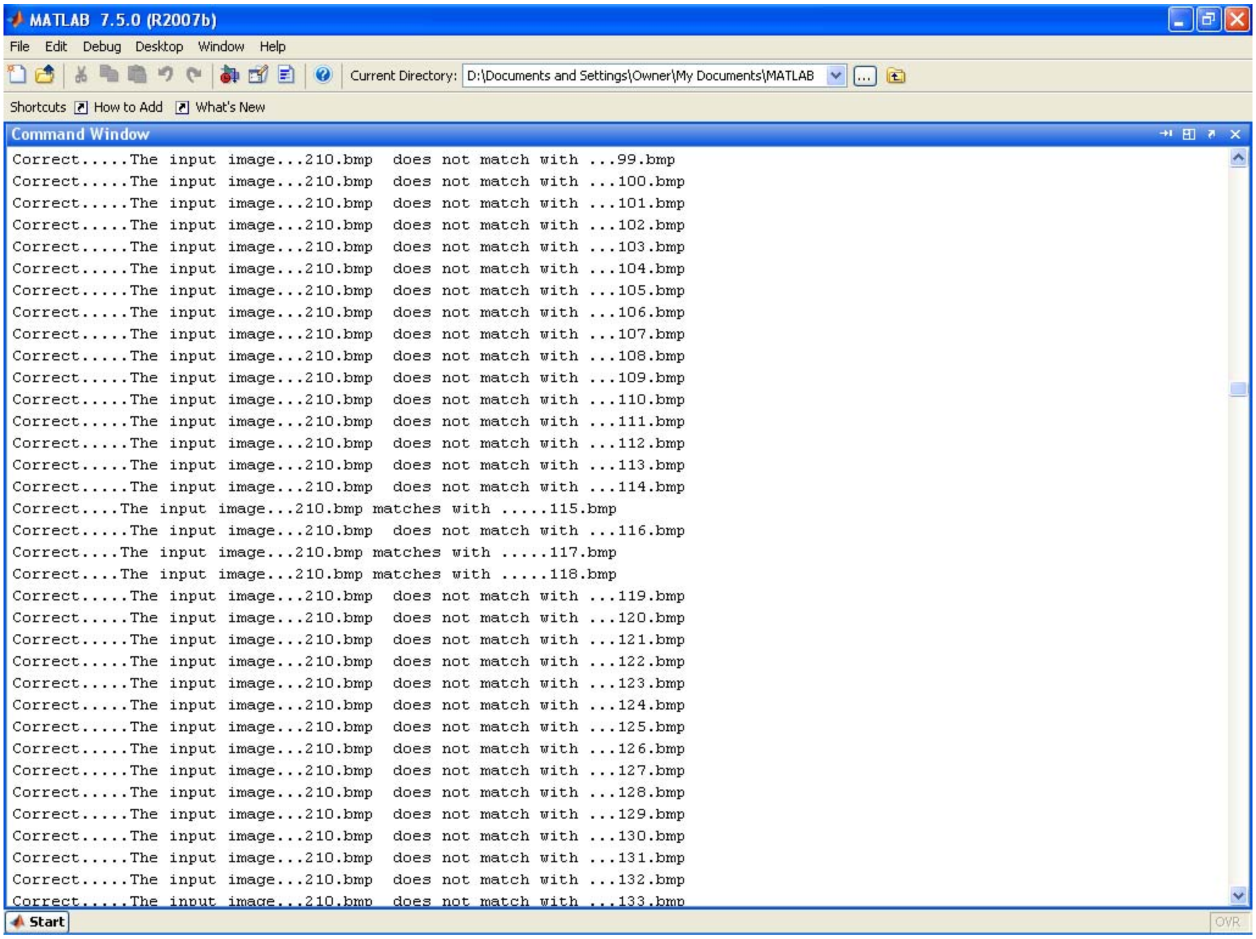
Table 5.3: **Percentage Recognition for 200 iris templates**

The following figures 5.9, 5.10, 5.11, 5.12 display the testing results of a input image that is compared with all the templates in the database one by one.



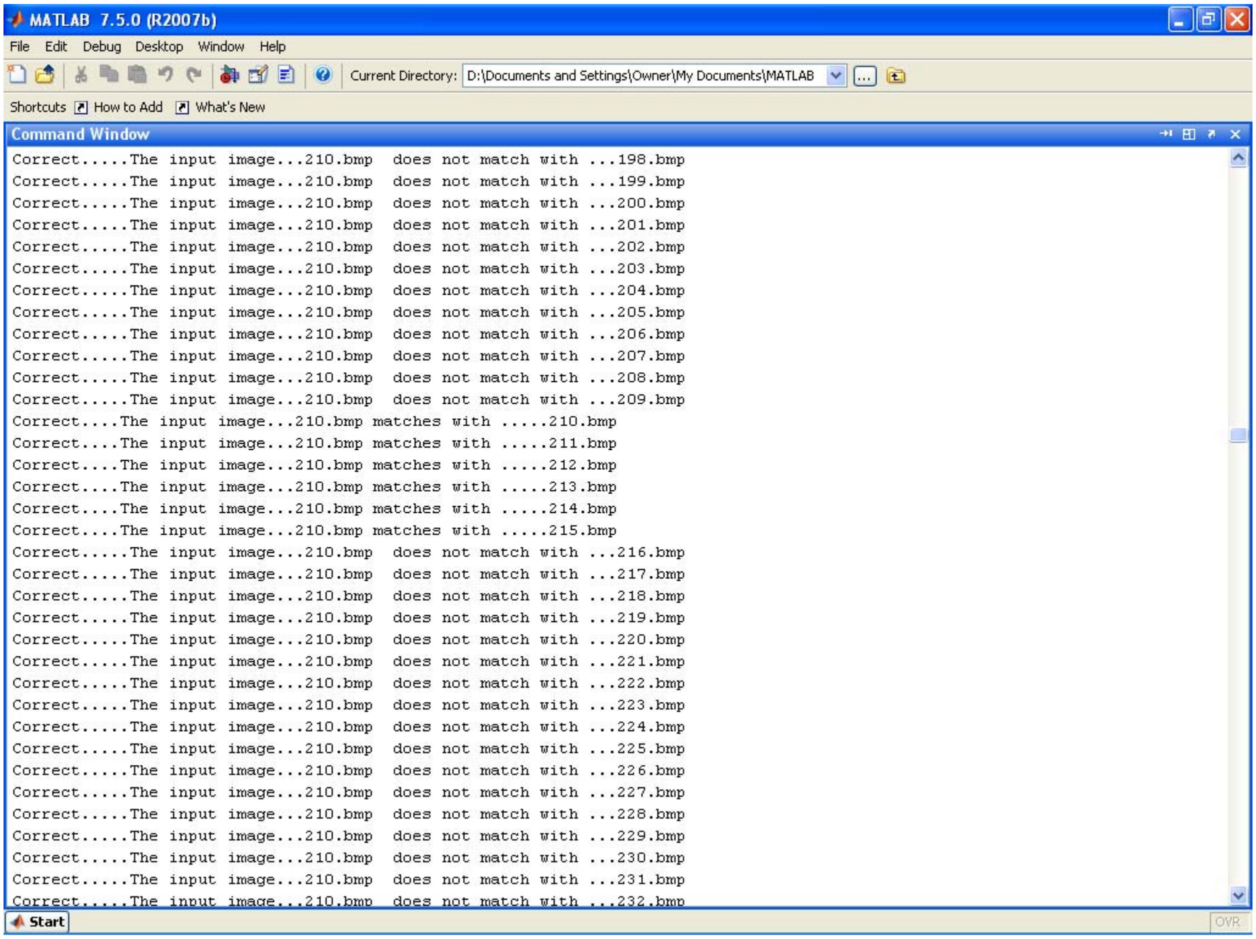
```
>> quest_super_match('210.bmp')
Correct.....The input image...210.bmp  does not match with ...1.bmp
Correct.....The input image...210.bmp  does not match with ...2.bmp
Correct.....The input image...210.bmp  does not match with ...3.bmp
Correct.....The input image...210.bmp  does not match with ...4.bmp
Correct.....The input image...210.bmp  does not match with ...5.bmp
Correct.....The input image...210.bmp  does not match with ...6.bmp
Correct.....The input image...210.bmp  does not match with ...7.bmp
Correct.....The input image...210.bmp  does not match with ...8.bmp
Correct.....The input image...210.bmp  does not match with ...9.bmp
Correct.....The input image...210.bmp  does not match with ...10.bmp
Correct.....The input image...210.bmp  does not match with ...11.bmp
Correct.....The input image...210.bmp  does not match with ...12.bmp
Correct.....The input image...210.bmp  does not match with ...13.bmp
Correct.....The input image...210.bmp  does not match with ...14.bmp
Correct.....The input image...210.bmp  does not match with ...15.bmp
Correct.....The input image...210.bmp  does not match with ...16.bmp
Correct.....The input image...210.bmp  does not match with ...17.bmp
Correct.....The input image...210.bmp  does not match with ...18.bmp
Correct.....The input image...210.bmp  does not match with ...19.bmp
Correct.....The input image...210.bmp  does not match with ...20.bmp
Correct.....The input image...210.bmp  does not match with ...21.bmp
Correct.....The input image...210.bmp  does not match with ...22.bmp
Correct.....The input image...210.bmp  does not match with ...23.bmp
Correct.....The input image...210.bmp  does not match with ...24.bmp
Correct.....The input image...210.bmp  does not match with ...25.bmp
Correct.....The input image...210.bmp  does not match with ...26.bmp
Correct.....The input image...210.bmp  does not match with ...27.bmp
Correct.....The input image...210.bmp  does not match with ...28.bmp
Correct.....The input image...210.bmp  does not match with ...29.bmp
Correct.....The input image...210.bmp  does not match with ...30.bmp
Correct.....The input image...210.bmp  does not match with ...31.bmp
Correct.....The input image...210.bmp  does not match with ...32.bmp
Correct.....The input image...210.bmp  does not match with ...33.bmp
Correct.....The input image...210.bmp  does not match with ...34.bmp
```

Figure 5.9: Results of testing of an image with 400 subjects.


MATLAB 7.5.0 (R2007b)
File Edit Debug Desktop Window Help
Current Directory: D:\Documents and Settings\Owner\My Documents\MATLAB
Shortcuts How to Add What's New
Command Window
Start
OVR

```
Correct.....The input image...210.bmp  does not match with ...99.bmp
Correct.....The input image...210.bmp  does not match with ...100.bmp
Correct.....The input image...210.bmp  does not match with ...101.bmp
Correct.....The input image...210.bmp  does not match with ...102.bmp
Correct.....The input image...210.bmp  does not match with ...103.bmp
Correct.....The input image...210.bmp  does not match with ...104.bmp
Correct.....The input image...210.bmp  does not match with ...105.bmp
Correct.....The input image...210.bmp  does not match with ...106.bmp
Correct.....The input image...210.bmp  does not match with ...107.bmp
Correct.....The input image...210.bmp  does not match with ...108.bmp
Correct.....The input image...210.bmp  does not match with ...109.bmp
Correct.....The input image...210.bmp  does not match with ...110.bmp
Correct.....The input image...210.bmp  does not match with ...111.bmp
Correct.....The input image...210.bmp  does not match with ...112.bmp
Correct.....The input image...210.bmp  does not match with ...113.bmp
Correct.....The input image...210.bmp  does not match with ...114.bmp
Correct....The input image...210.bmp matches with .....115.bmp
Correct.....The input image...210.bmp  does not match with ...116.bmp
Correct....The input image...210.bmp matches with .....117.bmp
Correct....The input image...210.bmp matches with .....118.bmp
Correct.....The input image...210.bmp  does not match with ...119.bmp
Correct.....The input image...210.bmp  does not match with ...120.bmp
Correct.....The input image...210.bmp  does not match with ...121.bmp
Correct.....The input image...210.bmp  does not match with ...122.bmp
Correct.....The input image...210.bmp  does not match with ...123.bmp
Correct.....The input image...210.bmp  does not match with ...124.bmp
Correct.....The input image...210.bmp  does not match with ...125.bmp
Correct.....The input image...210.bmp  does not match with ...126.bmp
Correct.....The input image...210.bmp  does not match with ...127.bmp
Correct.....The input image...210.bmp  does not match with ...128.bmp
Correct.....The input image...210.bmp  does not match with ...129.bmp
Correct.....The input image...210.bmp  does not match with ...130.bmp
Correct.....The input image...210.bmp  does not match with ...131.bmp
Correct.....The input image...210.bmp  does not match with ...132.bmp
Correct.....The input image...210.bmp  does not match with ...133.bmp
```

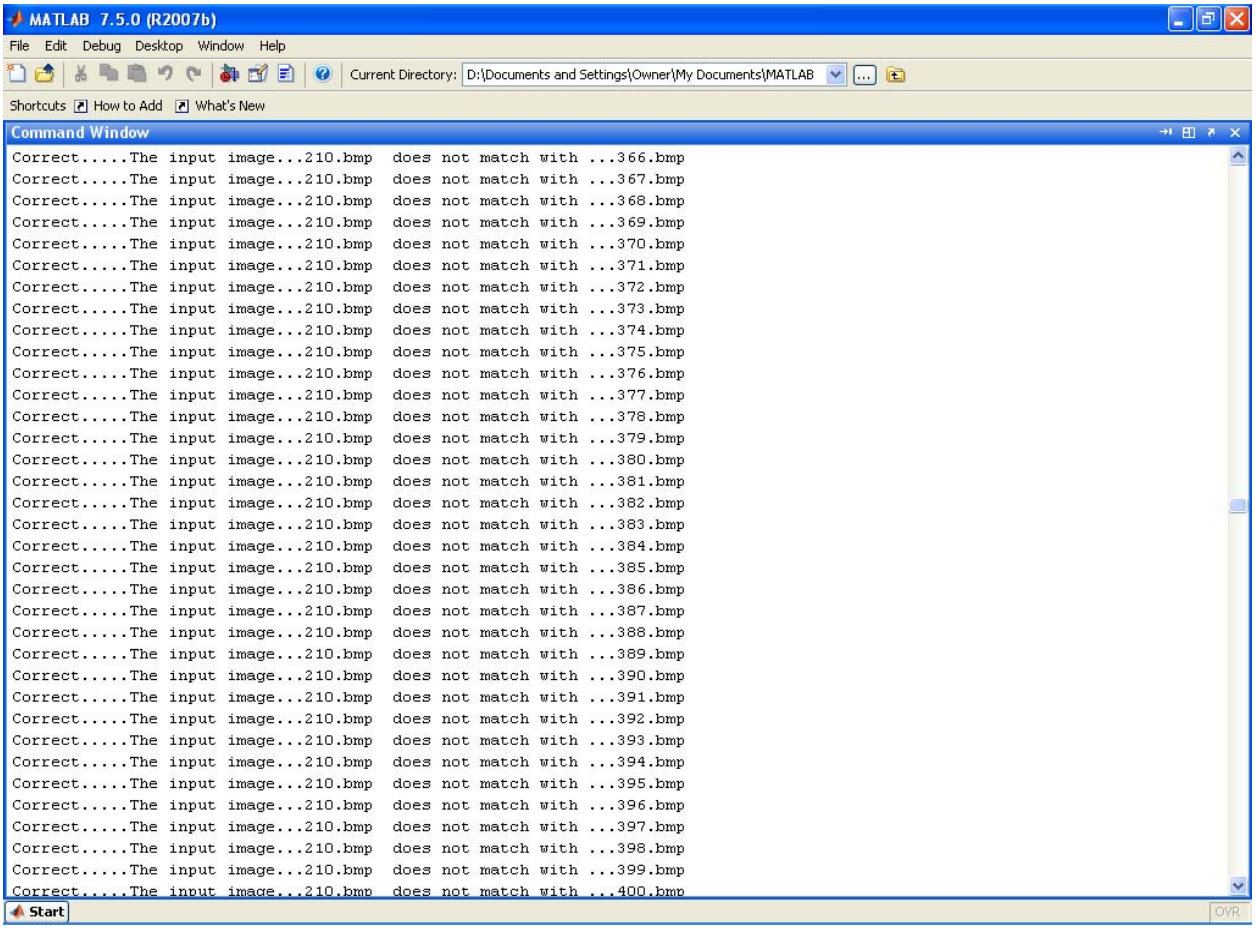
Figure 5.10: Results of testing of an image with 400 subjects.



Current Directory: D:\Documents and Settings\Owner\My Documents\MATLAB

```
Correct.....The input image...210.bmp  does not match with ...198.bmp
Correct.....The input image...210.bmp  does not match with ...199.bmp
Correct.....The input image...210.bmp  does not match with ...200.bmp
Correct.....The input image...210.bmp  does not match with ...201.bmp
Correct.....The input image...210.bmp  does not match with ...202.bmp
Correct.....The input image...210.bmp  does not match with ...203.bmp
Correct.....The input image...210.bmp  does not match with ...204.bmp
Correct.....The input image...210.bmp  does not match with ...205.bmp
Correct.....The input image...210.bmp  does not match with ...206.bmp
Correct.....The input image...210.bmp  does not match with ...207.bmp
Correct.....The input image...210.bmp  does not match with ...208.bmp
Correct.....The input image...210.bmp  does not match with ...209.bmp
Correct....The input image...210.bmp matches with .....210.bmp
Correct....The input image...210.bmp matches with .....211.bmp
Correct....The input image...210.bmp matches with .....212.bmp
Correct....The input image...210.bmp matches with .....213.bmp
Correct....The input image...210.bmp matches with .....214.bmp
Correct....The input image...210.bmp matches with .....215.bmp
Correct.....The input image...210.bmp  does not match with ...216.bmp
Correct.....The input image...210.bmp  does not match with ...217.bmp
Correct.....The input image...210.bmp  does not match with ...218.bmp
Correct.....The input image...210.bmp  does not match with ...219.bmp
Correct.....The input image...210.bmp  does not match with ...220.bmp
Correct.....The input image...210.bmp  does not match with ...221.bmp
Correct.....The input image...210.bmp  does not match with ...222.bmp
Correct.....The input image...210.bmp  does not match with ...223.bmp
Correct.....The input image...210.bmp  does not match with ...224.bmp
Correct.....The input image...210.bmp  does not match with ...225.bmp
Correct.....The input image...210.bmp  does not match with ...226.bmp
Correct.....The input image...210.bmp  does not match with ...227.bmp
Correct.....The input image...210.bmp  does not match with ...228.bmp
Correct.....The input image...210.bmp  does not match with ...229.bmp
Correct.....The input image...210.bmp  does not match with ...230.bmp
Correct.....The input image...210.bmp  does not match with ...231.bmp
Correct.....The input image...210.bmp  does not match with ...232.bmp
```

Figure 5.11: Results of testing of an image with 400 subjects.


MATLAB 7.5.0 (R2007b)
File Edit Debug Desktop Window Help
Current Directory: D:\Documents and Settings\Owner\My Documents\MATLAB
Shortcuts How to Add What's New
Command Window
Start
OVR

```
Correct.....The input image...210.bmp  does not match with ...366.bmp
Correct.....The input image...210.bmp  does not match with ...367.bmp
Correct.....The input image...210.bmp  does not match with ...368.bmp
Correct.....The input image...210.bmp  does not match with ...369.bmp
Correct.....The input image...210.bmp  does not match with ...370.bmp
Correct.....The input image...210.bmp  does not match with ...371.bmp
Correct.....The input image...210.bmp  does not match with ...372.bmp
Correct.....The input image...210.bmp  does not match with ...373.bmp
Correct.....The input image...210.bmp  does not match with ...374.bmp
Correct.....The input image...210.bmp  does not match with ...375.bmp
Correct.....The input image...210.bmp  does not match with ...376.bmp
Correct.....The input image...210.bmp  does not match with ...377.bmp
Correct.....The input image...210.bmp  does not match with ...378.bmp
Correct.....The input image...210.bmp  does not match with ...379.bmp
Correct.....The input image...210.bmp  does not match with ...380.bmp
Correct.....The input image...210.bmp  does not match with ...381.bmp
Correct.....The input image...210.bmp  does not match with ...382.bmp
Correct.....The input image...210.bmp  does not match with ...383.bmp
Correct.....The input image...210.bmp  does not match with ...384.bmp
Correct.....The input image...210.bmp  does not match with ...385.bmp
Correct.....The input image...210.bmp  does not match with ...386.bmp
Correct.....The input image...210.bmp  does not match with ...387.bmp
Correct.....The input image...210.bmp  does not match with ...388.bmp
Correct.....The input image...210.bmp  does not match with ...389.bmp
Correct.....The input image...210.bmp  does not match with ...390.bmp
Correct.....The input image...210.bmp  does not match with ...391.bmp
Correct.....The input image...210.bmp  does not match with ...392.bmp
Correct.....The input image...210.bmp  does not match with ...393.bmp
Correct.....The input image...210.bmp  does not match with ...394.bmp
Correct.....The input image...210.bmp  does not match with ...395.bmp
Correct.....The input image...210.bmp  does not match with ...396.bmp
Correct.....The input image...210.bmp  does not match with ...397.bmp
Correct.....The input image...210.bmp  does not match with ...398.bmp
Correct.....The input image...210.bmp  does not match with ...399.bmp
Correct.....The input image...210.bmp  does not match with ...400.bmp
```

Figure 5.12: Results of testing of an image with 400 subjects.

5.4 Comparison of Results with other techniques

The algorithm was implemented using 3.0 GHZ Pentium 4 machine with Windows XP and MATLAB 7.5 as the development tool. This section compares results of my technique with the results of other techniques i.e. iris recognition using neural networks and libor masek's implementation of iris recognition. All the images are stored on hard drive to acquire the memory saving.

First comparison shown is the testing time per image comparison. The time taken by the proposed technique and other techniques is displayed in the graph. The time values are calculated against 50,100,200 and 400 images. Also the total testing time of proposed technique and other techniques are also displayed in another graph. In another graph the percentage of correct decisions are displayed for the proposed technique and other techniques.

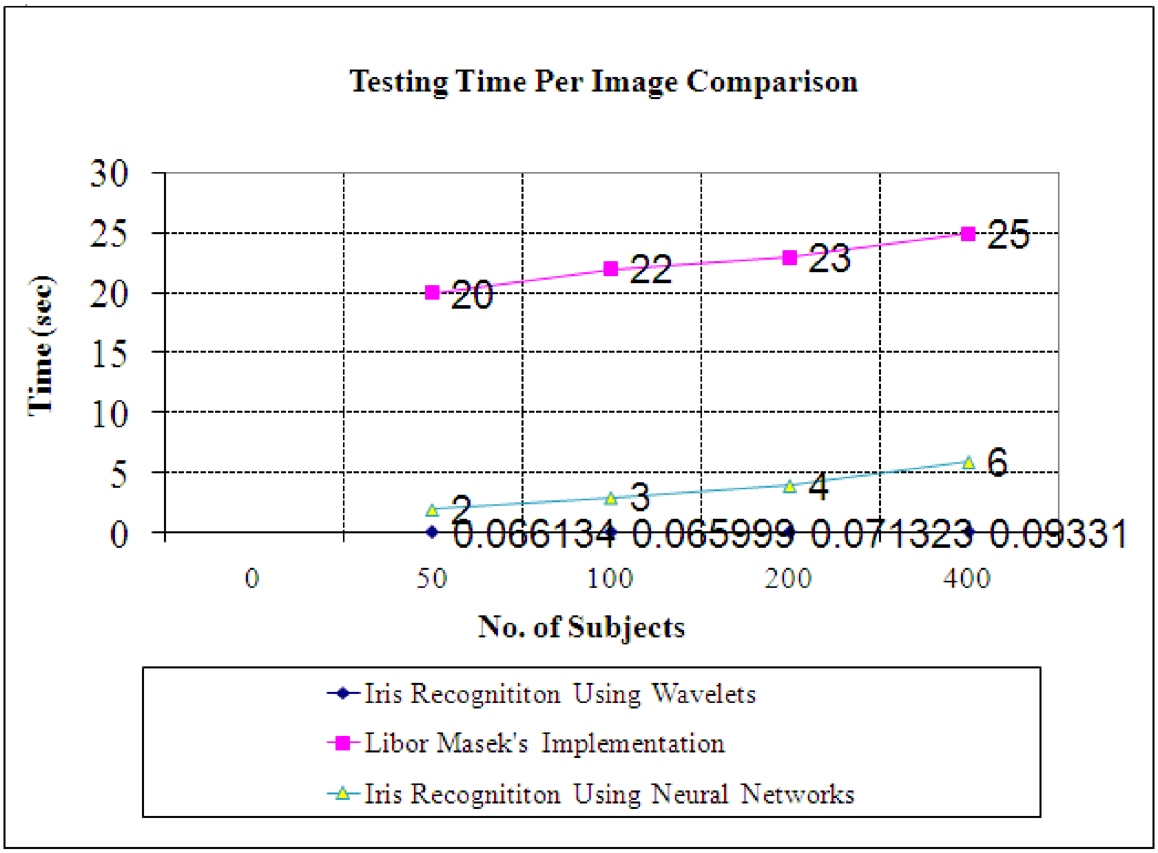


Figure 5.13: Testing time per image comparison

The graph in figure 5.13 displays the comparison result for testing time per image of the proposed technique with other techniques i.e. NN technique and iris implementation by labor masek. Here it can be clearly seen that the proposed technique outperforms other techniques because of the effective and fast processing of the suggested algorithm.

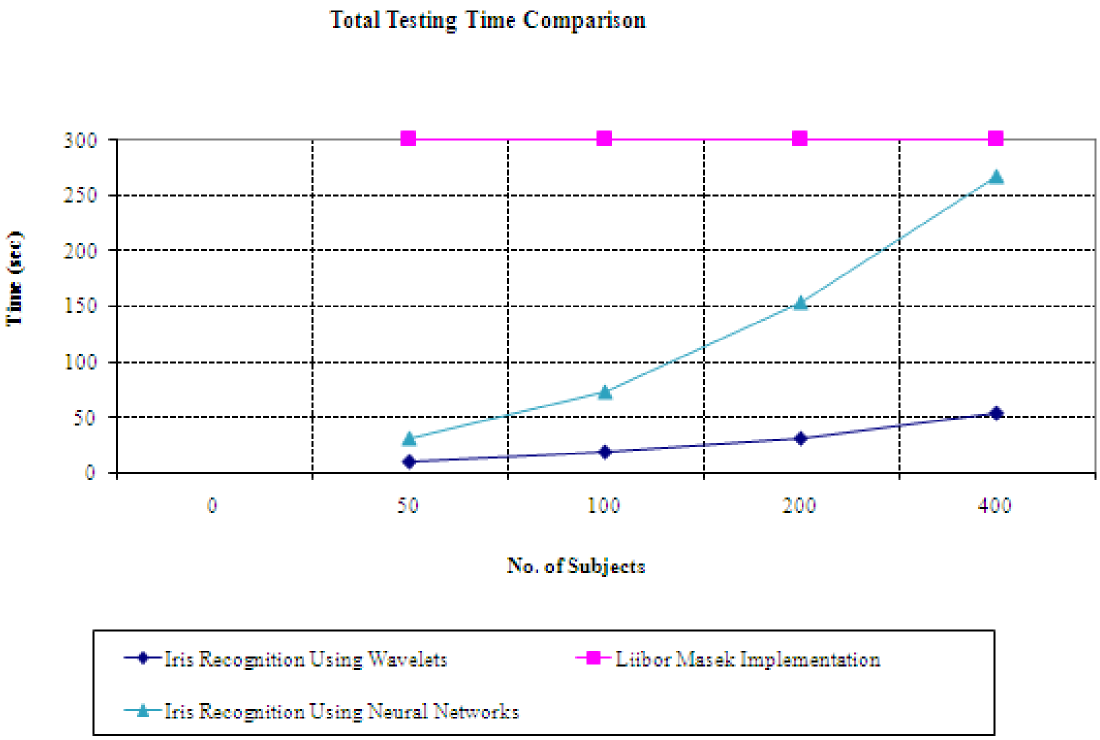


Figure 5.14: Total testing time comparison

The graph in figure 5.14 shows the total testing time required for various techniques including the proposed work. The graph is plotted as number of subjects against time required to compare images in seconds. The testing time is proportional with number of subjects, it increases with the number of subjects and vice versa but it increases linearly for the proposed technique as compared to the exponential increase of the labor masek's implementation.

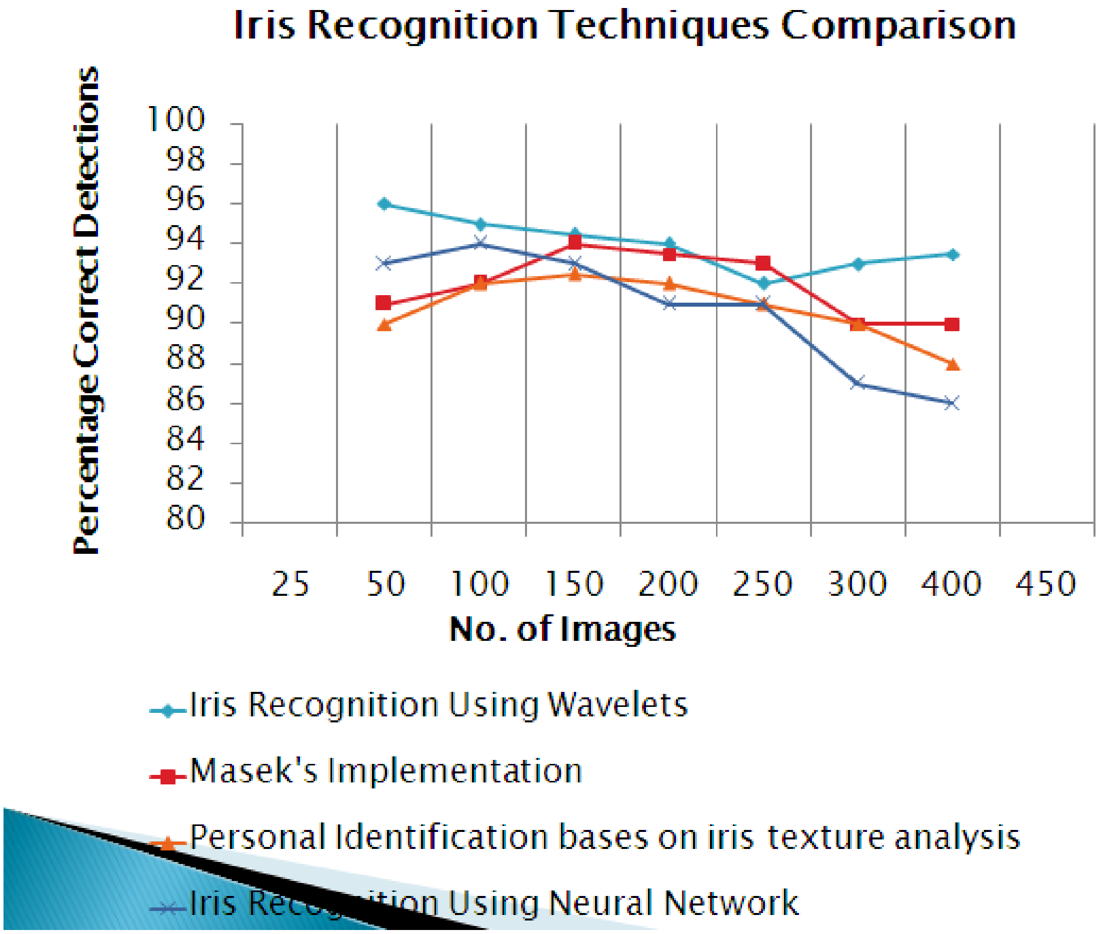


Figure 5.15: Percentage of correct decisions

The graph in figure 5.15 shows the comparison result of various techniques including the proposed work. The graph is plotted as number of subjects against percentage of correct decisions. It can be clearly seen that the proposed technique is outperforming the other techniques.

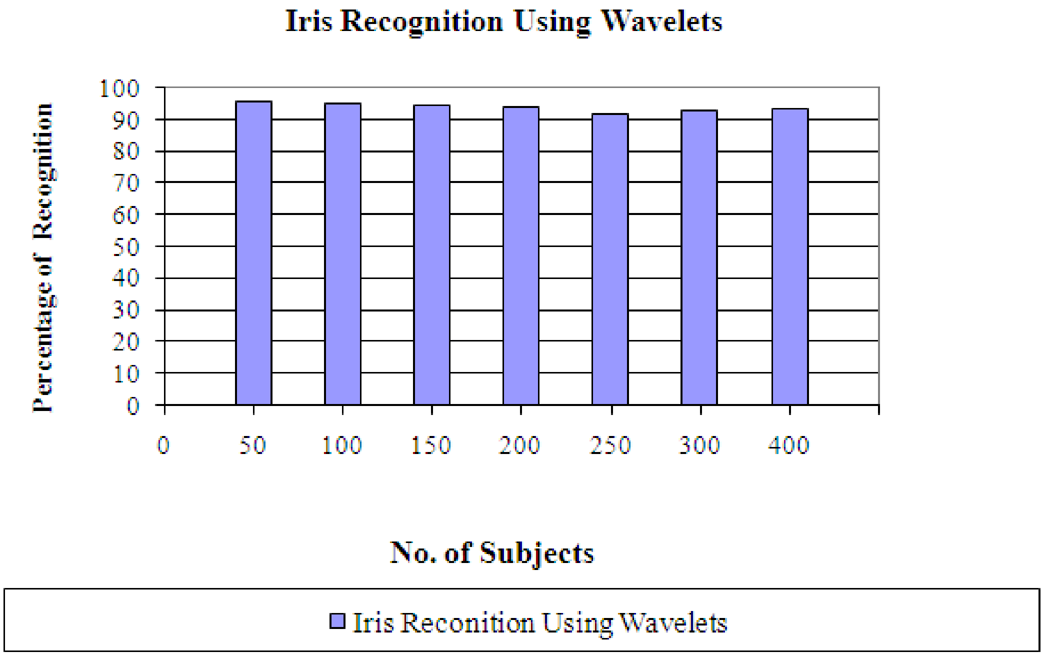


Figure 5.16: Percentage of recognition

Figure 5.16 displays the proposed implemented technique percentage of recognition at various number of subjects, as the size of database is increased, percentage deteriorates a bit, but still giving such high recognition rate without any heavy pre-processing technique and just by using time effective pre-processing and implementation.

5.5 Comparison of Rate of Success on Other Iris Databases

In this section the experimental results of the system are presented for the different stages of the iris recognition system. Three iris eye images databases were used:

Database	CASIA	MMU	Uni. Of Bath database
Rate of success	96%	66%	70%

The propose system is developed primarily on CASIA. The results are varied for other databases because the each database has different images having different size, noise level, illumination, reflections etc.

Chapter 6: Future Work & Conclusion

For at least a century [4], it has been suggested that the iris can sub serve biometrically based recognition of human individuals. Recent efforts in machine vision have yielded automated systems that take strides toward realizing this potential. As currently instantiated, these systems are relatively compact and efficient and have shown promising performance in preliminary testing. Extant systems require a fair amount of operator participation and work at rather close range. Therefore, they are best suited to controlled assessment scenarios (e.g., portal entry and the like).The notion that the iris is a useful biometric for recognition stems largely from anecdotal clinical and indirect developmental evidence. This body of evidence suggests that the structure of individual irises is highly distinctive and stable with age. Empirical testing of documented iris recognition systems provides additional support for these claims; however, these tests were limited in scope.

An important direction for future efforts is the design and execution of controlled, large-scale, longitudinal studies. Only via reference to such studies can the true accuracy of iris recognition be determined for both the verification and identification tasks. Another potentially rich direction for future research would be to relax the constraints under which current iris-recognition systems operate. In this regard, it would be particularly desirable to decrease the required level of operator participation even while increasing the physical distance from which evaluation takes place. If such goals can be achieved, then iris recognition can provide the basis for truly noninvasive biometric assessment. Further, if these enhancements can be had while maintaining compact, efficient, and low-cost implementations, then iris recognition will be well positioned for widespread deployment.

A new iris recognition system is provided in this paper that's implements Integro differential operator in a new optimized manner which resulted in time reduction to perform the comparisons as well as provide more efficiency. After the detection of iris area, normalized template is calculated. From normalized template, the features are extracted after convolving the normalized template with wavelets and then the feature template is stored in the database.

Iris recognition, as a biometric technology, has great advantages, such as variability, stability and security, thus it will have a variety of applications. Reliable automatic recognition of persons has long been an attractive goal. As in all pattern recognition problems, the key issue is the relation between interclass and intra-class variability [2]: objects can be reliably classified only if the variability among different instances of a given class is less than the variability between different classes. For example in face recognition, difficulties arise from the fact that the face is a changeable social organ displaying a variety of expressions, as well as being an active 3D object whose image varies with viewing angle, pose, illumination, accoutrements, and age. It has been shown that for facial images taken at least one year apart, even the best current algorithms have error rates of 43% (Phillips et al. 2000) to 50% (Pentland et al. 2000). Against this intra-class (same face) variability, inter-class variability is limited because different faces possess the same basic set of features, in the same canonical geometry.

For all of these reasons, iris patterns become interesting as an alternative approach to reliable visual recognition of persons when imaging can be done at distances of less than a meter, and especially when there is a need to search very large databases without incurring any false matches despite a huge number of possibilities. Although small (11 mm) and sometimes problematic to image, the iris has the great mathematical advantage that its pattern variability among different persons is enormous. In addition, as an internal (yet externally visible) organ of the eye, the iris is well protected from the environment, and stable over time. As a planar object its image is relatively insensitive to angle of illumination, and changes in viewing angle cause only affine transformations; even the nonaffine pattern distortion caused by pupillary dilation is readily reversible. Finally, the ease of localizing eyes in faces, and the distinctive annular shape of the iris, facilitates reliable and precise isolation of this feature and the creation of a size-invariant representation.

Why, then, is iris recognition not more common? The answer seems simple: money. In order to implement iris recognition it is necessary to have a computer and an adjustable camera to accommodate people of differing stature. Obviously this is more costly on a large scale than encoded cards or memorized PIN's. However, as identity theft becomes more widespread and restricted access premises seek more powerful security solutions, iris recognition systems will become worth the cost.

References

[1] S. Sanderson, J. Erbetta. Authentication for secure environments based on iris scanning technology. *IEE Colloquium on Visual Biometrics*, 2000.

[2] J. Daugman. How iris recognition works. Proceedings of 2002 International Conference on Image Processing, Vol. 1, 2002.

[3] E. Wolff. *Anatomy of the Eye and Orbit*. 7th edition. H. K. Lewis & Co. LTD, 1976.

[4] R. Wildes. Iris recognition: an emerging biometric technology. *Proceedings of the IEEE*, Vol. 85, No. 9, 1997.

[5] J. Daugman. Biometric personal identification system based on iris analysis. United States Patent, Patent Number: 5,291,560, 1994.

[6] J. Daugman. 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.

[7] R. Wildes, J. Asmuth, G. Green, S. Hsu, R. Kolczynski, J. Matey, S. McBride. A system for automated iris recognition. *Proceedings IEEE Workshop on Applications of Computer Vision*, Sarasota, FL, pp. 121-128, 1994.

[8] W. Boles, B. Boashash. A human identification technique using images of the iris and wavelet transform. *IEEE Transactions on Signal Processing*, Vol. 46, No. 4, 1998.

[9] S. Lim, K. Lee, O. Byeon, T. Kim. Efficient iris recognition through improvement of feature vector and classifier. *ETRI Journal*, Vol. 23, No. 2, Korea, 2001.

[10] S. Noh, K. Pae, C. Lee, J. Kim. Multiresolution independent component analysis for iris identification. *The 2002 International Technical Conference on Circuits/Systems, Computers and Communications*, Phuket, Thailand, 2002.

[11] Y. Zhu, T. Tan, Y. Wang. Biometric personal identification based on iris patterns. *Proceedings of the 15th International Conference on Pattern Recognition*, Spain, Vol. 2, 2000.

[12] C. Tisse, L. Martin, L. Torres, M. Robert. Person identification technique using human iris recognition. *International Conference on Vision Interface*, Canada, 2002.

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

[14] C. Barry, N. Ritter. *Database of 120 Greyscale Eye Images*. Lions Eye Institute, Perth Western Australia.

[15] 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, 2001.

[16] L. Ma, Y. Wang, T. Tan. Iris recognition using circular symmetric filters. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, 2002.

[17] 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.

[18] M. Kass, A. Witkin, D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision***,** 1987.

[19] N. Tun. *Recognising Iris Patterns for Person (or Individual) Identification*. Honours thesis. The University of Western Australia. 2002.

[20] D. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America*, 1987.

[21] P. Burt, E. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*. Vol. 31 No. 4. 1983.

[22] P. Kovesi. *MATLAB Functions for Computer Vision and Image Analysis*.Available at: http://www.cs.uwa.edu.au/~pk/Research/MatlabFns/index.html

[23] A. Oppenheim, J. Lim. The importance of phase in signals. *Proceedings of the IEEE 69*, 529-541, 1981.

[24] P. Burt, E. Adelson. The laplacian pyramid as a compact image code. *IEE Transactions on Communications*, Vol. COM-31, No. 4, 1983.

[25] J. Daugman. Biometric decision landscapes. *Technical Report No. TR482, University of Cambridge Computer Laboratory,* 2000.

[26] T. Lee. Image representation using 2D gabor wavelets. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, Vol. 18, No. 10, 1996.