Fisher Linear Discriminant Analysis Technique for Iris Recognition

MS COMPUTER SOFTWARE ENGINEERING

By

Qazi Emad 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

Fisher Linear Discriminant Analysis Technique for Iris Recognition

Abstract

With an increasing emphasis on security, automated personal identification and verification based on biometrics has been receiving extensive attention over the past decade. Iris recognition, as an emerging biometric recognition approach, is becoming a very active topic in both research and practical applications. The motivation for this endeavor stems from the observation that the human iris provides a particularly interesting structure on which to base a technology for noninvasive biometric assessment. In particular, the biomedical literature suggests that irises are as distinct as fingerprints or patterns of retinal blood vessels. Further, since the iris is an overt body, its appearance is amenable to remote examination with the aid of a machine vision system. In general, a typical iris recognition system includes iris imaging, iris liveness detection, and recognition.

This research work deals with the implementation of iris recognition using Fisher Linear Discriminant Analysis and PCA method on iris images. Our projection method is based on Fisher's Linear Discriminant and produces well separated classes in a low-dimensional subspace. The proposed system contains four parts i.e. preprocessing, segmentation, feature extraction and matching. The preprocessing part further contains pupil localization, image refinement, iris localization and normalization processes.

First of all, we acquire the grayscale eye image. In second step, the radius and centre coordinates of the pupil is calculated. For this purpose, apply the thresholding with the value zero in order to make the pupil appear alone. we calculate the longest black row horizontally, and the longest column vertically, by counting the number of black points in each row and in each column as in figure, then find out the coordinate of the center easily and also the radius.

In third step, we find the threshold value (T) Using Basic Global Thresholding algorithm. Implement the thresholding with T from previous algorithm on original image.

In fourth step, complement of the image is taken and applied the median filter for image refinement. In the fifth step, applying sobel edge detection on the refined image to detect the edges in the image.

In the sixth step, after the edge detection of the image, we apply the Hough transform circle algorithm to find the centre coordinates and radius of the iris.

The iris is the area between outer circle and inner circle. We got the pixels of this area and colored the background with black. So we cut the iris region in donut like shape. Then we whiten the non-iris region and display the iris region in the image only.

Typical next step is to "normalize" the iris to a rectangular representation. 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.

In the next step, we create the database of the normalized images. Read gray normalized images one by one and convert it into 1-D image and save it into associative memory.

In the 11th step, feature extraction process takes over in which firstly build the projection matrix by projecting central image onto eigen space by using Principal Component Analysis and this matrix becomes an input to Fisher Linear Discriminant Analysis Method to build Fisher projection matrix by projecting images onto Fisher linear space. 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.

In the final 12th step, when the iris identification is required then the test image is compared with the feature templates that are stored in database and calculate the Euclidean distance between the test images and stored images. The image with the

minimum Euclidean distance compares with threshold value that describes the identification of test image either it exists in the database or not.

The proposed algorithm was tested on CASIA database. The empirical results provide the accuracy of 97% with time delay of 0.01064 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. The robustness and time efficiency of the proposed algorithm makes it perfect system for real time applications.

Table of Contents

List of Figures

Chapter 1: Introduction

1. Biometrics

A biometric system provides automatic recognition of an individual based on some unique characteristic owned by the individual. Biometric systems have been developed based on fingerprints, facial features, hand geometry, and the one presented in this report, the iris.

Biometric systems work based on capturing sample records, such as face image, eye image, hand shape, finger shape. These sample records are transformed using some mathematical function to build biometric training path. These training paths will provide some characteristic property for identification. Most of the biometric systems allow two operation modes, which are an enrollment mode to build database and identification mode to give access [1].

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.

Nowadays, biometric systems play very important role with a maximum security is required where bank, military base, prison, airport, laboratory, intelligence office and etc. In order to produce a good biometric system, probability of having same characteristic should be minimal for individuals, so that the feature of individual does not change over time. Also the system should easily capture images in order to provide efficient access.

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[2].

Human iris on the other hand as an internal organ of the eye and as well protected from the external environment, yet it is easily visible from within one meter of distance makes it a perfect biometric for an identification system with the ease of speed, reliability and automation.

In this report we produced a biometric system which is based on the iris recognition. In order to complete verification process we have used Fisher Linear Dicriminant (FLD) and Principal Component Analysis (PCA) Method to recognize a specified image. Fisher Linear Dicriminant (FLD) and Principal Component Analysis Method is one of the most successful techniques that have been used to recognize images. We have adapted FLD and PCA methodology to Iris Recognition.

1.1 Types of Biometric System

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 [3]. It is also important to note that biometric modalities are in varying stages of maturity. Some of the major modalities include;

1.1.1 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. One of the major and recent is face recognition through Neural Networks, on which this research work is based upon.

1.1.2 Finger Print Recognition

Recognition through fingerprints is a well-known biometrics technique. Because of there 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 [3].

1.1.3 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.

1.1.4 Iris Recognition

Iris recognition is the process of recognizing a person by analyzing the random pattern of the iris. 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.

Although the coloration and the structure of the iris is 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 irides 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 it can cause injury to human eye [2].

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 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 measures how heat dissipates off the face of an individual.

Chapter 2: Introduction to Iris Recognition

2.1 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 [4].

The iris has been historically recognized to possess characteristics unique to each individual. In the mid-1980s, two ophthalmologists--Drs. Leonard Flom and Aran Safir--proposed the concept that no two irises are alike. They researched and documented the potential of using the iris for identifying people and were awarded a patent in 1987. Soon after, the intricate and sophisticated algorithm that brought the concept to reality was developed by Dr. John Daugman [1] and patented in 1994. The original work and continued development have established Iridian's [2] iris recognition algorithm as the mathematically unrivaled means for authentication.

The technical performance capability of the iris recognition algorithm far surpasses that of any other biometric technology now available. Objective measures, such as a cross-over error rate, are at levels that cannot be reached by other biometrics. Iridian's algorithm is designed for rapid (seconds) exhaustive search of very large databases; a distinctive capability required for authentication today.

2.2 What is Iris?

The iris is the plainly visible, colored ring that surrounds the pupil. It is a muscular structure that controls the amount of light entering the eye, with intricate details that can be measured, such as striations, pits, and furrows. The iris is not to be confused with the retina, which lines the inside of the back of the eye. Fig.1 shows human eye characteristics. No two irises are alike. There is no detailed correlation between the iris patterns of even identical twins, or the right and left eye of an individual. The amount of information that can be measured in a single iris is much greater than fingerprints, and the accuracy is greater than DNA [5].

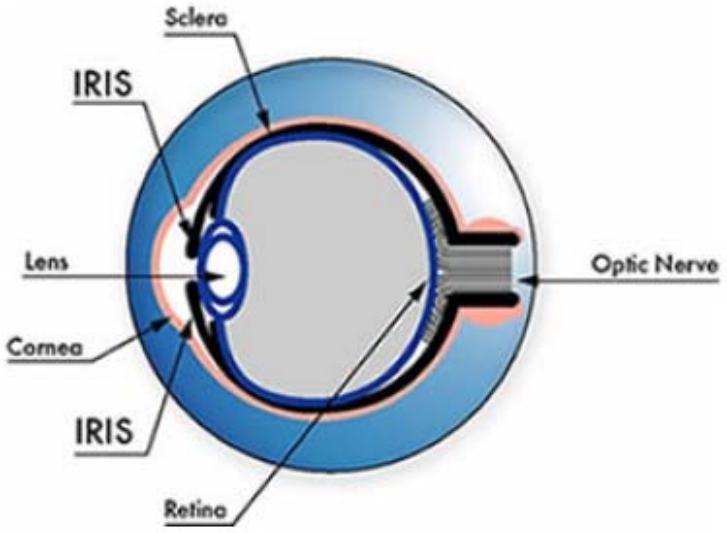


Figure 2.1: Human Eye

2.2.1 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 [2].

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 [3]. An outer ciliary zone and an inner pupillary zone, and these two zones are divided by the collarette – which appears as a zigzag pattern.

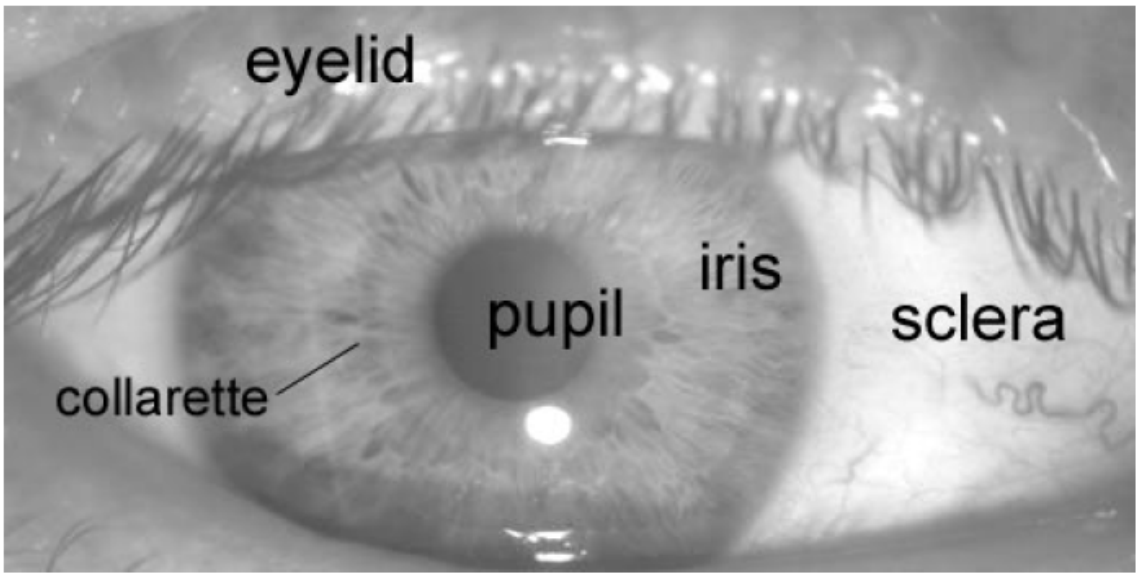


Figure 2.2: Details of Human Eye

Formation of the iris begins during the third month of embryonic life [3]. 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 [4]. 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 [3].

2.3 How Iris Recognition System Woks

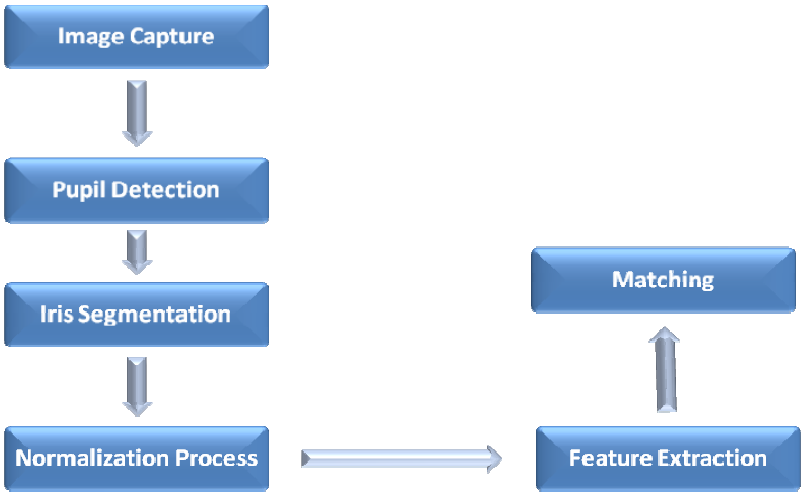


Figure 2.3: Working of Iris Recognition System

2.4 Process Diagram

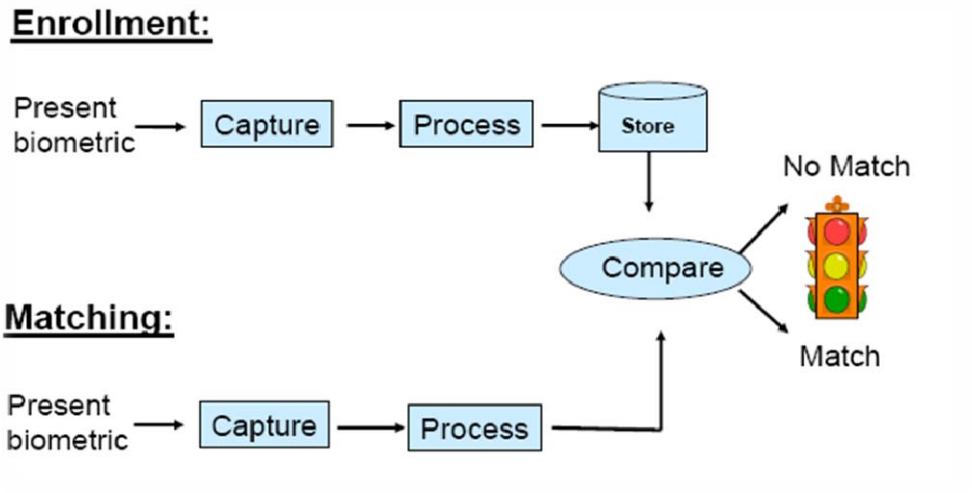


Figure 2.4: Processing layout of Iris Recognition System

2.5 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 1078 [5].

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 he 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.

2.6 Iris Recognition

The iris is an externally visible, yet protected organ whose unique epigenetic pattern remains stable throughout adult life. 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 [4].

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 [1][2]. The Daugman system is patented [5] 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. [7][4], Boles and Boashash [8], Lim et al. [9], 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 [7], 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 [1]. 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.

2.7 Objective

The objective will be to implement an open-source iris recognition system in order to verify the claimed performance of the technology. 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. To test the system, two data sets of eye images will be used

as inputs; a database of 756 grayscale eye images courtesy of The Chinese Academy of Sciences – Institute of Automation (CASIA).

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.

2.8 My Research work

In my research work the basic goal is of recognizing the Irises. This research work deals with the implementation of iris recognition using Fisher Linear Discriminant Analysis and PCA method on iris images. Our projection method is based on Fisher's Linear Discriminant and produces well separated classes in a low-dimensional subspace. The proposed system contains four parts i.e. preprocessing, segmentation, feature extraction and matching. The preprocessing part further contains pupil localization, image refinement, iris localization and normalization processes.

The proposed algorithm was tested on CASIA database. The empirical results provide the accuracy of 97% with time delay of 0.01064 sec per image. The robustness and time efficiency of the proposed algorithm makes it perfect system for real time applications.

2.8.1 Principal Component Analysis Method: An Introduction

Principal Components Analysis is a method that reduces data dimensionality by performing a covariance analysis between factors. As such, it is suitable for data sets in multiple dimensions, such as a large experiment in gene expression. Let’s take an example that illustrates how PCA works with a microarray experiment:

Say that you measure 10,000 genes in 8 different patients. These values could form a matrix of 8 x 10,000 measurements. Now imagine that each of these 10,000 genes is plotted in a multi-dimensional on a scatter plot consisting of 8 axes, 1 for each patient. The result is a cloud of values in multi-dimensional space [7].

To characterize the trends exhibited by this data, PCA extracts directions where the cloud is more extended. For instance, if the cloud is shaped like a football, the main direction of the data would be a midline or axis along the length of the football. This is called the first component, or the principal component. PCA will then look for the next direction, orthogonal to the first one, reducing the multidimensional cloud into a two-dimensional space. The second component would be the axis along the football width (Fig. 2.5).

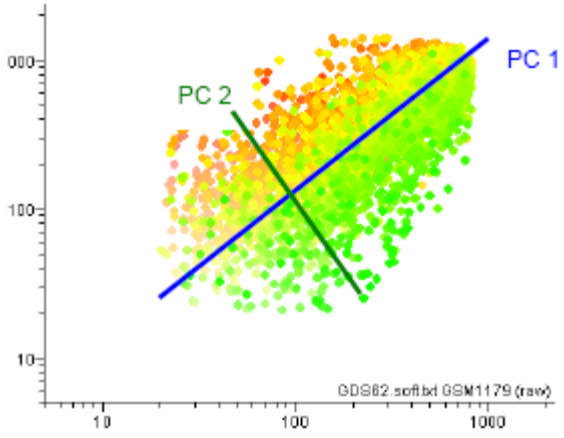


Figure 2.5: Football shaped data set with two main components

In this particular example, these two components explain most of the cloud's trends. In a more complex data set, more components might add information about interesting trends in the data. In GeneSpring, PCA can be performed based on gene expression profiles, or based on samples or conditions.

2.8.2 When to use Principal Components Analysis?

PCA is recommended as an exploratory tool to uncover unknown trends in the data. PCA on genes provide a way to identify predominant gene expression patterns. When applied on conditions, PCA will explore correlations between samples or conditions. Note that

because the goal of PCA is to 'summarize' the data, it is not considered a clustering tool. PCA does not attempt to group genes by user-specified criteria as does the clustering methods [8].

2.8.3 PCA Application to Computer Vision

PCA is used in computer vision, first showing how images are usually represented, and then showing what PCA can allow us to do with those images.

2.8.4 Representation

When using these sort of matrix techniques in computer vision, we must consider representation of images. A square N by N image can be expressed as an N2 dimensional vector

$$X = \begin{pmatrix} x_1 & x_2 & x_3 & . & . & x_{N^2} \end{pmatrix}$$

Where the rows of pixels in the image are placed one after the other to form a one dimensional image e.g. the first N elements (X1 – XN) will be the first row of the image; the next N elements are the next row, and so on. The values in the vector are the intensity values of the image, possibly a single grayscale value.

2.8.5 PCA to find patterns

Say we have 20 images. Each image is N pixels high by N pixels wide. For each image we can create an image vector as described in the representation section. We can then put all the images together in one big image-matrix like this:

$$ImagesMatrix = \begin{pmatrix} ImageVec1 \\ ImageVec2 \\ . \\ . \\ ImageVec20 \end{pmatrix}$$

Which gives us a starting point for our PCA analysis. Once we have performed PCA, we have our original data in terms of the eigenvectors we found from the covariance matrix. Why is this useful? Say we want to do Iris recognition, and so our original images were of people's irises. Then, the problem is, given a new image, whose iris from the original set is it? The way this is done is computer vision is to measure the difference between the new image and the original images, but not along the original axes, along the new axes derived from the PCA analysis [7].

It turns out that these axes works much better for recognizing irises, because the PCA analysis has given us the original images in terms of the differences and similarities between them. The PCA analysis has identified the statistical patterns in the data.

Since all the vectors are N2 dimensional, we will get N2 eigenvectors. In practice, we are able to leave out some of the less significant eigenvectors, and the recognition still performs well.

2.8.6 PCA for image compression

Using PCA for image compression also know as the Karhunen and Leove (KL), transform. If we have 20 images, each with N2 pixels, we can form N2 vectors, each with 20 dimensions. Each vector consists of all the intensity values from the same pixel from each picture. This is different from the previous example because before we had a vector for image, and each item in that vector was a different pixel, whereas now we have a vector for each pixel, and each item in the vector is from a different image. Now we perform the PCA on this set of data. We will get 20 eigenvectors because each vector is 20-dimensional. To compress the data, we can then choose to transform the data only using, say 15 of the eigenvectors. This gives us a final data set with only 15 dimensions, which has saved us ¼ of the space. However, when the original data is reproduced, the images have lost some of the information [8].

2.8.7 Fisher Linear Discriminant Method: An Introduction

Originally developed in 1936 by R.A. Fisher, Discriminant Analysis is a classic method of classification that has stood the test of time. Discriminant analysis often produces models whose accuracy approaches (and occasionally exceeds) more complex modern methods.

Discriminant analysis can be used only for classification (i.e., with a categorical target variable), not for regression. The target variable may have two or more categories.
To explain discriminant analysis, let's consider a classification involving two target categories and two predictor variables. The following figure 2.6 shows a plot of the two categories with the two predictors on orthogonal axes:

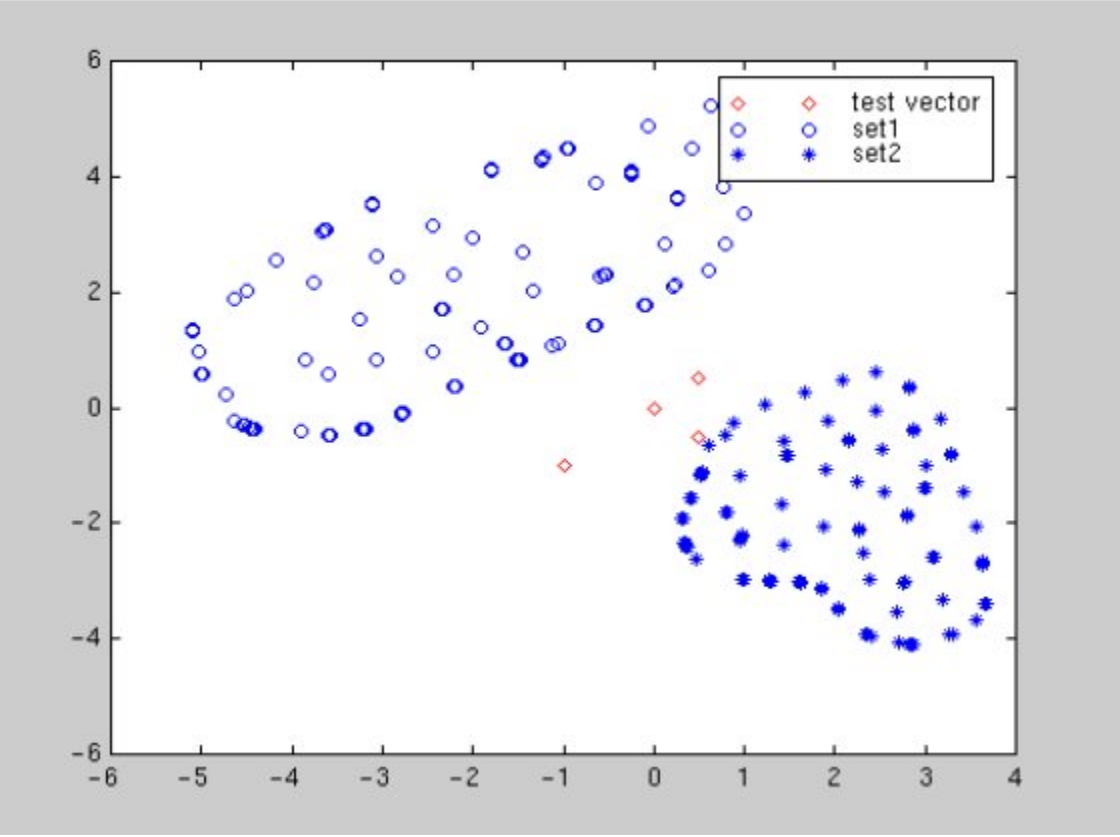


Figure 2.6: Plot of the two categories with the two predictors

A visual inspection shows that category 1 objects (open circles) tend to have larger values of the predictor on the Y axis and smaller values on the X axis. However, there is overlap

between the target categories on both axes, so we can't perform an accurate classification using only one of the predictors.

Linear discriminant analysis finds a linear transformation ("discriminant function") of the two predictors, X and Y, that yields a new set of transformed values that provides a more accurate discrimination than either predictor alone:

Transformed Target = C1*X + C2*Y

The following figure 2.7 shows the partitioning done using the transformation function:

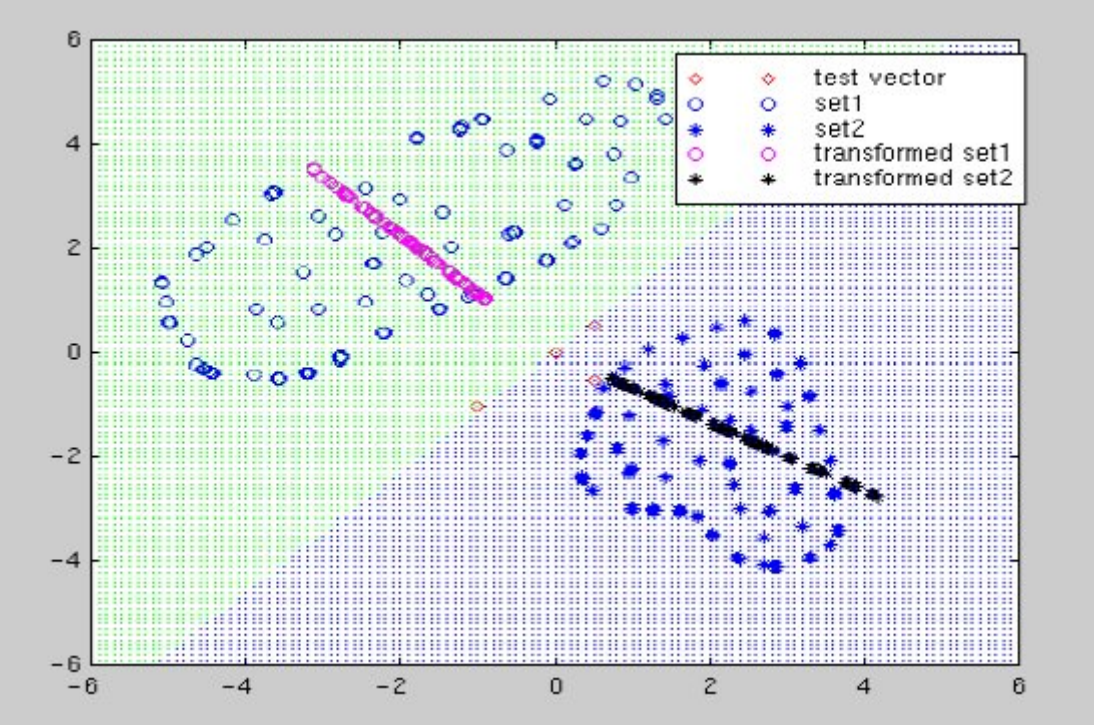


Figure 2.7: Partitioning using the transformation function

A transformation function is found that maximizes the ratio of between-class variance to within-class variance as illustrated by this figure 2.8:

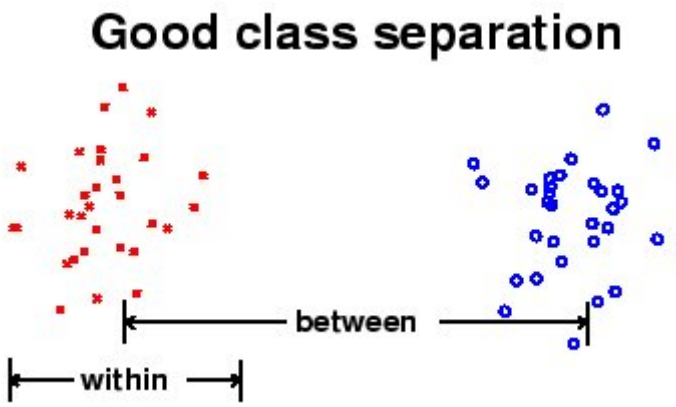


Figure 2.8: Between-class variance to within-class variance

The transformation seeks to rotate the axes so that when the categories are projected on the new axes, the differences between the groups are maximized. The following figure shows two rotates axes. Projection to the lower right axis achieves the maximum separation between the categories; projection to the lower left axis yields the worst separation.

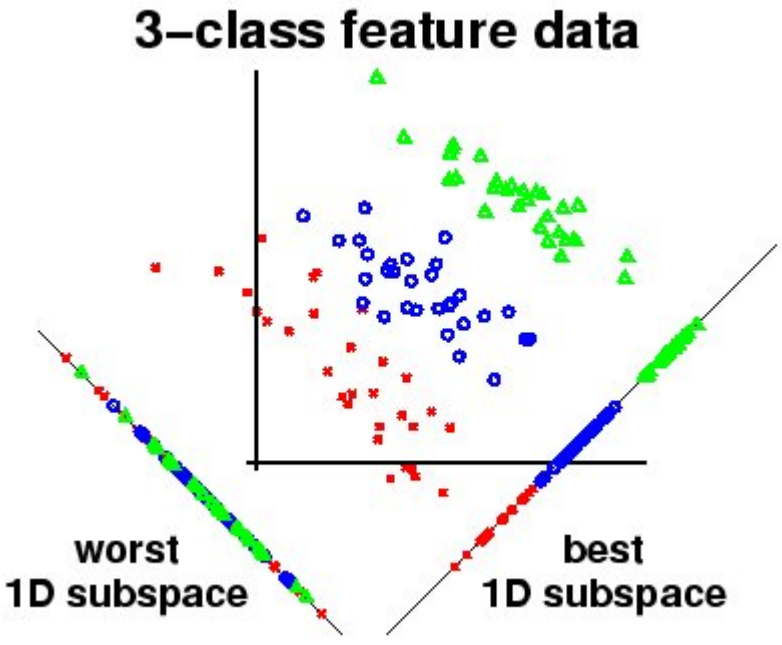


Figure 2.9: Two rotates axes

The following figure 2.10 illustrates a distribution projected on a transformed axis. Note that the projected values produce complete separation on the transformed axis, whereas there is overlap on both the original X and Y axes [10].

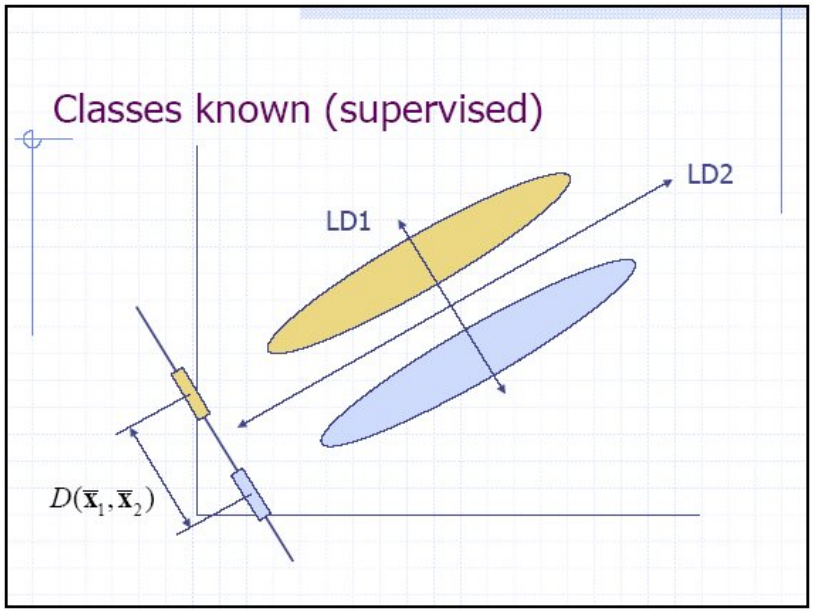


Figure 2.10: Distribution projected on a transformed axis

In the ideal case, a projection can be found that completely separates the categories (such as shown above). However, in most cases there is no transformation that provides complete separation, so the goal is to find the transformation that minimizes the overlap of the transformed distributions. The following figure 2.11 illustrates a distribution of two categories ("switch" in blue and "non-switch" in red). The black line shows the optimal axis found by linear discriminant analysis that maximizes the separation between the groups when they are projected on the line.

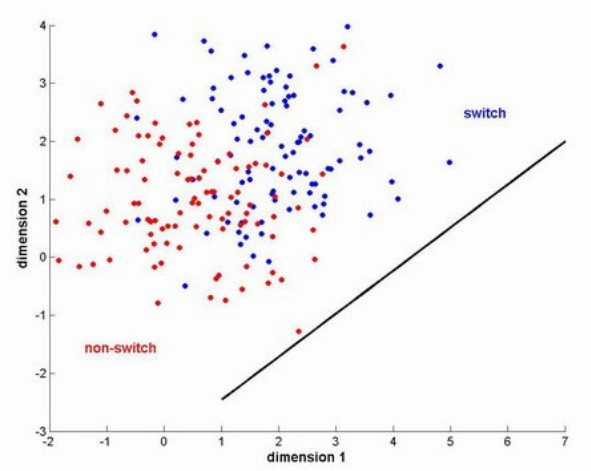


Figure 2.11: Transformation Distribution

The following figure 2.12 shows the distribution of the switch and non-switch categories as projected on the transformed axis (i.e., the black line shown in the figure above):

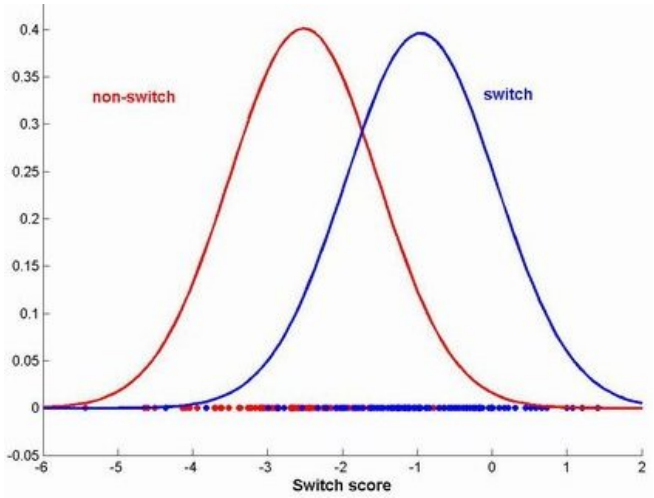


Figure 2.12: Distribution of the switch and non-switch categories

2.9 How Different/better

The main difference by which my Iris recognition task is different from other recognition algorithms is that it incorporates a Fisher Linear Discriminant Analysis Method [10] for the purpose of recognition and by use of this approach many useful advantages are there as it is computationally very efficient. It's approximately 0.01064 seconds per image.

Then in pre processing stage, I have used the efficient algorithms that reduced the image segmentation time and recognition task speeds up.

So all these techniques one after the other makes this research work a different from previously used techniques/algorithms. With respect to results even being done using Fisher Linear Discriminant Analysis Method, it perform at high percentages. So its much better than previously used techniques and my research work results are compared with other algorithms results in the results and discussion chapter.

2.10 Why use FLD Analysis Method?

Fisher Linear Discriminant Analysis Method, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques.
A transformation function is found that maximizes the ratio of between-class variance to within-class variance.
This method applies linear projection to the original image space to achieve dimensionality reduction. The projected space can be seen as a feature space where each component is seen as a feature. In other terms, PCA tries to find projection vectors where the variation among the projected samples is the greatest. FLD Analysis takes into account that the variation within the same class (in this case, identity) should be minimized and the variation between classes should be maximized [11].

Chapter 3: Biometric Techniques Comparison

3.1 Iris Recognition vs. Facial Recognition

Facial recognition technology has gained publicity to scan large crowds and populations. However, facial recognition is relatively easy to fool. Age, facial hair, surgery, head coverings, and masks all affect results. For this reason, it will most likely remain a surveillance tool instead of a starting line identifier, and will not be used for critical match applications such as border control or restricted access [12].

3.2 Weaknesses of Facial Recognition

Although facial recognition has success to verification, lighting, age, glasses, hair and beard shape and face covering masks may change verification success rate. Lower success rate may occur for large populations. As a result, secondary possessing is required for surveillance operations. Also people do not know always their pictures is being taken and searched for database or picture may be taken without permission of the user.

3.3 Iris Recognition vs. Fingerprint

Fingerprint technology has been widely accepted. Fingerprint readers are not ideally suited to handle the large variation of populations that need to be enrolled. “Outliers,” or those in the population that deviate from the average, can be a natural limit to enrollment or recognition. In large-scale deployments it takes many minutes, not seconds, to conduct a single search and a search may require ancillary data (such as age, sex, etc.) to partition the database for more speed. Further, multiple candidate matches may be returned. In a high volume, high speed environment, fingerprints do not have the accuracy, reliability or ability to handle large, diverse populations as is needed for critical transportation applications such as border control or restricted access. For these reasons, the best use of fingerprint technology within the transportation industry may be background checks [12].

3.4 Weakness of Fingerprint

Fingerprint is not accurate as iris recognition. False accepts rate may occur and is approximately 1 in 100,000. Iris recognition false accepts rate is 1 in 1.2 million statistically [12]. Most of the fingerprint systems measure approximately 40 – 60 characteristics but iris recognition looks about 240 characteristic to create eigenfaces. Iris recognition can perform matching in a high speed but fingerprint search take much longer and it may require filtering may give wrong identity matches. Most of the biometric systems required physical contact with scanner device and that needs to be kept clean and this is not hygiene issue. Iris recognition has a standard but on the other hand because of large number of different symmetry means no fingerprint standard. Fingerprint readability also may be affected by the work an individual does. For example, transportation workers such as mechanics, food workers, or maintenance workers may present fingerprints that are difficult to read due to dryness or the presence of foreign substances, such as oil or dirt, on fingers. Trauma or some disease on finger may change or damage finger details meaning that user may be needed scan his or her finger may times [12].

3.5 Iris Recognition vs. Hand Geometry

Hand Geometry is on of the first biometric verification system including access control, sale application, employee working time logs etc. Actually it is easy to use, also expensive, but needs large equipments which may limit the application. Hand geometry carries other challenges as well. Weather, temperature and medical conditions affect hand size. Hand size and geometry change greatly over time, which is especially obvious different in the very young and very old. It always needs upgrades for hand geometry. These challenges make hand geometry unsuitable for the high volume, large population applications of the transportation industry.

3.6 Weakness of Hand Geometry

Hand geometry not matches with large databases. Whether and medical conditions such

as pregnancy or certain medications can affect hand size. Hand size and geometry changes during the life cycle of people, especially in the very young and the very old. Most of the biometric systems required physical contact with scanner device and that needs to be kept clean and this is not hygiene issue. Big size of equipments may result difficulty in application. Expensive equipments are required.

3.7 Strengths of Iris Recognition

Iris recognition allow user to hands – free operation in application. Iris recognition has highest proven accuracy, had no false matches in over two million cross – comparison, according to Biometric Testing Final Report [12]. It allow high speed also for large populations, just look into a camera for a few seconds. The iris is stable for each individual through his or her life and do not change with age.

Chapter 4: Iris Recognition Technologies

This chapter will describe how iris recognition works, which steps the process consists of and how these steps can be implemented. First, a generalized structure for IR systems will be described, and then methods found in literature to implement the different parts of the IR process will be presented followed by a description of methods developed during the course of the thesis work.

4.1 The Iris Recognition process

A number of iris recognition systems have been studied ([17], [7], [25], [19], [18], [24], [15], [26], [14], [13]) and found to be very similar in structure. The IR pipeline in these systems can be described using these four steps:

1. Image acquisition. Obtaining an image of the subjects eye.
2. Localization and extraction of the iris part in the image.
3. Iris feature extraction. Extraction of discriminating properties of the iris, resulting in a unique iris signature, often called iris template or iris pattern.
4. Matching. Comparison of different templates for degree of match.

All four steps are as important for good recognition, but as the aim of this thesis is not to study the acquisition of eye images this section will focus on steps two to four. These steps can further be divided into the steps shown in figure below where step two is segmentation step three in normalization and feature extraction. All the steps shown in figure 4.1 below;

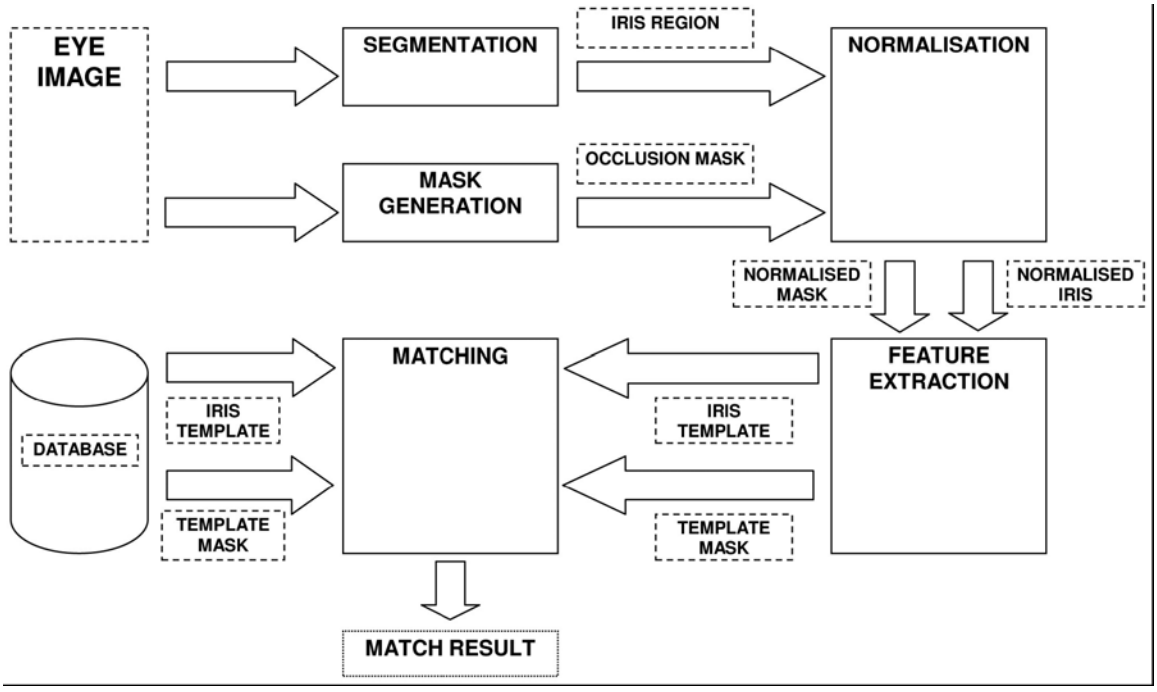


Figure 4.1: Schematic description of a generic IR system.

4.2 Segmentation

The objective of the segmentation step is to locate the iris region in the image. This consists of finding the inner boundary between the pupil and the iris and the outer boundary between the iris and the sclera. These boundaries, although not always perfectly circular, are modeled in most methods as two unconcentric circles, with the exception of the method purposed in [18] in which the boundaries are approximated as ellipses. A result of boundary fitting with a circular model is illustrated in Figure 4.2. Using these models the problem is then reduced to finding the circles or ellipses best describing the iris boundaries. A variety of methods has been proposed to do this and the most successful ones will be presented in this section.

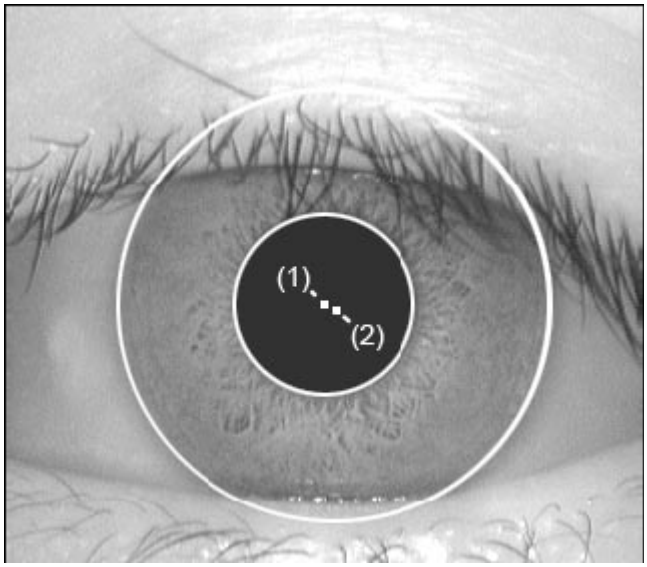


Figure 4.2: Iris segmentation using unconcentric circles.(1) Center of inner circle. (2) Center of outer circle

4.2.1 Hough transform based methods

Methods based on the Hough transform for circles, described in section 3.2.4, have been proposed by Tisse et al. [24], Ma et al.[15] and Wildes et al.[25]. The methods all apply an edge detector to the iris image followed by searching for the best inner and outer circular boundaries for the iris using the Hough transform. The edge detectors used vary between the methods, and in the case of the method proposed in [25] also between detection of the outer and inner boundaries of the iris. These methods all rely on good threshold parameters for the edge detection step, making them sensitive to varying imaging conditions.

4.2.2 Daugmans integro-differential operator

John Daugman presented in 1993 the segmentation method described in [7]. This method is based on his integro differential operator, defined in equation below which searches for the best fitting circles for the inner and outer boundaries of the iris. The operator is used twice, once for the inner boundary and once for the outer boundary searching iteratively for the best center coordinates (x0, y0) in the image I. This is done by looking for the

max value of the derivative in the radius dimension of the result of a circular contour integration. This search is performed iteratively from a coarse scale down to pixel level through convolution with a Gaussian kernel function [21], G(r) with decreasing size.

$$max_{(r,x_0,y_0)} \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r,x_0,y_0} \frac{I(x,y)}{2\pi r} ds \right|$$

This operator is frequently referred to in literature and different versions of this operator have been tested, for example in [18] in which a more general model was applied for the boundaries, modeling the iris as a rotated ellipse rather than a circle.

4.2.3 Methods based on thresholding

Segmentation methods based on the assumption that the iris can be separated from the eye because of the fact that the iris is generally lighter than the pupil and darker than the sclera have been proposed in Cho et al. [4], Zhu et al. [26], and Liam et al. [13]. The iris image is thresholded using an upper and lower intensity limits and then passed on to a circular edge detector. This approach aims to simplify for the edge detection step, which it does, but instead introduces the problem of finding good threshold levels.

4.3 Normalization

After segmentation has been completed, normalization is performed in all studied iris recognition systems to obtain invariance to iris size, position and different degrees of pupil dilation when matching different iris patterns at a later stage. The problem of that the iris can be rotated in the image is not resolved by this transformation, but is instead taken care of during matching. The method that is widely accepted for doing this is applying Daugmans rubber sheet model [7] to the iris and transforming it into a rectangular block. This transform to polar coordinates is defined in equations below and is illustrated in figure 4.3.

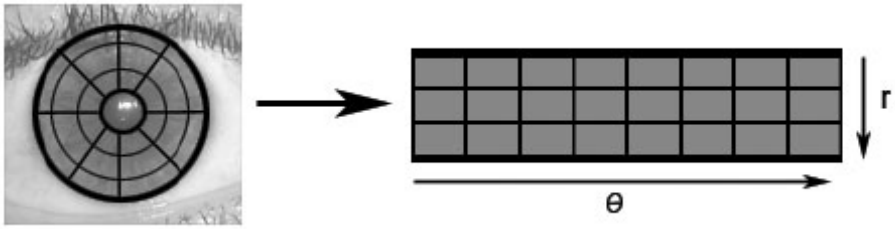


Figure 4.3: Illustration of normalization

$$I\left(x\left(r,\theta\right),y\left(r,\theta\right)\right)\rightarrow I\left(r,\theta\right)$$

$$x\left(r,\theta\right)=\left(1-r\right)x_{i}\left(\theta\right)+r\cdot x_{o}+cos(\theta)\cdot\left(r_{i}+r\cdot\left(r_{o}-r_{i}\right)\right)$$

$$y\left(r,\theta\right)=\left(1-r\right)y_{i}\left(\theta\right)+r\cdot y_{o}+sin(\theta)\cdot\left(r_{i}+r\cdot\left(r_{o}-r_{i}\right)\right)$$

4.4 Mask generation

The problems associated with occlusion of the iris by eyelids and eyelashes can be resolved by identifying these areas and marking them as occluded. The occluded areas can be marked as ones in a binary image, or mask, with the same size as the original. Different methods have been purposed to do this, Daugman [7] used a modified version of his integro-differential operator to detect the eyelids through searching for arcs instead of circles, but does not specify in detail how this is done. Masek [17] proposed to find the eyelids by locating lines by finding the maximum of a linear hough transformation of the image.

4.5 Encoding and matching

The encoding, or feature extraction, aims to extract as many discriminating features as possible from the iris and results in an iris signature, or template, containing these features. The matching process between two templates aims to maximize the probability of a true match for authentic identification tries and minimize false matches for

impostors. In other words, images of the same iris taken at different times should be identified as being from the same person and images from different irises should be marked as coming from different persons. In this section Daugmans methods for encoding and matching [7] will be presented in detail as this method was used in the testing along with an overview of other methods found in literature.

4.5.1 Daugmans method

Since 1993, when Daugman proposed the first successfully tested iris recognition system, many alternate methods have been proposed with the aim of improving recognition rate. However, only a few systems perform as good as Daugmans and none have been as thoroughly tested.

4.5.2 Encoding

Daugman applies a set of Gabor wavelets [6], to extract local phase, F(r, θ) information utilizing information around the base frequency !0 from the normalized iris I (r, ø) in N discrete positions along the iris according to equation below. The parameters α and β control the size of the wavelet. The resulting 2D phase information is quantizised into four levels according to the sign of the imaginary and real part of the result, generating a binary representation, h, of the features. If the feature information in (rn, θn) would be extracted as (1+í) this point ends up in the first quadrant in the imaginary plane and would be encoded as binary (11) and (1−í) would be encoded as binary (10) et cetera. This reduces the amount of information significantly, compressing the feature information to fit into standard magnetic cards.

$$F(r_n, \theta_n) = \int_s \int_\phi I\,(r_n, \theta_n) \cdot e^{-i\omega_0(\theta_n - \phi)} \cdot e^{-(r_n - s)^2/\alpha^2} \cdot e^{-(\theta_n - \phi)^2/\beta^2} d\phi ds$$

$$h_{(Re,Im)} = sgn_{\{Re,Im\}} F(r, \theta)$$

4.5.3 Matching

Matching is performed, using the normalized Hamming distance measure defined in equation below, taking into account the occluded regions defined as masks by not comparing the feature points extracted from these regions. The result is the number of bits that are different between the binary codes in the non-occluded regions, divided by the number of bits compared.

$$HD = \frac{\|codeA \otimes codeB \cap maskA \cap maskB\|}{\|maskA \cap maskB\|}$$

If the iris images would not be subject to noise and segmentation errors would not occur, the Hamming distance between two pictures taken of the same iris would be 0, however even at the most perfect conditions for imaging this is not the case. Daugman reports that the mean Hamming distance between two iris templates of the same iris, imagined at different occasions and under near perfect conditions, is 0.11 [7].

The theoretical Hamming distance between two different irises results in a Bernoulli trial, or coin toss test, Pp(m|N), where p is the probability that two bits are equal, m is the number of equal bits and N is the total number of bits compared. For uncorrelated iris codes p = 0.5. The fractional function associated with the binomial distribution is defined in equation 4.8 where x = N/m is the normalized hamming distance spanning from 0 to 1. Real life examples of these distributions can be seen in figure 5.11.

$$f(x) = \frac{N!}{m!(N-m)!}p^m(1-p)^{(N-m)}$$

Invariance of rotation of irises is resolved during matching by performing a series of matchings between rotated versions of the encoded irises and using the best match as the final result.

4.6 Other methods

A variety of methods have been purposed in literature and can be roughly categorized into correlation based methods and filter based methods. Correlation based methods utilizes a more direct texture matching approach in comparison to the filter based methods. This section will list a number of successful methods as a pointer to further reading on the subject.

4.6.1 Correlation based methods

Wildes et al. proposed in [25] a method based of normalized correlation of iris patterns, in which the iris images is divided up in laplacian pyramids which are then correlated separately for each individual level, resulting in a matching vector. Fishers linear discriminant [10] is then applied on this vector, resulting in a scalar equality score.

Another method based on correlation, is the one proposed by Miyazawa et. al. in [18]. This method applies band-phase correlation to discriminate between authentic and impostor matches.

4.6.2 Wavelet and filter based methods

To perform a wavelet transform or apply some sort of filter are the most common solutions for feature extraction. A variety of methods and filters have been proposed. Ma et al [15] purposed use of filters similar to [7] together with an enhanced matching method based on an extended version of Fishers linear discriminant. Lim et. al. purposed in [14] a method based on the Haar wavelet transform [22] and a matching algorithm based on a neural network.

Chapter 5: Principal Component Analysis (PCA)

Principal component analysis (PCA) has been called one of the most valuable results from applied linear algebra. PCA is used abundantly in all forms of analysis - from neuroscience to computer graphics - because it is a simple, non-parametric method of extracting relevant information from confusing data sets. With minimal additional effort PCA provides a roadmap for how to reduce a complex data set to a lower dimension to reveal the sometimes hidden, simplified structure that often underlie it.

Principal component analysis is appropriate when you have obtained measures on a number of observed variables and wish to develop a smaller number of artificial variables (called principal components) that will account for most of the variance in the observed variables. The principal components may then be used as predictor or criterion variables in subsequent analyses.

Principal Components Analysis is one of the best known and most used Multivariate Exploratory Analysis technique [17].

5.1 Goal of Principal Components Analysis

Given a data set described by a set of numerical variables {x1, x2 , ..., xp}, the goal of Principal Components Analysis is to describe this data set with a smaller set of new, synthetic variables. These variables will be linear combinations of the original variables, and are called Principal Components.
Quite generally, reducing the number of variables used to describe data will lead to some loss of information. PCA operates in a way that makes this loss minimal, in a sense that will be given a precise meaning.

Therefore, PCA may be regarded as a dimensionality reduction technique.

5.1.1 A Variable Reduction Procedure

Principal component analysis is a variable reduction procedure. It is useful when you have obtained data on a number of variables (possibly a large number of variables), and believe that there is some redundancy in those variables. In this case, redundancy means that some of the variables are correlated with one another, possibly because they are measuring the same construct. Because of this redundancy, you believe that it should be possible to reduce the observed variables into a smaller number of principal components (artificial variables) that will account for most of the variance in the observed variables.
Because it is a variable reduction procedure, principal component analysis is similar in many respects to exploratory factor analysis. In fact, the steps followed when conducting a principal component analysis are virtually identical to those followed when conducting an exploratory factor analysis. However, there are significant conceptual differences between the two procedures, and it is important that you do not mistakenly claim that you are performing factor analysis when you are actually performing principal component analysis. The differences between these two procedures are described in greater detail in a later section titled "Principal Component Analysis is Not Factor Analysis."

5.2 What is a Principal Component?

How principal components are computed. Technically, a principal component can be defined as a linear combination of optimally-weighted observed variables. In order to understand the meaning of this definition; it is necessary to first describe how subject scores on a principal component are computed. In the course of performing a principal component analysis, it is possible to calculate a score for each subject on a given principal component. For example, in the preceding study, each subject would have scores on two components: one score on the satisfaction with supervision component, and one score on the satisfaction with pay component. The subject's actual scores on the seven questionnaire items would be optimally weighted and then summed to compute their scores on a given component [19].

Below is the general form for the formula to compute scores on the first component extracted (created) in a principal component analysis:

C1 = b 11(X1) + b12(X 2) + ... b1p(Xp)

where

C1 = the subject's score on principal component 1 (the first component extracted)

b1p = the regression coefficient (or weight) for observed variable p, as used in creating principal component 1

Xp = the subject's score on observed variable p.

5.2.1 Properties of the Principal Components

Number

Although the ultimate goal is to use only a small number of Principal Components, PCA first identifies p such components, that is, the same number as the number of original variables. Only later will the analyst decide on the number of Components to be retained. "Retaining k Principal Components" means "Replacing the observations by their orthogonal projections in the k-dimensional subspace spanned by the first k Principal Components".

5.2.2 Orthogonality of the Principal Components

The Principal Components define orthogonal directions in the space of observations. In other words, PCA just makes a change of orthogonal reference frame, the new variables being replaced by the Principal Components [20].

5.2.3 Ordering the Principal Components, optimal projection subspaces

The fundamental property of the Principal Components is that they can be ordered by decreasing order of "importance" in the following sense:
If the analyst decides to describe the data with only k ($k < p$) linear combinations of the original variables and yet loose as little information as possible in the process, then these k linear combinations have to be the first k Principal Components. So the fundamental property of the PCs is that the best k-dimensional projection subspace is spanned by the first k Principal Components. In other words, the optimal subspaces are nested, a strong, useful and not at all obvious property.

5.3 Applications of Principal Components Analysis

5.3.1 Exploratory data analysis

PCA is mostly used for making 2-dimensional plots of the data for visual examination and interpretation. For this purpose, data is projected on factorial planes that are spanned by pairs of Principal Components chosen among the first ones (that is, the most significant ones).
From these plots, one will try to extract information about the data structure, such as:

- The detection of outliers (observations that are very different from the bulk of the data).
- The identification of clusters that suggest that several subpopulations might coexist within the data set.
- Interpretation of the Principal Components. Whereas the original variables have”native” interpretations, Principal Components derive from a mathematical definition. A successful PCA will allow interpreting the Principal Components in terms of realistic, if not measured, properties of the observations. When this is possible, it is sometimes said that PCA has revealed the existence of "latent variables".

5.3.2 Data preprocessing, dimensionality reduction

All multivariate modelisation techniques are prone to the bias-variance tradeoff, which states that the number of variables entering a model should be severely restricted. Data is often decribed by many more variables than necessary for building the best model. Sometimes, specific techniques exist for selecting a "good" subset of variables (see for instance Mutiple Linear Regression), but dimensionality reduction techniques such as PCA may also be considered for feeding the model with a reduced number of variables. For example, Multiple Linear Regression may be replaced by a model using only a reduced number of Principal Components as regressors (Principal Components Regression) [21].

5.4 PCA incorporated in other techniques

Although created from a downright applicative perspective (data visualization), the mathematical machinery of PCA is quite general and is at the heart of other important modelization techniques. Let's mention:

- Discriminant Factor Analysis, that can be regarded as a PCA on the ponderated class barycenters using the Mahalanobis distance.
- Ridge Regression that receives a very illuminating interpretation in terms of PCA.

5.5 Steps in Conducting Principal Component Analysis

Principal component analysis is normally conducted in a sequence of following steps;

5.5.1 Step 1: Initial Extraction of the Components

In principal component analysis, the number of components extracted is equal to the number of variables being analyzed. The first component can be expected to account for a fairly large amount of the total variance. Each succeeding component will account for progressively smaller amounts of variance. Although a large number of components may be extracted in this way, only the first few components will be important enough to be retained for interpretation [21].

5.5.2 Step 2: Determining the Number of "Meaningful" Components to Retain

Earlier it was stated that the number of components extracted is equal to the number of variables being analyzed, necessitating that decide just how many of these components are truly meaningful and worthy of being retained for rotation and interpretation. In general, expect that only the first few components will account for meaningful amounts of variance, and that the later components will tend to account for only trivial variance. The next step of the analysis, therefore, is to determine how many meaningful components should be retained for interpretation. This section will describe four criteria that may be used in making this decision:

A. The eigen value-one criterion. In principal component analysis, one of the most commonly used criteria for solving the number-of-components problem is the eigen value-one criterion.

B. The scree test. With the scree test, plot the eigen values associated with each component and look for a "break" between the components with relatively large eigen values and those with small eigen values. The components that appear before the break are assumed to be meaningful and are retained for rotation; those appearing after the break are assumed to be unimportant and are not retained.

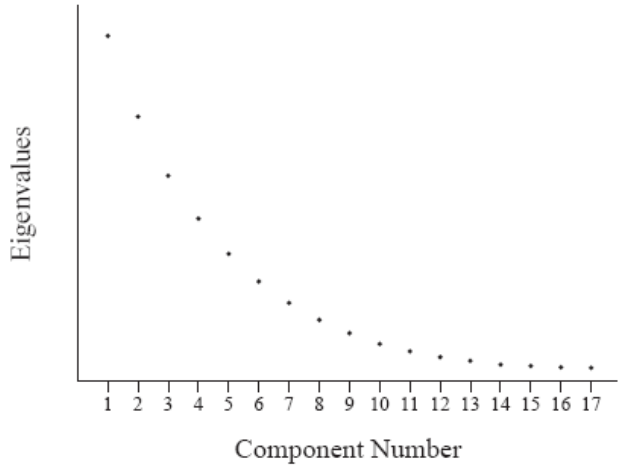


Figure 5.1: A Scree Plot with No Obvious Break

C. Proportion of variance accounted for. A third criterion in solving the number of factors problem involves retaining a component if it accounts for a specified proportion (or percentage) of variance in the data set. For example, you may decide to retain any component that accounts for at least 5% or 10% of the total variance. This proportion can be calculated with a simple formula [22]:

$$\text{Proportion} = \frac{\text{Eigenvalue for the component of interest}}{\text{Total eigenvalues of the correlation matrix}}$$

In principal component analysis, the "total eigenvalues of the correlation matrix" is equal to the total number of variables being analyzed (because each variable contributes one unit of variance to the analysis).

D. The interpretability criteria. Perhaps the most important criterion for solving the "number of - components" problem is the **interpretability criterion**: interpreting the substantive meaning of the retained components and verifying that this interpretation makes sense in terms of what is known about the constructs under investigation.

5.5.3 Step 3: Rotation to a Final Solution

Factor patterns and factor loadings. After extracting the initial components, PROC FACTOR will create an un-rotated factor pattern matrix. The rows of this matrix represent the variables being analyzed, and the columns represent the retained components (these components are referred to as FACTOR1, FACTOR2 and so forth in the output). The entries in the matrix are factor loadings. A factor loading is a general term for a coefficient that appears in a factor pattern matrix or a factor structure matrix. In an analysis that results in oblique (correlated) components [22], the definition for a factor loading is different depending on whether it is in a factor pattern matrix or in a factor structure matrix. However, the situation is simpler in an analysis that results in orthogonal components (as in the present chapter): In an orthogonal analysis, factor

loadings are equivalent to bivariate correlations between the observed variables and the components.

5.5.4 Step 4: Interpreting the Rotated Solution

Interpreting a rotated solution means determining just what is measured by each of the retained components. Briefly, this involves identifying the variables that demonstrate high loadings for a given component, and determining what these variables have in common. Usually, a brief name is assigned to each retained component that describes its content. The first decision to be made at this stage is to decide how large a factor loading must be to be considered "large." The following text provides a structured approach for interpreting this factor pattern.

1. Read across the row for the first variable.
2. Repeat this process for the remaining variables, scratching out any variable that loads on more than one component.
3. Review all of the surviving variables with high loadings on component 1 to determine the nature of this component.
4. Repeat this process to name the remaining retained components.
5. Determine whether this final solution satisfies the interpretability criteria.

5.5.5 Step 5: Creating Factor Scores or Factor-Based Scores

Once the analysis is complete, it is often desirable to assign scores to each subject to indicate where that subject stands on the retained components. For example, the two components retained in the present study were interpreted as a financial giving component and an acquaintance helping component. Now assign one score to each subject to indicate that subject's standing on the financial giving component, and a different score to indicate that subject's standing on the acquaintance helping component. With this done, these component scores could be used either as predictor variables or as criterion variables in subsequent analyses [23].

5.5.6 Step 6: Summarizing the Results

Now summarize the results of analysis, it is generally desirable to prepare a table that presents the rotated factor pattern.

5.6 Conclusion

Principal component analysis is a powerful tool for reducing a number of observed variables into a smaller number of artificial variables that account for most of the variance in the data set. It is particularly useful when you need a data reduction procedure that makes no assumptions concerning an underlying causal structure that is responsible for co-variation in the data. When it is possible to postulate the existence of such an underlying causal structure, it may be more appropriate to analyze the data using exploratory factor analysis.

Chapter 6: Fisher Linear Discriminant Analysis

6.1 Classification and variable selection

Dimensionality reduction is one of the central issues of statistical Data Modeling. Recall that the problem is to describe the data set with as few variables as possible, and yet discard as little information as possible in doing so.

* The "brute force" approach to dimensionality reduction is variable selection: of all the variables in the data table, only a handful are retained, that are hoped to capture most of the data structure. This approach is most common in Multiple Linear Regression.

* A more subtle approach is depicted in the following illustration. Suppose we want to build a classifier for the purpose of discriminating between two classes C1 and C2. Of the two variables x1 and x2 describing the data, which one should we retain in the final 1-variable-only-model.

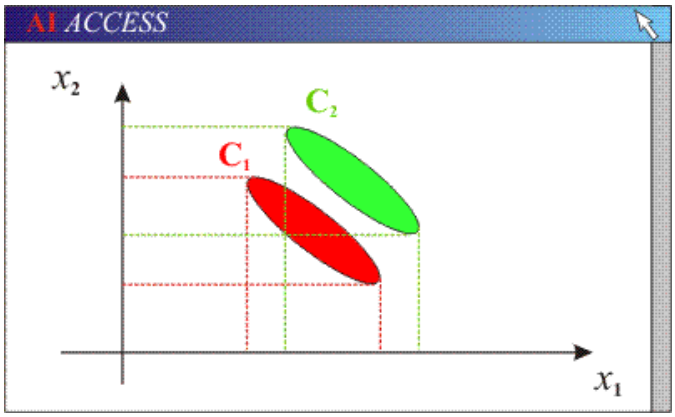


Figure 6.1: Discrimination between two classes

Suppose we decide to retain x1. This means that all the data points are projected on the x1 axis, or, equivalently, that their x2 coordinate is ignored. The classifier is just :

- A "threshold" value T_0,
- and an assignment rule :
  - "If $x_1 < T_0$, assign observation to $\mathbf{C}_1$,
  - else assign observation to $\mathbf{C}_2$."

Clearly this classifier is very poor because of the considerable overlap of the projections of $\mathbf{C}_1$ and $\mathbf{C}_2$ on x_1. Any threshold T_0 will be conducive to a high misclassification rate. Unfortunately, retaining x_2 and discarding x_1 is no better. So we are in a situation where any variable selection is worse than no variable selection at all, and dimensionality reduction seems impossible because too much information is lost in the process of eliminating any of the two original variables [24].

Yet, it is easy to build a perfect classifier (lower image in the illustration above): just project both classes on the blue line *F*. The two projected classes are then completely separated, and it's easy to find a threshold T_0 such that the above assignment rule gives a "0" misclassification rate.

Is this classifier a 1-variable or a 2-variable classifier?

* Both x_1 and x_2 are involved in defining *F*, so it seems that we have a 2-variable classifier.

* But if we denote by *x* the abscissa of the projection of a data point on *F* (with respect to an arbitrary origin), then considering the value of *x* alone is enough to have a perfect assignement rule. So we in fact have a 1-variable-only classifier, and this variable is *x*.

So we have found a data reduction technique more complex than variable selection, but also obviously more powerful. Fisher's linear discriminant is just a formalization of the above example [24].

6.2 Fisher's linear discriminant

Fisher's linear discriminant deals with:

* Two classes only C_1 and C_2,

* Described by an arbitrary large number of original variables x_1 , x_2 , ..., x_p,

* The goal being to identify the 1-Dimensional straight line *F* on which the class projections will be "as separated as possible".

Ideally, we would therefore like to define some measure of overlap between two sets of points, and find the projection direction that makes this measure minimal. Unfortunately, there is no simple and elegant measure of overlap between two sets of points, and we will have to be satisfied with defining a criterion that :

* gets larger with the distance between the projected class barycenters,

* gets larger when the projected classes become more "compact" according to some measure,

Hoping that if the class projections are compact and the projected barycenters are far from each other, then the class projections will exhibit little overlap.

In general, "distance between projected barycenters" and "class projections compacity" cannot be maximized simultaneously. Fisher's criterion J is a trade-off between these two incompatible objectives.

Fisher's linear discriminant F is the direction of space that makes Fisher's criterion maximum. It is generally unique.

Once it is found, it is then the practioner's responsibility to establish the value T0 of the threshold T that will minimize the misclassification rate, and this turns Fisher's discriminant into a complete classifier [24].

6.2.1 Optimality of Fisher's linear discriminant

Fisher's linear discriminant criterion is defined in a somewhat heuristic way, and therefore cannot claim to always define the best 1-D projection space in terms of minimizing the eventual misclassification rate. As a matter of fact, other criteria of "optimal class projections" may be defined and are occasionally used instead of Fisher's criterion.

Yet, in the special case where the two classes are multinormal with identical Covariance Matrices, it can be shown that Fisher's discriminant is indeed optimal. The classifier derived from it is then identical to the discriminant function obtained by the least-squares errors approach. The classification threshold is then also provided by theory, and does not have to be adjusted "by hand".

6.2.2 Fisher's linear discriminant and Discriminant Analysis

Fisher's linear discriminant is the first step on the way to Discriminant Factor Analysis.

- It can be generalized to more than just 2 classes.
- If the data sits in a *n*-dimensional space and there are *p* classes with $p < n$, then (p - 1) orthogonal 1-D projection lines may be defined. The first one is just the generalized Fisher's linear discriminant. The second one is defined as sitting in the (n - 1)-dimensional subspace orthogonal to it, and maximizing the generalized Fisher's criterion for the classes once projected in that subspace, a.s.o.. These are the discriminant factors.

The most famous example of dimensionality reduction is “principal components analysis”. This technique searches for directions in the data that have largest variance and subsequently project the data onto it. In this way, we obtain a lower dimensional representation of the data, that removes some of the ”noisy” directions. There are many difficult issues with how many directions one needs to choose, but that is beyond the scope of this note. PCA is an unsupervised technique and as such does not include label information of the data. For instance, if we imagine 2 cigar like clusters in 2 dimensions, one cigar has y = 1 and the other y = -1 [25]. The cigars are positioned in parallel and very closely together, such that the variance in the total data-set, ignoring the labels, is in the direction of the cigars. For classification, this would be a terrible projection, because all labels get evenly mixed and we destroy the useful information. A much more useful projection is orthogonal to the cigars, i.e. in the direction of least overall variance, which would perfectly separate the data-cases (obviously, we would still need to perform classification in this 1-D space). So the question is, how do we utilize the label information in finding informative projections? To that purpose Fisher-LDA considers maximizing the following objective:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}} \qquad (1)$$

where SB is the "between classes scatter matrix" and SW is the "within classes scatter matrix". Note that due to the fact that scatter matrices are proportional to the covariance matrices we could have defined J using covariance matrices – the proportionality constant would have no effect on the solution. The definitions of the scatter matrices are:

$$S_B = \sum_c N_c(\boldsymbol{\mu}_c - \bar{\mathbf{x}})(\boldsymbol{\mu}_c - \bar{\mathbf{x}})^T \quad (2)$$

$$S_W = \sum_c \sum_{i \in c} (\mathbf{x}_i - \boldsymbol{\mu}_c)(\mathbf{x}_i - \boldsymbol{\mu}_c)^T \quad (3)$$

where,

$$\mu_c = \frac{1}{N_c} \sum_{i \in c} x_i \quad (4)$$

$$\bar{x} = = \frac{1}{N} \sum_i x_i = \frac{1}{N} \sum_c N_c \mu_c \quad (5)$$

and Nc is the number of cases in class c. Oftentimes you will see that for 2 classes SB is defined as S'B = $(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$. This is the scatter of class 1 with respect to the scatter of class 2 and you can show that $S_B = \frac{N_1 N_2}{N} S'_B$, but since it boils down to multiplying the objective with a constant is makes no difference to the final solution. Why does this objective make sense. Well, it says that a good solution is one where the class-means are well separated, measured relative to the (sum of the) variances of the data assigned to a particular class. This is precisely what we want, because it implies that the gap between the classes is expected to be big. It is also interesting to observe that since the total scatter,

$$S_T = \sum_i (\mathbf{x}_i - \bar{x})(\mathbf{x}_i - \bar{x})^T \quad (6)$$

is given by ST = SW + SB the objective can be rewritten as,

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_T \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}} - 1 \qquad (7)$$

and hence can be interpreted as maximizing the total scatter of the data while minimizing the within scatter of the classes. An important property to notice about the objective J is that is is invariant w.r.t. re-scaling of the vectors $\mathbf{w} \rightarrow \alpha\mathbf{w}$. Hence, we can always choose w such that the denominator is simply $\mathbf{w}^T S_W \mathbf{w} = 1$, since it is a scalar itself. For this reason we can transform the problem of maximizing J into the following constrained optimization problem [25],

$$\min_{\mathbf{w}} \quad -\frac{1}{2}\mathbf{w}^T S_B \mathbf{w} \qquad (8)$$

$$\text{s.t.} \quad \mathbf{w}^T S_W \mathbf{w} = 1 \qquad (9)$$

corresponding to the lagrangian,

$$\mathcal{L}_P = -\frac{1}{2}\mathbf{w}^T S_B \mathbf{w} + \frac{1}{2}\lambda(\mathbf{w}^T S_W \mathbf{w} - 1) \qquad (10)$$

(the halves are added for convenience). The KKT conditions tell us that the following equation needs to hold at the solution,

$$S_B \mathbf{w} = \lambda S_W \mathbf{w} \quad \Rightarrow \quad S_W^{-1} S_B \mathbf{w} = \lambda \mathbf{w} \qquad (11)$$

This almost looks like an eigen-value equation, if the matrix $S_W^{-1} S_B$ would have been symmetric (in fact, it is called a generalized eigen-problem). However, we can apply the following transformation, using the fact that SB is symmetric positive definite and can hence be written as $S_B^{\frac{1}{2}} S_B^{\frac{1}{2}}$, where $S_B^{\frac{1}{2}}$ is constructed from its eigen value decomposition as

$S_B = U\Lambda U^T \rightarrow S_B^{\frac{1}{2}} = U\Lambda^{\frac{1}{2}}U^T$. Defining $\mathbf{v} = S_B^{\frac{1}{2}}\mathbf{w}$ we get,

$$S_B^{\frac{1}{2}} S_W^{-1} S_B^{\frac{1}{2}} \mathbf{v} = \lambda \mathbf{v} \tag{12}$$

This problem is a regular eigen value problem for a symmetric, positive definite matrix $S_B^{\frac{1}{2}} S_W^{-1} S_B^{\frac{1}{2}}$ and for which we can find solution λ_k and $\mathbf{v}_k$ that would correspond to Solutions $\mathbf{w}_k = S_B^{-\frac{1}{2}}\mathbf{v}_k$.

Remains to choose which eigenvalue and eigenvector corresponds to the desired solution. Plugging the solution back into the objective J, we find,

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}} = \lambda_k \frac{\mathbf{w}_k^T S_W \mathbf{w}_k}{\mathbf{w}_k^T S_W \mathbf{w}_k} = \lambda_k \tag{13}$$

from which it immediately follows that we want the largest eigen value to maximize the objective.

Chapter 7: Iris Recognition Pre-Processing

7.1 Iris Detection

Iris detection is one of the most accurate and secures means of biometric identification while also being one of the least invasive. Fingerprints of a person can be faked--dead people can come to life by using a severed thumb. Thief's can don a nifty mask to fool a simple face recognition program. The iris has many properties which make it the ideal biometric recognition component.

The iris has the unique characteristic of very little variation over a life's period yet a multitude of variation between individuals. Irises not only differ between identical twins, but also between the left and right eye. Because of the hundreds of degrees of freedom the iris gives and the ability to accurately measure the textured iris, the false accept probability can be estimated at 1 in 10^31. Another characteristic which makes the iris difficult to fake is its responsive nature. Comparisons of measurements taken a few seconds apart will detect a change in iris area if the light is adjusted--whereas a contact lens or picture will exhibit zero change and flag a false input.

7.2 Iris Recognition: Detecting the Pupil

7.2.1 Acquiring the Picture

Beginning with a 320x280 pixel photograph of the eye taken from 4 centimeters away using a near infrared camera. The near infrared spectrum emphasizes the texture patterns of the iris making the measurements taken during iris recognition more precise. All images tested in this program were taken from the Chinese Academy of Sciences Institute of Automation (CASIA) iris database.

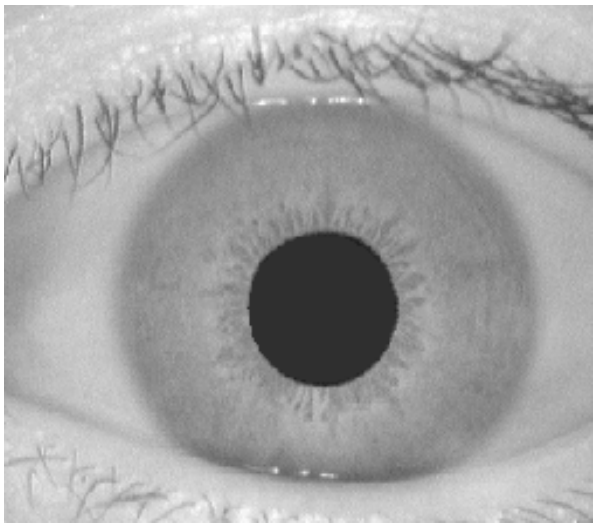


Figure 7.1: Near-infrared image of eye from CASIA Database

7.2.2 Edge Detection

Since the picture was acquired using an infrared camera the pupil is a very distinct black circle. The pupil is in fact so black relative to everything else in the picture a simple edge detection should be able to find its outside edge very easily. Furthermore, the thresholding on the edge detection can be set very high as to ignore smaller less contrasting edges while still being able to retrieve the entire perimeter of the pupil.

The best edge detection algorithm for outlining the pupil is sobel edge detection. This algorithm uses horizontal and vertical gradients in order to deduce edges in the image. After running the sobel edge detection on the image a circle is clearly present along the pupil boundary.

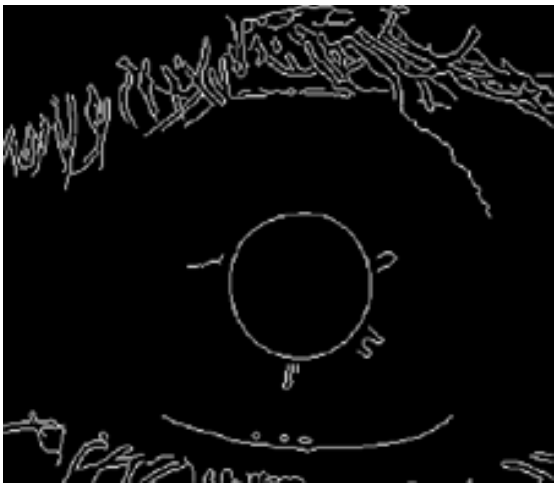


Figure 7.2: sobel edge detected image of the eye

7.2.3 Image Clean Up

A variety of other filters can be used in order decrease the extraneous data found in the edge detection stage. The first step in cleaning up the image is to dilate all the edge detected lines. By increasing the size of the lines nearby edge detected components are likely to coalesce into a larger line segment. In this way complete edges not fully linked by the edge detector can form. Thus the dilation will give us a higher probability that the perimeter of the pupil is a complete circle.

Knowing that the pupil is well defined more filters can be used without fear of throwing out that important information. Assuming the image is centered a filter can be used to fill in the circle defined by the pupil's perimeter. In this way we clearly define the entire area of the pupil. After this, a filter which simply throws out sections of connected pixels with an area below a threshold can be used effectively to throw out smaller disconnected parts of the image the edge detector found. Finally, any holes in the pupil caused by reflections or other distortions can be filled, by looking for sections of blank pixels with an area below a threshold. After this processing we achieve a picture that highlights the pupil area while being fairly clean of extraneous data.

Figure 7.3: Image after final filters

7.2.4 Pupil Information Extraction

Having pre-processed the image sufficiently the extraction of the pupil center and radius can begin. By computing the euclidean distance from any non-zero point to the nearest zero valued point an overall spectrum can be found. This spectrum shows the largest filled circle that can be formed within a set of pixels. Since the pupil is the largest filled circle in the image the overall intensity of this spectrum will peak in it.

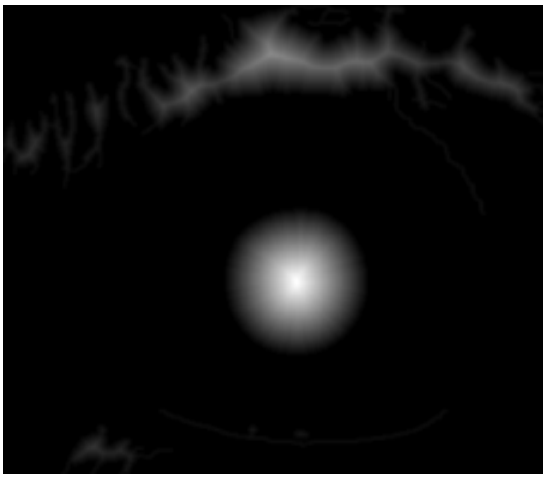


Figure 7.4: Image after computing minimal euclidean distance to non-white pixel.

In the pupil circle the exact center will have the highest value. This is due to the simple fact that the center is the one point inside the circle that is farthest from the edges of the circle. Thus the maximum value must correspond to the pupil center, and furthermore the

value at that maximum (distance from that point to nearest non-zero) must be equal to the pupil radius.

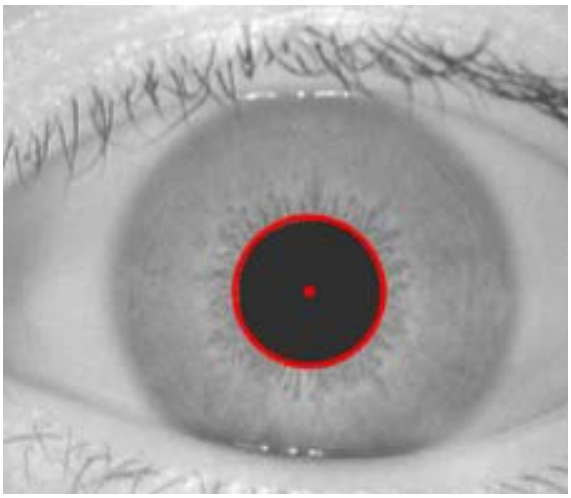


Figure 7.5: The original image of the eye with the pupil center and perimeter, found with the algorithm, highlighted

7.3 Iris Recognition: Detecting the Iris

7.3.1 Iris Detection

With the information on the pupil discovered the location of the iris can now begin. It is important to note that the pupil and iris are not concentric. Consequently, the pupil information does not help directly determine the same parameters for the iris. However, the pupil information does give a good starting point, the pupil center.

Most modern iris detection algorithms use random circles to determine the iris parameters. Having a starting point at the pupil, these algorithms guess potential iris centers and radii. They then integrate over the circumference in order to determine if it is on the border of the iris. While this is highly accurate the process can consume a lot of time. This module explains an alternate approach which is much more lightweight but with higher error rates.

7.3.2 Iris Radius Approximation

The first step in finding the actual iris radius is to find an approximation of the iris radius. This approximation can then be fine tuned to find the actual iris parameters. In order to find this approximation a single edge of the iris must be found. Knowing that eyes are most likely to be distorted in the top and bottom parts due to eyelashes and eyelids, the best choice for finding an unobstructed edge is along the horizontal line through the pupil center.

Having decided on where to attempt to detect the iris edge, the question of how to do it arises. It seems obvious that some type of edge detection should be used. It happens that for any edge detection it is a good idea to blur the image to subtract any noise prior to running the algorithm, but too much blurring can dilate the boundaries of an edge, or make it very difficult to detect. Consequently, a special smoothing filter such as the median filter should be used on the original image. This type of eliminates sparse noise while preserving image boundaries. The image may need to have its contrast increased after the median filter.

Subfigure A Subfigure B

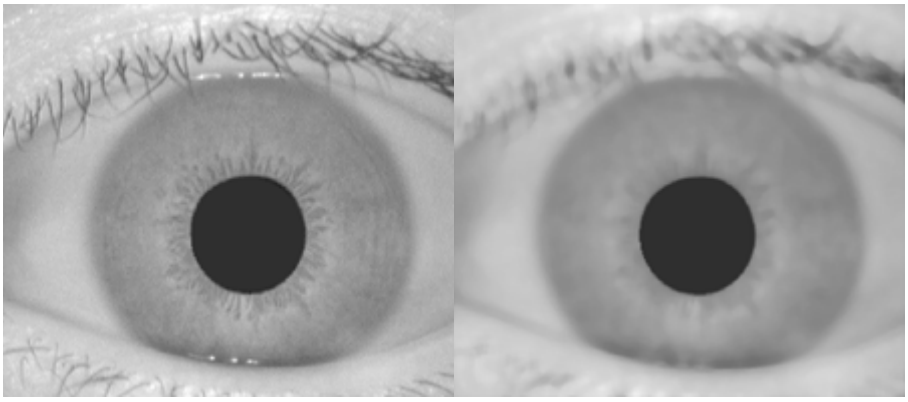
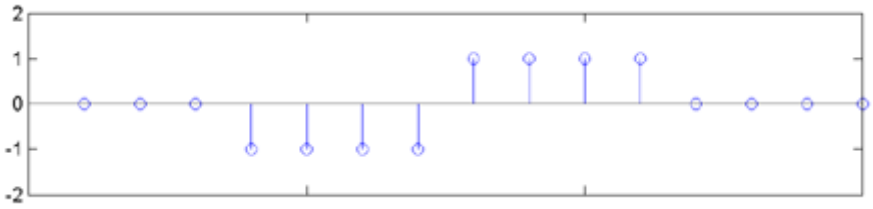


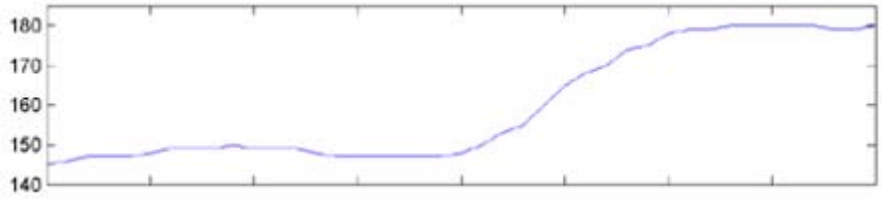
Figure 7.6: The original image after running through a median filter. A median filter works by assigning to a pixel the median value of its neighbors.

Now that the image is prepped the edge detection can be done. Since there is such a noticeable rising edge in luminescence at the edge of the iris, filtering with a haar wavelet should act as a simple edge detector. The area of interest is not just the single horizontal line through the iris, but the portion of that line to the left of the pupil. This is so that the rising luminescence from the transition from iris to white is the only major step.

Subfigure A: Haar Wavelet



Subfigure B: The area of interest



Subfigure C: The area of interest after filtering with the haar wavelet

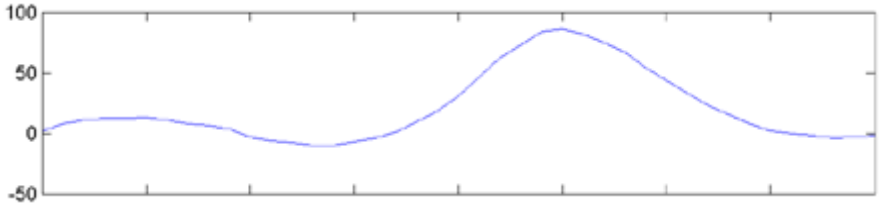


Figure 7.7: By filtering the area of interest with a haar wavelet all rises in luminance are transformed into high valued components of the output. The sharpness of change in luminance affects the overall height of the component.

The iris should represent the steepest luminance change in the area of interest. Consequently, this area of the image should correspond to the highest valued component of the output from the filter. By finding this maximal value the edge of the iris to the right of the pupil should be found. It should be noted that since the iris may not be concentric

with the pupil the distance from the pupil center to this edge may not correspond to the iris' radius.

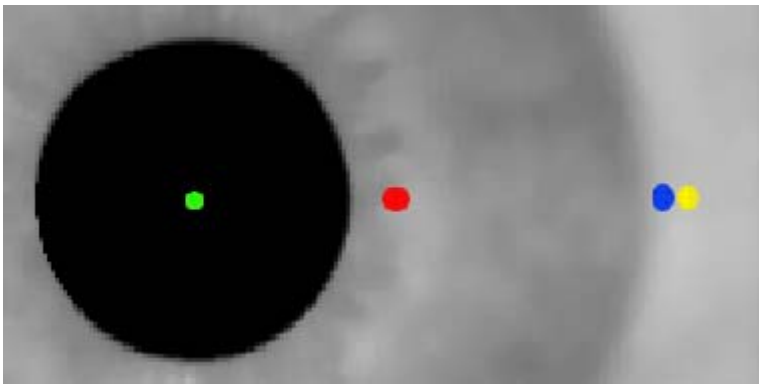


Figure 7.8: The green point is the pupil center found using the pupil detection techniques. The red point indicates the starting point of the area of interest. The blue point is the approximate radius found. The yellow point is the padded radius for use in finding the actual iris parameters.

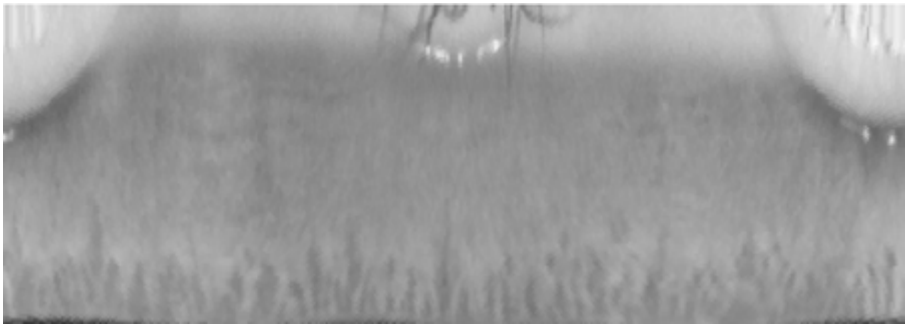
7.3.3 Iris Translation

Having acquired an approximate radius, a small pad of this value should produce a circle centered on the pupil which contains the entire iris. Furthermore, with the perimeter of the pupil known, an annulus may be formed which should have the majority of it's area filled by the iris. This annulus can then be unrolled into cartestian coordinates through a straight discretized transformation. (r --> y, θ --> x) The details of this procedure are described in Step 3.

If the iris is perfectly centered on the pupil, the unrolled image should have a perfectly straight line along its top. However, if the iris is off center even a little this line will be wavy. The line represents the overall distance the iris is at from the pupil center. It is this line which will help to determine the iris' center and radius. Consequently, an edge detection algorithm must be run on the strip in order to determine the line's exact location. Once again canny edge detection is used. However, before the edge detection can be run the image should undergo some simple pre-processing to increase the contrast

of the line. This will allow for a higher thresholding on the edge detection to eliminate extraneous data.

Subfigure A



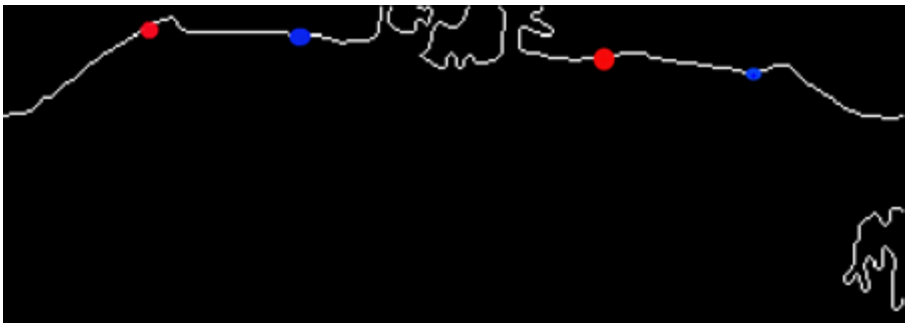
Subfigure B

Figure 7.9: The unrolled iris before and after edge detection

7.3.4 Iris Information Extraction

In order to extrapolate the iris' center and radius, two chords of the actual iris through the pupil must be found. This can be easily accomplished with the information gained in the previous step. By examing points with x values on the strip offset by half of the length of the strip a chord of the iris is formed through the pupil center. It is important to pick the vectors for these chords so they are both maximally independent of each other, while also being far from areas where eyelids or eyelashes may interfere.

Subfigure A



Subfigure B

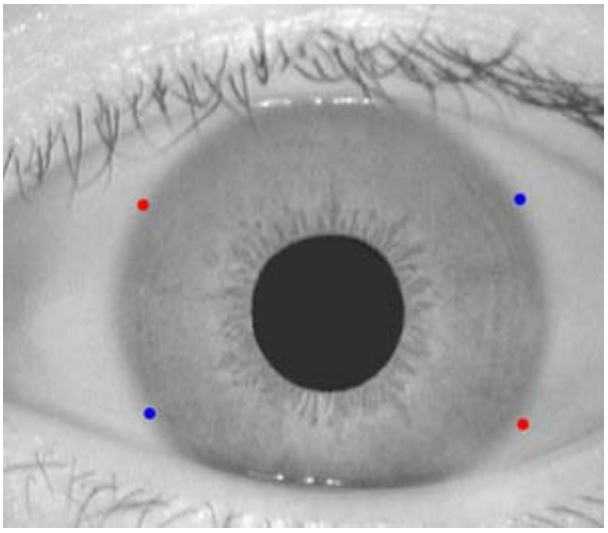


Figure 7.10: The points selected on the strip to form the chords of the iris through the pupil

The center of the iris can be computed by examining the shift vectors of the chords. By looking at both sides of a chord and comparing their lengths an offset can be computed. If the center was shifted by this vector it would equalize the two components of the chord. By doing this with both of the chords two different offset vectors can be computed. However, by just shifting the center through both of these vectors some components of shift will be overcompensated for due to the vectors not being orthogonal. Thus, the center should be shifted through the first vector and the orthogonal component of the second to the first.

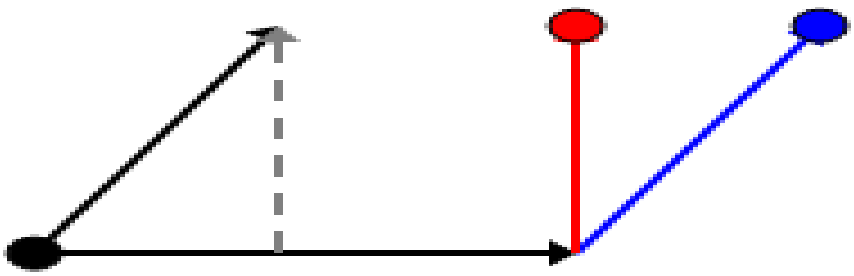


Figure 7.11: The change vectors (black) represent the shift of the pupil (black circle) in order to find the iris center. By just adding the vectors (blue vector) the result (blue circle) is offset by any vector the two change vectors share. Consequently, by adding the orthogonal component (gray vector) of one vector to the other (red vector), the actual iris center (red circle) is found.

The diameter of the iris can now be estimated by simply averaging the two diameters of the chords. While this is not a perfect estimate, that would require a single chord through the iris center, it is a very good approximation.

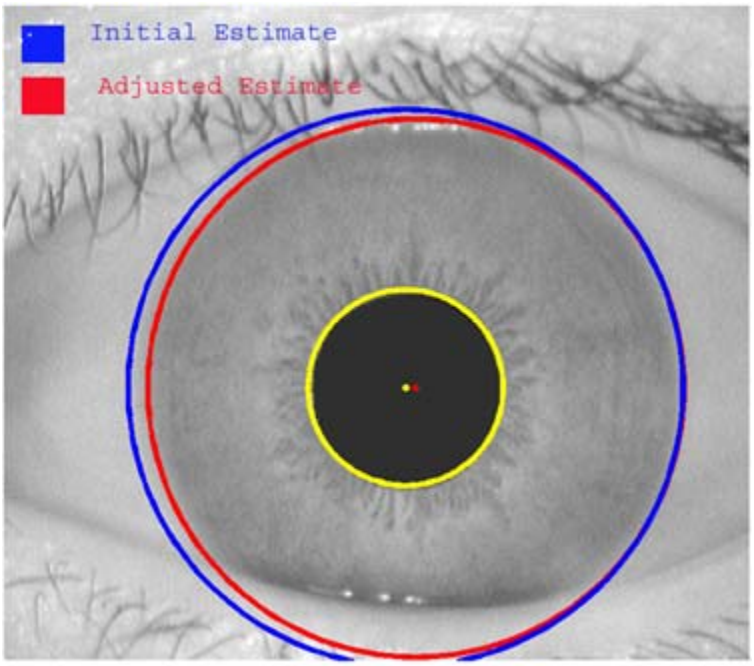


Figure 7.12: The pupil center and perimeter, along with the original estimate of the iris perimeter and the determined iris center and perimeter

7.4 Iris Recognition: Unwrapping the Iris

7.4.1 Why is unwrapping needed?

Image processing of the iris region is computationally expensive. In addition the area of interest in the image is a 'donut' shape, and grabbing pixels in this region requires repeated rectangular-to-polar conversions. To make things easier, the iris region is first unwrapped into a rectangular region using simple trigonometry. This allows the iris decoding algorithm to address pixels in simple (row, column) format.

7.4.2 Asymmetry of the eye

Although the pupil and iris circles appear to be perfectly concentric, they rarely are. In fact, the pupil and iris regions each have their own bounding circle radius and center location. This means that the unwrapped region between the pupil and iris bounding does not map perfectly to a rectangle. This is easily taken care of with a little trigonometry.
There is also the matter of the pupil, which grows and contracts its area to control the amount of light entering the eye. Between any two images of the same person's eye, the pupil will likely have a different radius. When the pupil radius changes, the iris stretches with it like a rubber sheet. Luckily, this stretching is almost linear, and can be compensated back to a standard dimension before further processing.

7.4.3 The unwrapping algorithm

In figure 7.13, points Cp and Ci are the detected centers of the pupil and iris respectively. We extend a wedge of angle dθ starting at an angle θ, from both points Cp and Ci, with radii Rp and Ri, respectively. The intersection points of these wedges with the pupil and iris circles form a skewed wedge polygon P1 P2 P3 P4. The skewed wedge is subdivided radially into N blocks. and the image pixel values in each block are averaged to form a pixel (j,k) in the unwrapped iris image, where j is the current angle number and k is the current radius number.

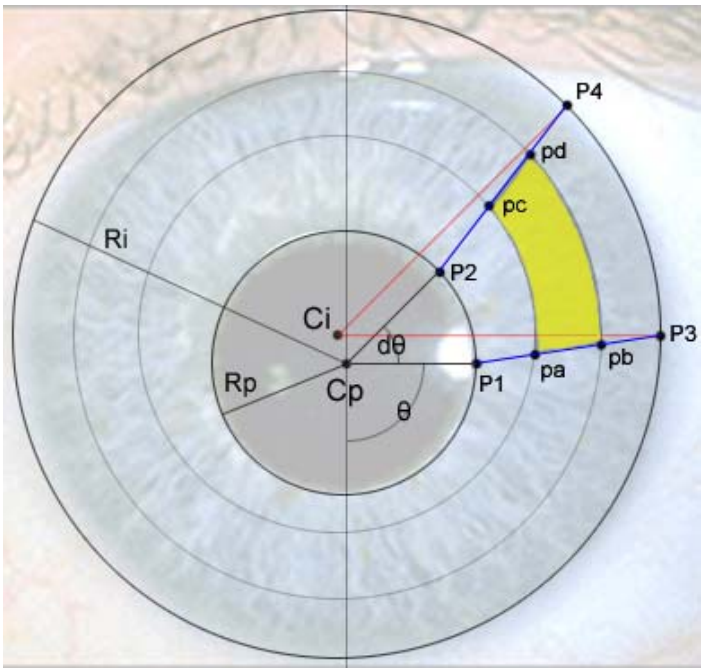


Figure 7.13: Algorithm for unwrapping the iris region.

For this project, the standard dimensions of the extracted iris rectangle are 128 rows and 8 columns (see Figure 4). This corresponds to N=128 wedges, each of angle 2n/128, with each wedge divided radially into 8 sections. The equations below define the important points marked in Figure 1. Points Pa through Pd are interpolated along line segments P1-P3 and P2-P4.

$P1=Cp+Rp(\cos(\theta) -\sin(\theta))$

$P2=Cp+Rp(\cos(\theta+d\theta) -\sin(\theta+d\theta))$

$P3=Ci+Ri(\cos(\theta) -\sin(\theta))$

$P4=Ci+Ri(\cos(\theta+d\theta) -\sin(\theta+d\theta))$

$Pa = P1\ (1 – K/N) + (P3K/N)Pb = P1\ (1 – (K+1)/n) + (P3\ (K+1)/N)Pc = P2(1-K/N) + (P4k/N)Pd = P2(1 – (K+1)/N) + P4(K+1)/N$

Subfigure: A B C

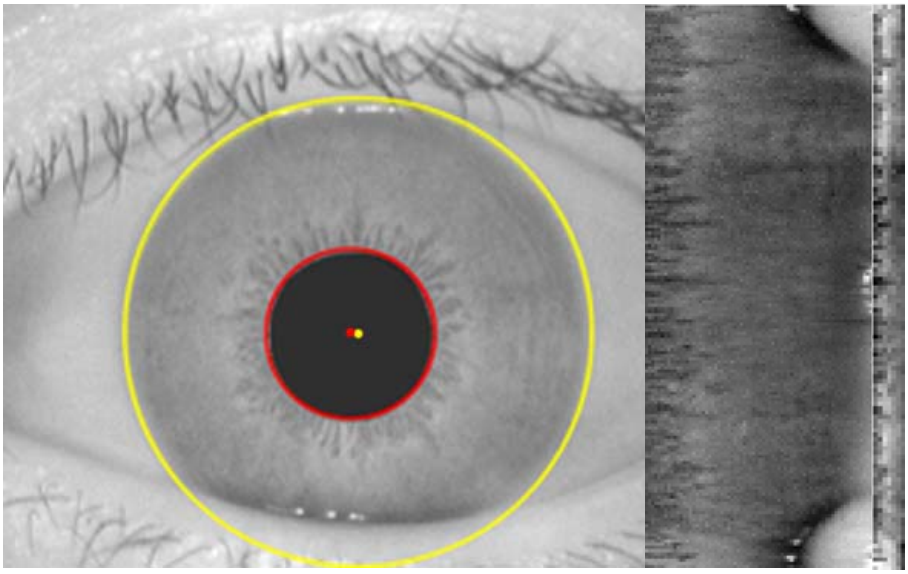


Figure 7.14: (A) Detected iris and pupil circles. (B) Iris extracted into 180 angle divisions, 73 radius divisions. (C) Iris extracted into 128 angle divisions, 8 radius divisions.

7.4.4 Masking of Extraneous Regions

Above figure demonstrates a high-resolution unwrapping. Note the large eyelid regions at the top and bottom of the image. These are the areas inside the iris circle that are covered by an eyelid. These regions do not contain any useful data and need to be discarded. One way to do this is to detect regions of the image that are unneeded and note the position of pixels within the region. Then, when the iris pattern is decoded and compared to another image, only regions that are marked "useful" in both images are considered.

A less robust method of ignoring the eyelid regions is to extract the inner 60% of the region between the pupil and iris. This assumes that an eyelid within this 50% will be detected before unwrapping and the image will be discarded. While simpler to implement, this method has the drawback that less iris data is extracted for comparison.

Chapter 8: Proposed Technique & Implementation Details

8.1 Problem Statement:

"Develop an Iris Recognition System Using Fisher Linear Discriminant Analysis Method."

8.2 Flow Chart of Proposed Iris Recognition System

8.2.1 Flow Chart of Overall System

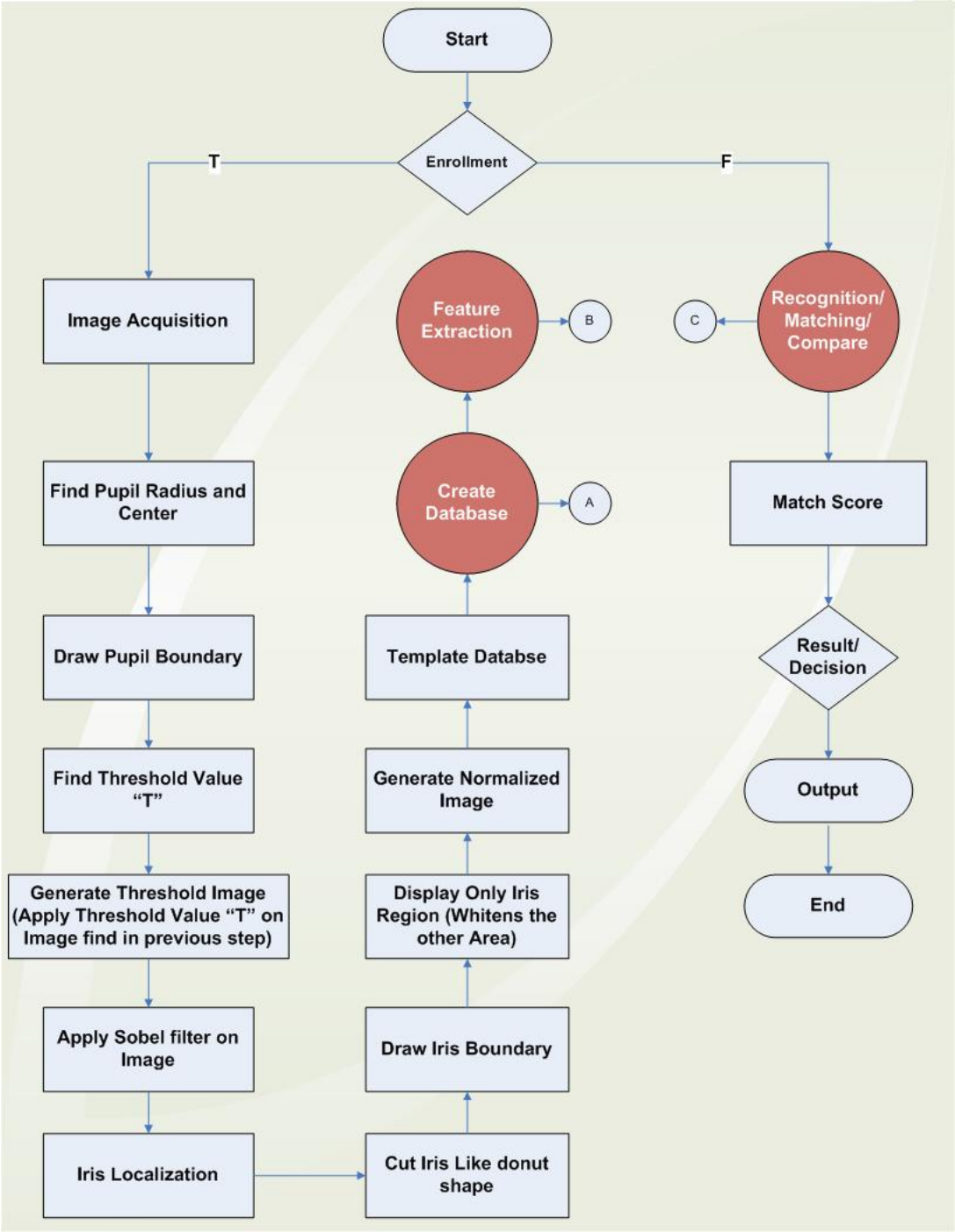


Figure 8.1: Flow Chart of Overall Iris Recognition System

8.2.2 Flow Chart of Create Database

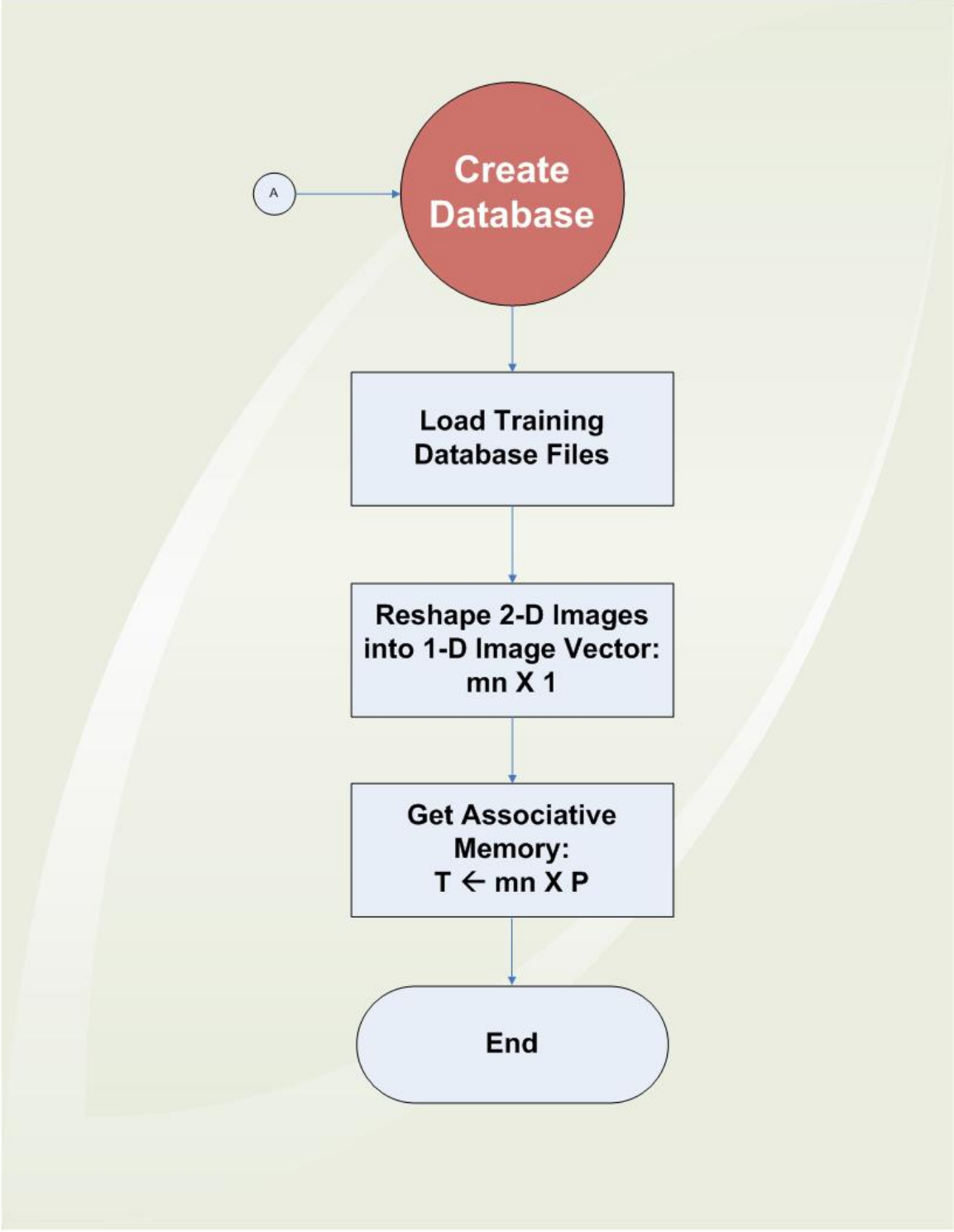


Figure 8.2: Flow Chart of Create Database Module

8.2.3 Flow Chart of Feature Extraction

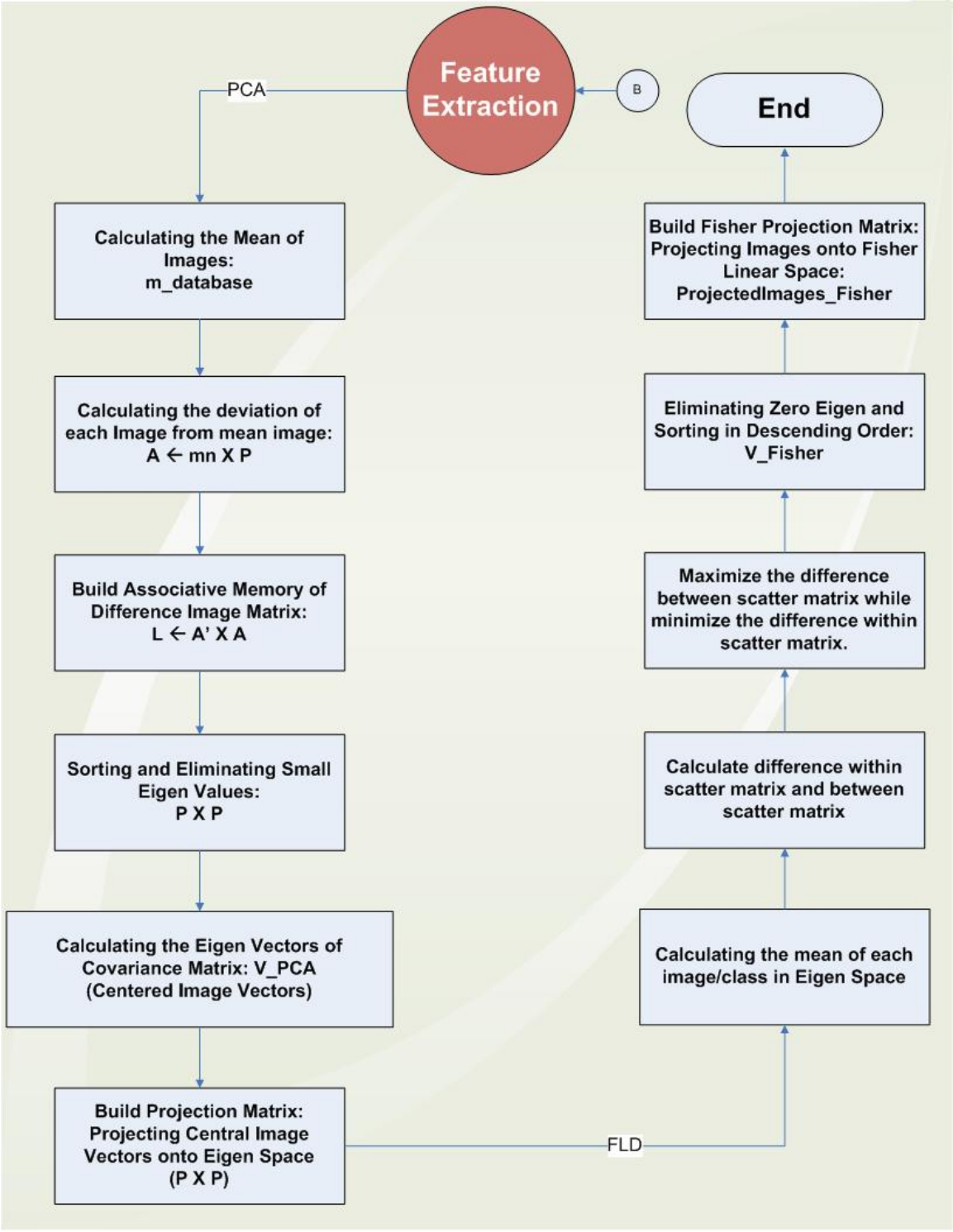


Figure 8.3: Flow Chart of Feature Extraction Module

8.2.4 Flow Chart of Recognition / Matching

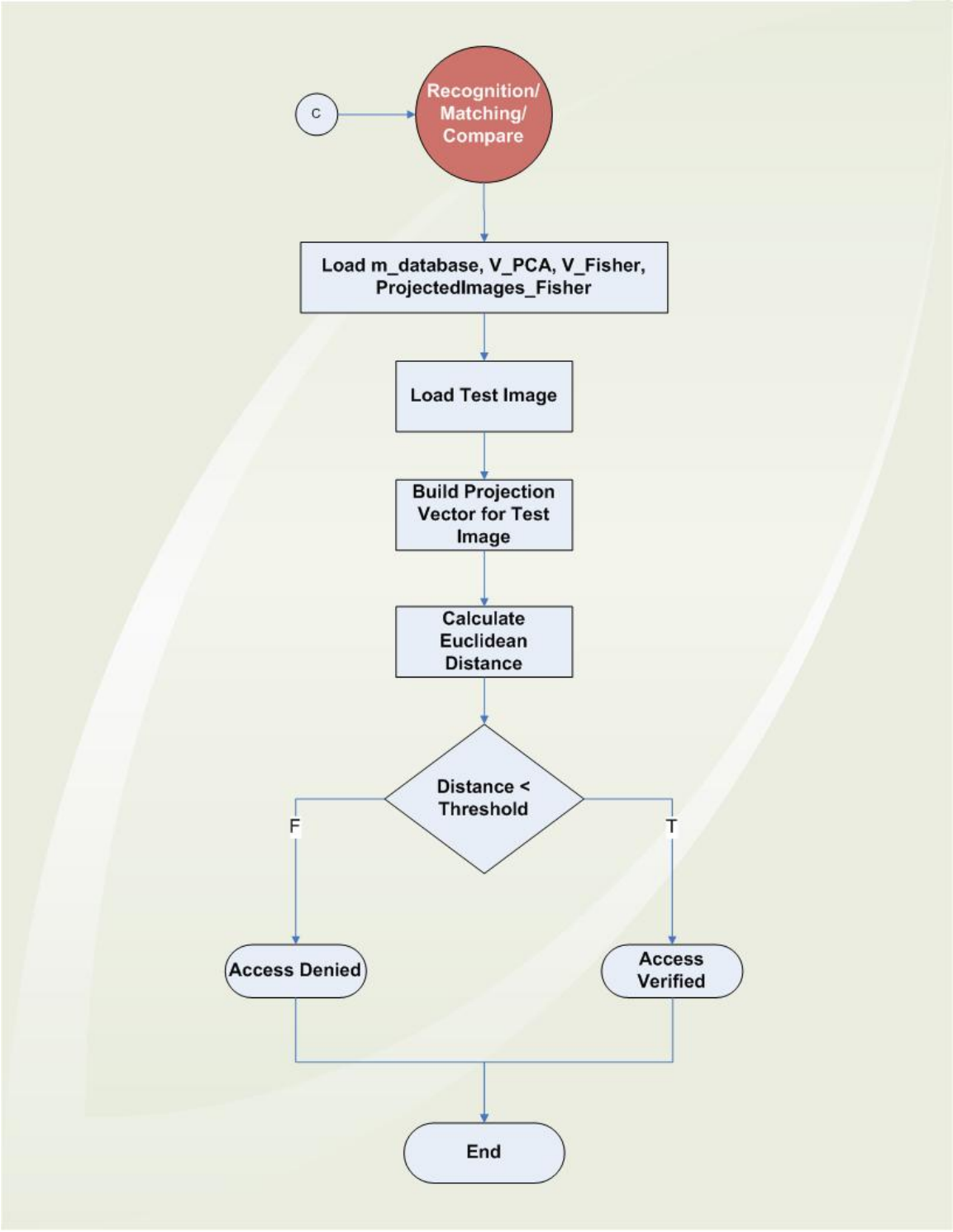


Figure 8.4: Flow Chart of Recognition / Matching Module

8.3 How Proposed System Works

8.3.1 Data Sets: Chinese Academy of Sciences - Institute of Automation

The Chinese Academy of Sciences - Institute of Automation (CASIA) eye image database is used in this research work. The images were captured especially for iris recognition research using specialized digital optics developed by the National Laboratory of Pattern Recognition, China. The eye images are mainly from persons of Asian decent, whose eyes are characterized by irises that are densely pigmented, and with dark eyelashes. Due to specialized imaging conditions using near infra-red light, features in the iris region are highly visible and there is good contrast between pupil, iris and sclera regions.

8.3.2 Image Capture

In practical application of a workable system, an image of the eye to be analyzed must be acquired first in digital form suitable for analysis.

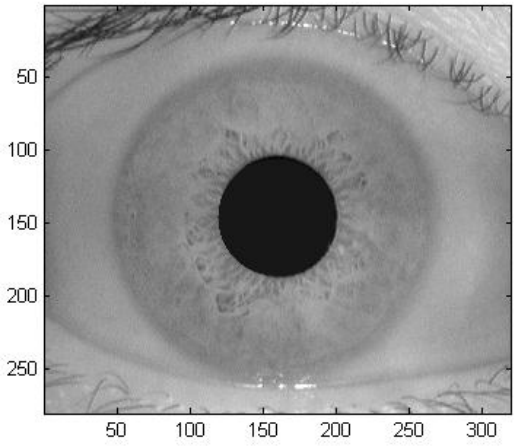


Figure 8.5: Eye Image

8.3.3 Pupil Detection

Apply the thresholding with the value zero in order to make the pupil appear alone. Using the following algorithm:

1. *Input (Original_image[height][width], height, width)*
2. *Output (threshold_image)*
3. *Begin*
4. *Initialize (T=0, x=0, y=0)*
5. *do x ← 0*
6. *do y ← 0*
7. *if Original_image[x][y] <= T then Original_image[x][y] ← 0*
8. *else Original_image[x][y] ← 255*
9. *end if*
10. *y ← y + 1*
11. *until y = width – 1*
12. *x ← x + 1*
13. *until x = height – 1*
14. *Return Original_image*
15. *End*

8.3.4 Pupil Detection Algorithm

The following algorithm is used to find the centre and (x,y) coordinates of the centre.

1. *To determine the center of the pupil (x, y), find the maximum number of black points in each row and column*
2. *Count every pixel in each row, Find the index of the row that contains the maximum number of black points*
3. *y is the row index.*
4. *Count every pixel in each column, Find the index of the column that contains the maximum number of black points*
5. *y1 is the column index.*
6. *Find out the coordinates of the edge points in the row "y" and column "y1“*

7. *Find the X coordinate of the center,*

   Xc = floor((X1 + X2) / 2);

8. *Find the Y coordinate of the center*

   Yc = floor((Y1 + Y2) / 2);

9. *Radius = ((X1 - X2)^2 + (y - y)^2) ^ 0.5 / 2;*

8.3.5 Pupil with Longest Row, Column

we calculate the longest black row horizontally, and the longest column vertically, by counting the number of black points in each row and in each column as in figure, then find out the coordinate of the center easily and also the radius.

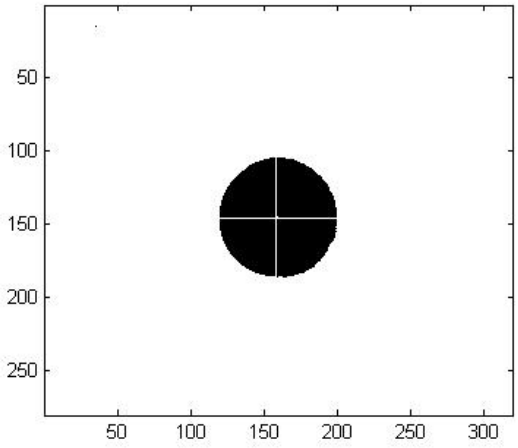


Figure 8.6: Pupil with Longest Row, Column of black points

In Matlab the X-axis is the vertical one and Y-axis is the horizontal one, because of that i swap between Xc, Yc;

Original_image(Yc, Xc) = 255;

We will notice that the white point is in the middle of the pupil.

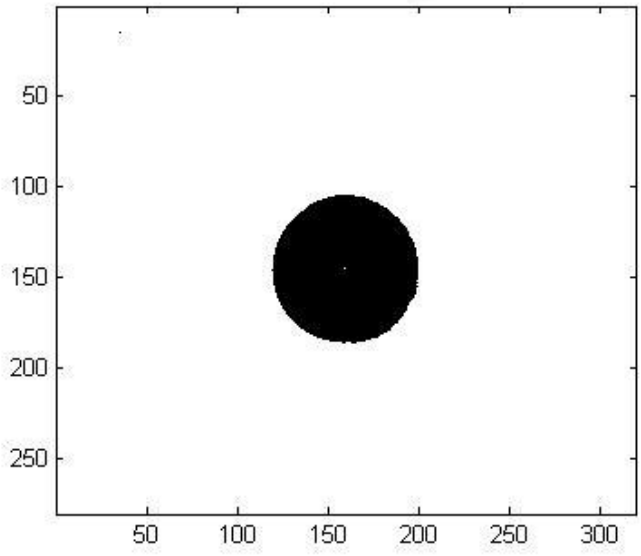


Figure 8.7: White point is in the middle of the pupil

Now draw the radius of the Pupil using center coordinates (X, Y) and radius (R).

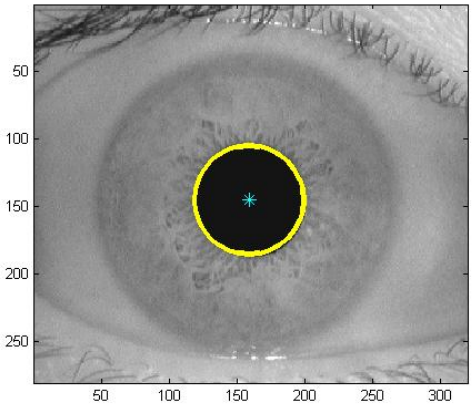


Figure 8.8: Radius of Pupil

8.3.6 Iris Localization

Using Basic Global Thresholding algorithm to find thresholding value (T). Implement the thresholding with T from previous algorithm on Original_image.

$$\text{Original\_image} = \begin{cases} 0 & \text{Original\_image} <= T \\ 255 & \text{Original\_image} > T \end{cases}$$

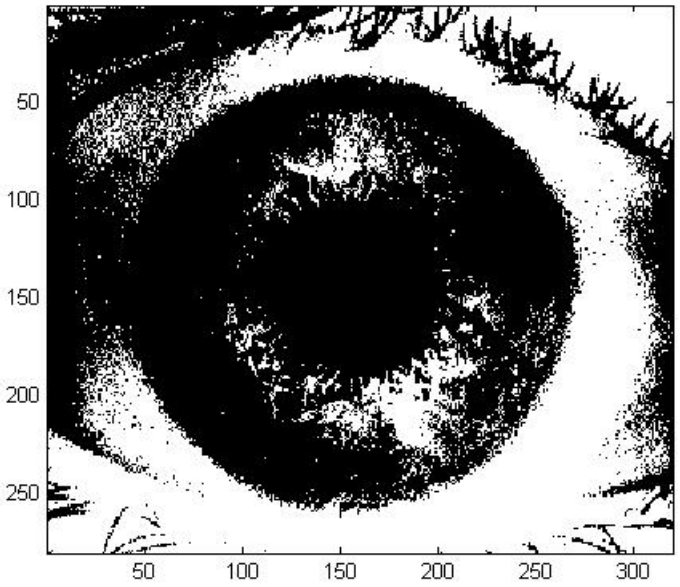


Figure 8.9: Thresholded Eye Image

8.3.7 Basic Global Thresholding Algorithm to Find Value of T.

Now apply the global thresholding algorithm to find the value thresholding value 'T'. Implement the thresholding with T from previous algorithm on Original _ image.

1. *Input (Original_image[height][width], height, width)*
2. *Output (thresholding_value (T))*
3. *Begin*
4. *Initialize (M1=0, M2=0, S1=0, S2=0, C1=0, C2=0, T=0)*
5. *M1 ← min (Original_image)*
6. *M2 ← max (Original_image)*
7. *do*

8. T ← (M1+M2)/2

9. S1 ← S2 ← C1 ← C2 ← 0

10. do x ← 0

11. do y ← 0

12. if Original_image[x][y] > T then

13. S1 ← S1 + Original_image[x][y]

14. C1 ← C1 + 1

15. else

16. S2 ← S2 + Original_image

17. C2 ← C2 + 1

18. end if

19. y ← y + 1

20. until y = width – 1

21. x ← x + 1

22. until x = height – 1

23. M1 ← S1/C1

24. M2 ← S2/C2

25. Until T=(M1+M2)/2

26. Return T

27. End

8.3.8 Image Complement and Filtration

To remove the black points that may appear we apply complement and smoothing technique (Median filter with size 5x5) two times.

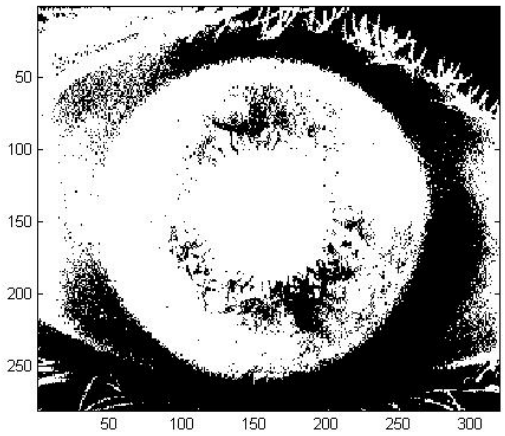


Figure 8.10: Complement of the image

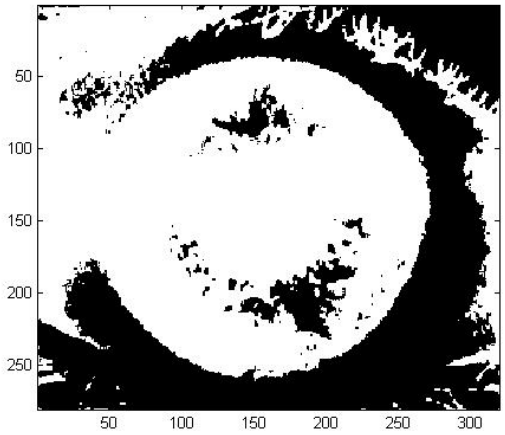


Figure 8.11: Apply median filter

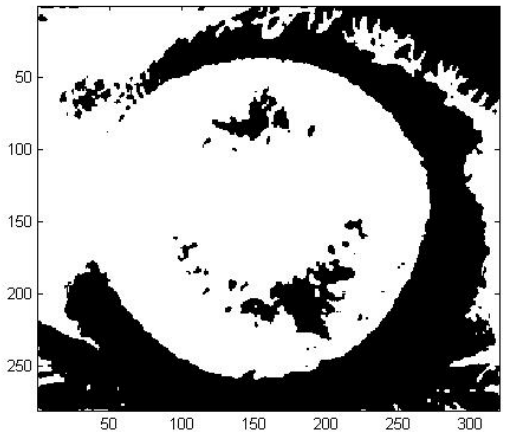


Figure 8.12: Again apply median filter

The eye image has a big different between the values of pupil and iris so we can make segmentation to it by using edge detection techniques (convolution x and y direction Sobel masks on the Original_image and get the maximum between them). Now applying Sobel Edge Detection on threshold image.

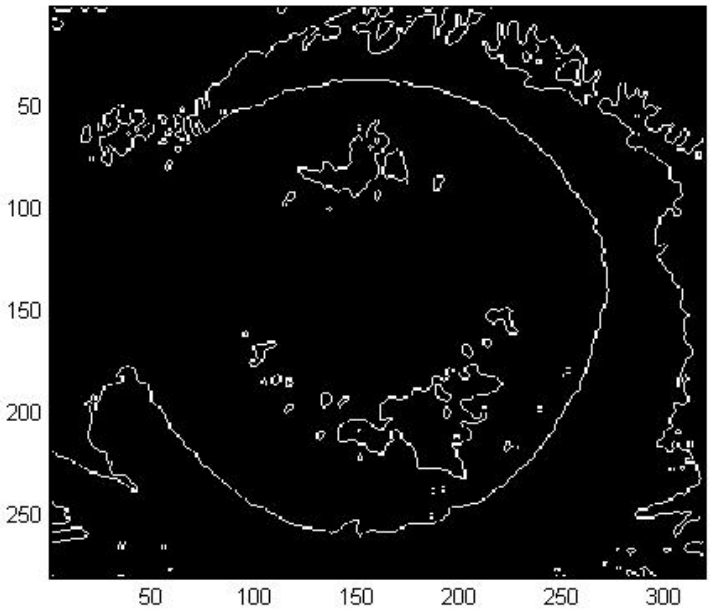


Figure 8.13: Edge detection using sobel filter

8.3.9 Using Hough Transform Circle algorithm to find outer radius (outer_r) and use the center pupil coordinates (Xc, Yc) as (C1, C2).

1. Input (Original_image[height][width],
height, width, Xc, Yc, inner_r)
2. Output (Output_r)
3. Begin
*4. Initialize (C3min = 4 * inner_r^2, C3max*
*= 9*inner_r^2, C3 = 0, C2 = Xc, C1 = Yc,*
A[C3max – C3min + 1] = {0}, x = 0, y = 0)
5. do x ← 0
6. do y ← 0
7. If Original_image[x][y] = 1 then
C3 ← (x – C1)^2 + (y – C2)^2
8. If C3min < C3 and C3 < C3max then
A[C3 – C3min] ← A[C3 – C3min] + 1
9. end if
10. end if
11. y ← y + 1
12. Until y = width - 1
13. x ← x + 1
14. Until x = height - 1

After applying the algorithm given on right side, now we take the index or the offset of the location that has the maximum value:

15. z = 1;
16. range = C3max-C3min+1;

17. for i = 1 : range
18. if (A(i) > A(z))

19. z = i;

20. end

21. end

22. Outer_r = floor ((C3min+z) ^ 0.5);

Which means find the index related to the location which has the maximum value in A; location of { max(A)}.

8.3.10 Draw the radius of the Iris using center coordinates (X, Y) and Outer Radius (R).

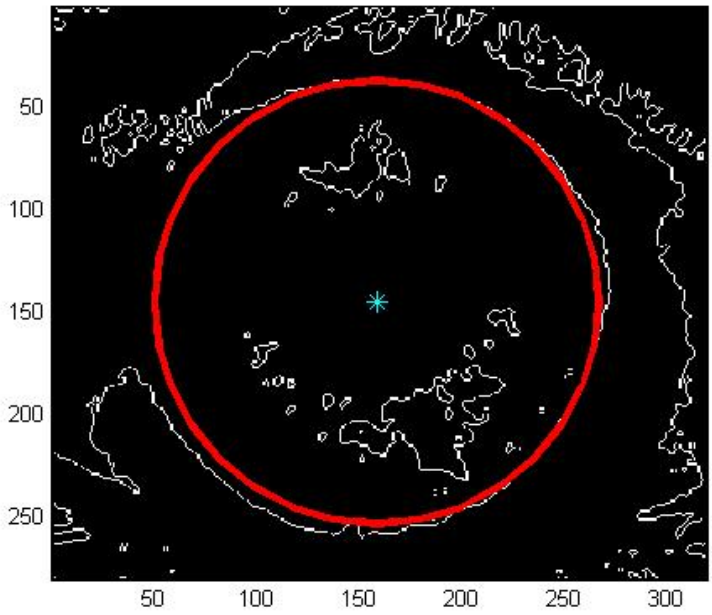


Figure 8.14: Radius of Iris

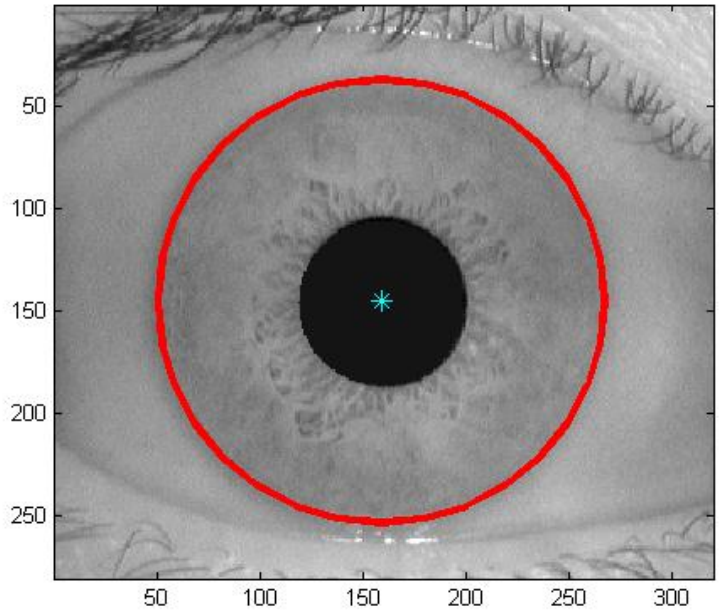


Figure 8.15: Draw Radius of Iris

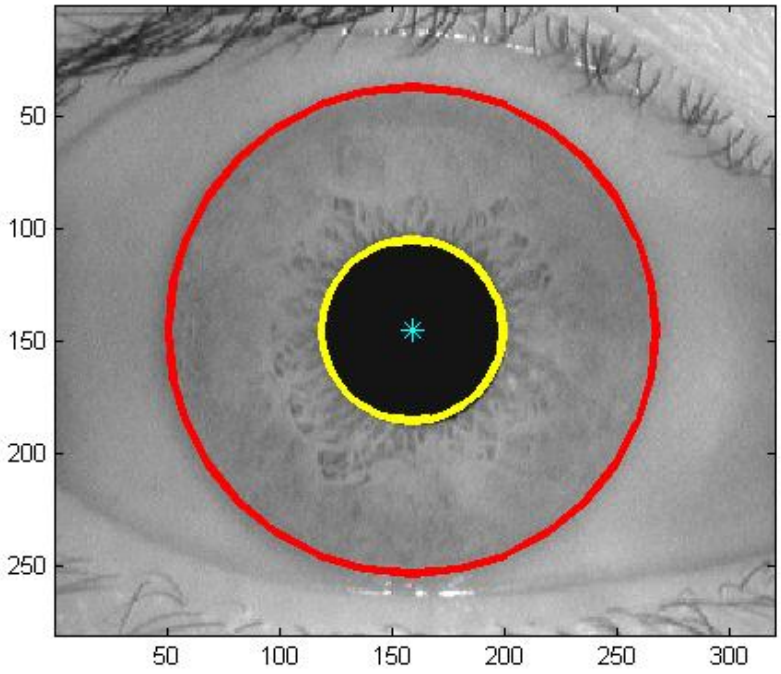


Figure 8.16: Draw radius of Pupil and Iris

8.3.11 Cut the Iris Region

The iris is the area between outer circle and inner circle. We got the pixels of this area and colored the background with black. The following pseudo code describes the steps;

1. Input (Original_image[height][width],
height, width, Xc, Yc, inner_r, outer_r)
2. Output (Original_image has iris image)
3. Begin
4. Initialize (r2=0, inner_r2 = inner_r^2, outer_r2 = outer_r^2)
5. do x ← 0
6. do y ← 0
7. r2 ← (x - Xc)^2 + (y - Yc)^2
8. If r2 < inner_r2 or r2 > outer_r2 then
9. Original_image[x][y] ← 0
10. end if
11. return Original_image
12. End

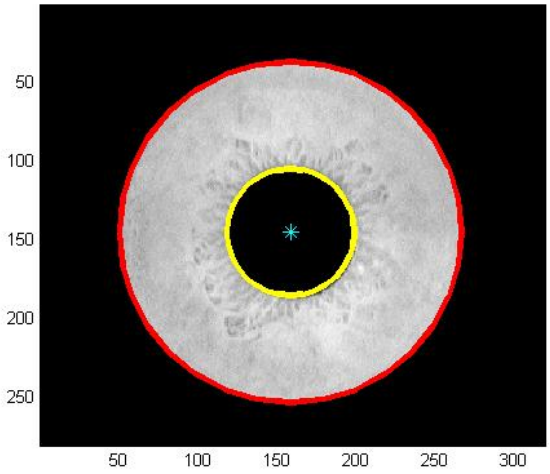


Figure 8.17: Cut the Iris region

8.3.12 Display the Iris Region

The following algorithm is used to display the Iris region only. This algorithm whitens the non-iris region.

1. *Input (Original_image[height][width], height, width)*
2. *Output (Original_image has iris region only)*
3. *Initialize(x=0, y=0)*
4. *do x ← 0*
5. *do y ← 0*
6. *if Original_image[x][y] = 0 then*
7. *Original_image ← 255*
8. *end if*
9. *y ← y + 1*
10. *until y = width – 1*
11. *x ← x + 1*
12. *until x = height – 1*
13. *return Original_image*
14. *end*

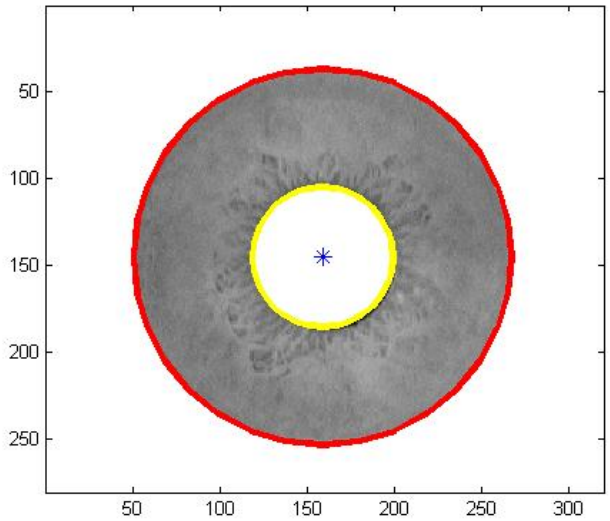


Figure 8.18: Whitens the Non-Iris Region

8.3.13 Iris Normalization

1. Typical next step is to “normalize” the iris to a rectangular representation.
2. Some processing may be performed to improve quality of the image.

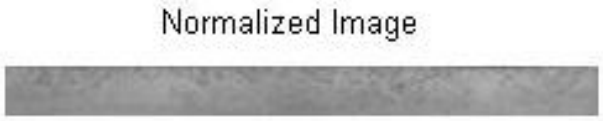


Figure 8.19: Normalized Image

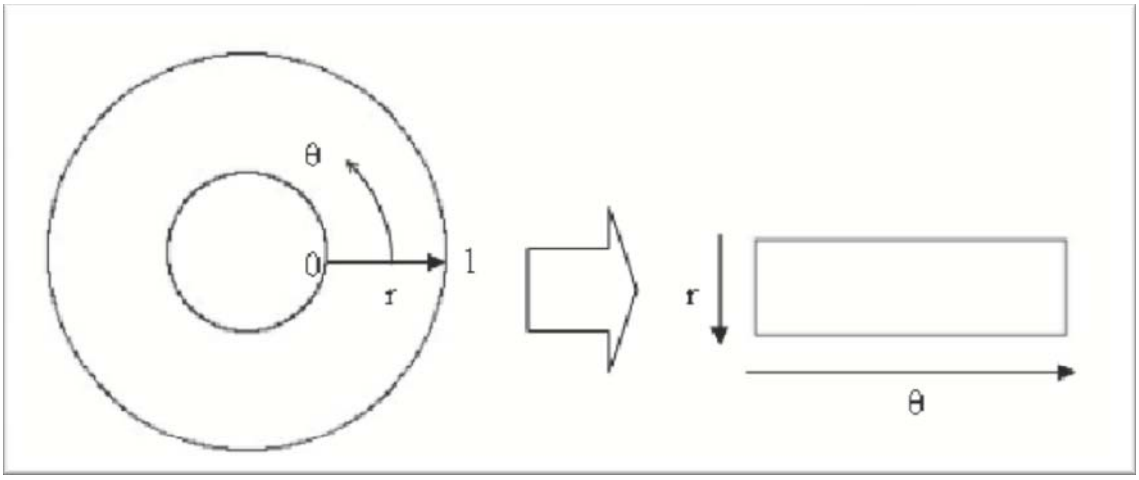


Figure 8.20: Conversion of Cartesian to polar coordinates system

There is dimensional inconsistency between the eye images due to the stretching of iris caused by pupil extraction / contraction from variation of illumination.

The normalization process projects iris region into a constant dimensional ribbon so that the two images of the same iris under different conditions have characteristic features at the same spatial location.

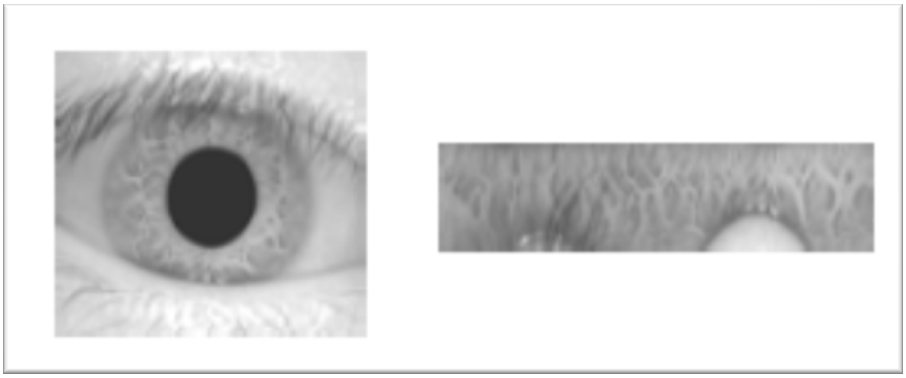


Figure 8.21: Normalized Iris Image

The radial resolution is set to 20 and angular resolution is set to 240 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.

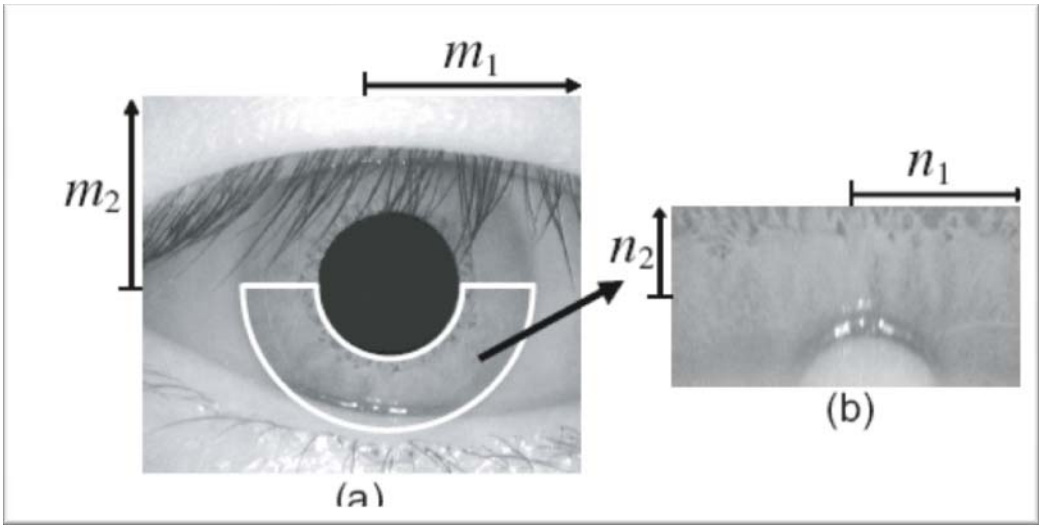


Figure 8.22: Normalized Iris Image by using lower half

8.3.14 Normalization Algorithm

1. *Upper and lower half of the iris region is used.*
2. *Upper region is usually covered by eye lashes.*

3. *The lower half is enough for the further processing i.e. feature extraction and matching etc.*
4. *The normalized iris region usually has low contrast.*
5. *Improve the contrast by using local histogram equalization*

8.3.15 Create Database

The following algorithm is used to create the database.

1. *Input training database path and output associative memory T*
2. *Read the training files by ignoring dot(.) double dot(..) and thumbs files from the training database.*
3. *Read gray images one by one and convert it into 1-D image using reshape function and save it into associative memory*
4. *Return T*

8.3.16 Feature Extraction

The following algorithm is used to extract the features from normalized iris region;

1. *Call feature extraction function by passing T and getting mean, v_pca, v_fisher and projected images fisher*
2. *Get no of classes and class population.*
3. *Calculate the mean of T matrix*
4. *Subtract this mean from every column of T matrix and get difference matrix A*
5. *Get the covariance matrix L= A' * A*
6. *Find eigenvectors of L*
7. *Find pca vectors by A*eigenvectors of L*
8. *Find projected pca by multiplying transpose of pca vectors and A*
9. *Calculate the mean of projected pca matrix*
10. *Initialize m as total patterns-no of classes by no of classes matrix.*

11. *Initialize Sw (scattered within) as total patterns-no of classes by total patterns-no of classes.*
12. *Initialize Sb (scattered between) as total patterns-no of classes by total patterns-no of classes.*
13. *Filling of m matrix;*
    a) *Run the loop up to no of distinct images.*
    b) *Each time find the mean of similar images w.r.t column and then take transpose of each and store it in mean matrix "m".*
    c) *Initialize "S" as total patterns-no of classes by total patterns-no of classes;*
        i. *Run a loop for no of pattern in single class.*
        ii. *Find the difference of each projected_PCA image from the mean of that respective class*
        iii. *Multiply this mean transpose of itself.*
        iv. *Add it in "S".*
    d) *Fill Sw with "S".*
    e) *Find difference of mean calculated in "b" and the mean of PCA and multiply it with transpose of itself.*
    f) *Fill Sb with the result of "e".*
14. *Find the Eigen Vector of Sb and Sw in such a way that Sb*Eigen Vector = Sw*Eigen Vector*D (where D is diagonal vector of Eigen values).*
15. *Then "fliplr" is used to rearrange the vectors of Eigen Vector from left to right.*
16. *Make a matrix "V_fisher" of largest vectors Eigen Vector matrix of class number -1.*
17. *Run a loop for total number of patterns.*
    a) *Take the transpose of "V_fisher" and multiply it with projected image PCA of respective pattern.*

8.3.17 Recognition

The following algorithm is used to recognized the test images from the database;

1. *Input test image, PCA Vector, Fisher _Vector and Projected_Fisher _Images and it return the index of recognized image or set the flag otherwise.*
2. *Get the no. of columns of Projected_Fisher _Image.*
3. *Get the test image and covert to grayscale.*
4. *Convert test image into 1-D vector.*
5. *Find the difference by subtracting from mean image.*
6. *Find Projected_Test _Image by multiplying PCA Vector, Fisher_Vector and difference.*
7. *Find Euclidean distance by subtracting every Projected_Fisher_Image from Projected_Test _Image then taking norm and then squaring.*
8. *Find minimum Euclidean distance*
9. *Give some threshold on Euclidean distance say 18.8835e-020 and find the matched image otherwise set the flag.*
10. *Return the output.*

8.3.18 Conclusion

- if (Euc_dist_min < 18.8835e-010)
    - Image exists, concatenate it with recognized image in training database and display both asked image and recognized image.
- Else
    - Image doesn't exist, inform the user.

Figure 8.23: Test Normalized Image

Figure 8.24: Equivalent Normalized Image

Chapter 9: Results/Comparisons & Discussion

In the previous chapter, the procedure of implementation is explained thoroughly. The iris pre-processing is performed and detects the iris region in the human eye. After the normalization and feature extraction of the iris, the system is tested upon the inputs.

Now in this chapter the results of the technique are described. Database used 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.

9.1 Scenario – 1

Implemented algorithm takes 10 subjects as input, whereas there are maximum of seven images for each subject and total number of images is 50, after testing I obtained the following results.

Number of Subjects	%age Recognition Testing images
10 (70 Images)	96.5 %

Table 1

The following figure 9.1 displays the testing results in which the input image is compared with all the templates in the database;

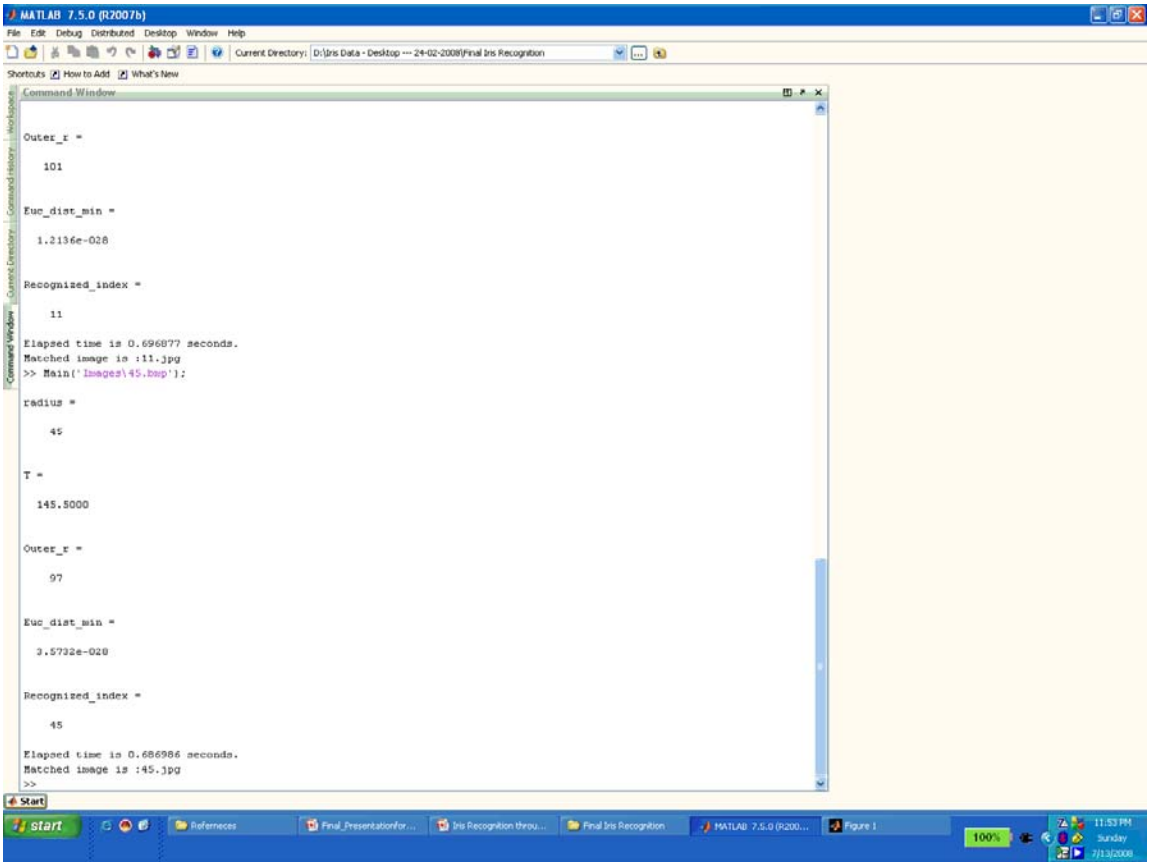


Figure 9.1: Results of testing of an image with 10 subjects

9.2 Scenario – 2

Implemented algorithm takes 19 subjects as input, whereas there are maximum of seven images for each subject and total number of images are 102, after testing I obtained the following results.

Number of Subjects	%age Recognition Testing images
19 (120 Images)	96 %

Table 2

The following figure 9.2 displays the testing results in which the input image is compared with all the templates in the database;

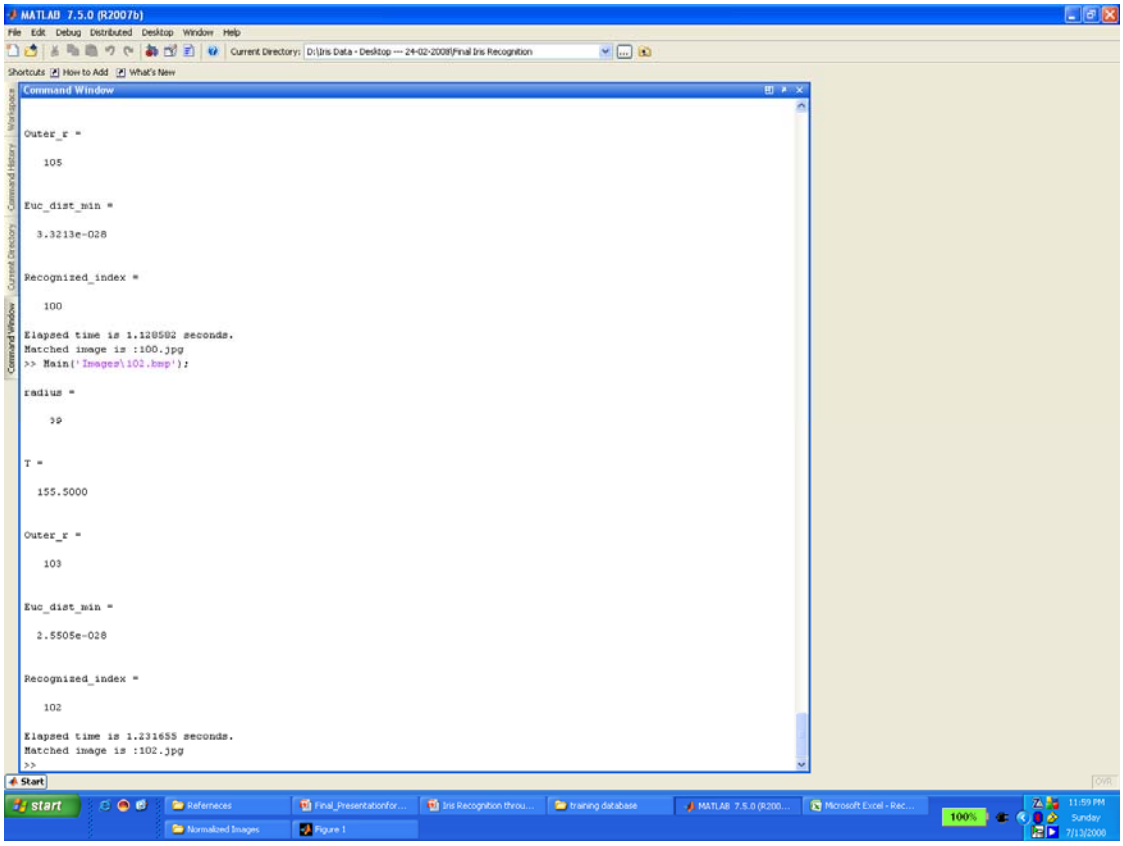


Figure 9.2: Results of testing of an image with 19 subjects

9.3 Scenario – 3

Implemented algorithm takes 29 subjects as input, whereas there are maximum of seven images for each subject and total number of images are 158, after testing I obtained the following results.

Number of Subjects	%age Recognition Testing images
29 (180 Images)	96.5 %

Table 3

The following figure 9.3 displays the testing results in which the input image is compared with all the templates in the database;

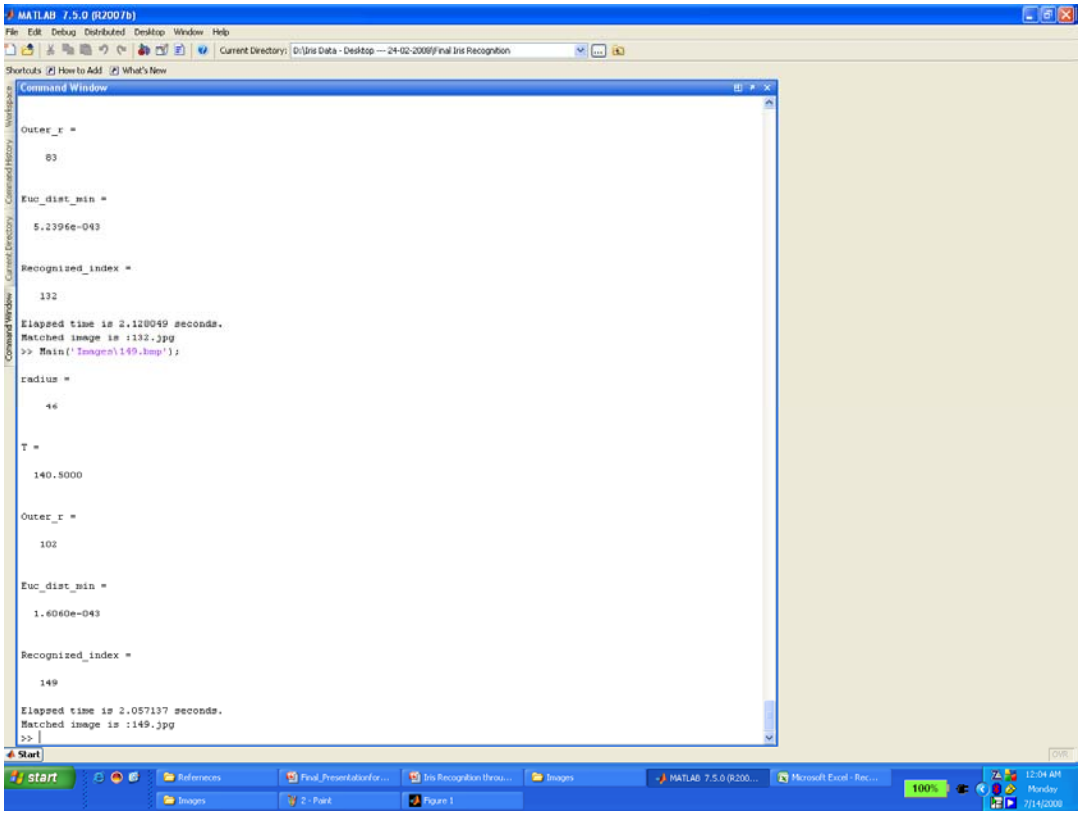


Figure 9.3: Results of testing of an image with 29 subjects

9.4 Scenario – 4

Implemented algorithm takes 34 subjects as input, whereas there are maximum of seven images for each subject and total number of images are 192, after testing I obtained the following results.

Number of Subjects	%age Recognition Testing images
34 (240 Images)	97 %

Table 4

The following figure 9.4 displays the testing results in which the input image is compared with all the templates in the database;

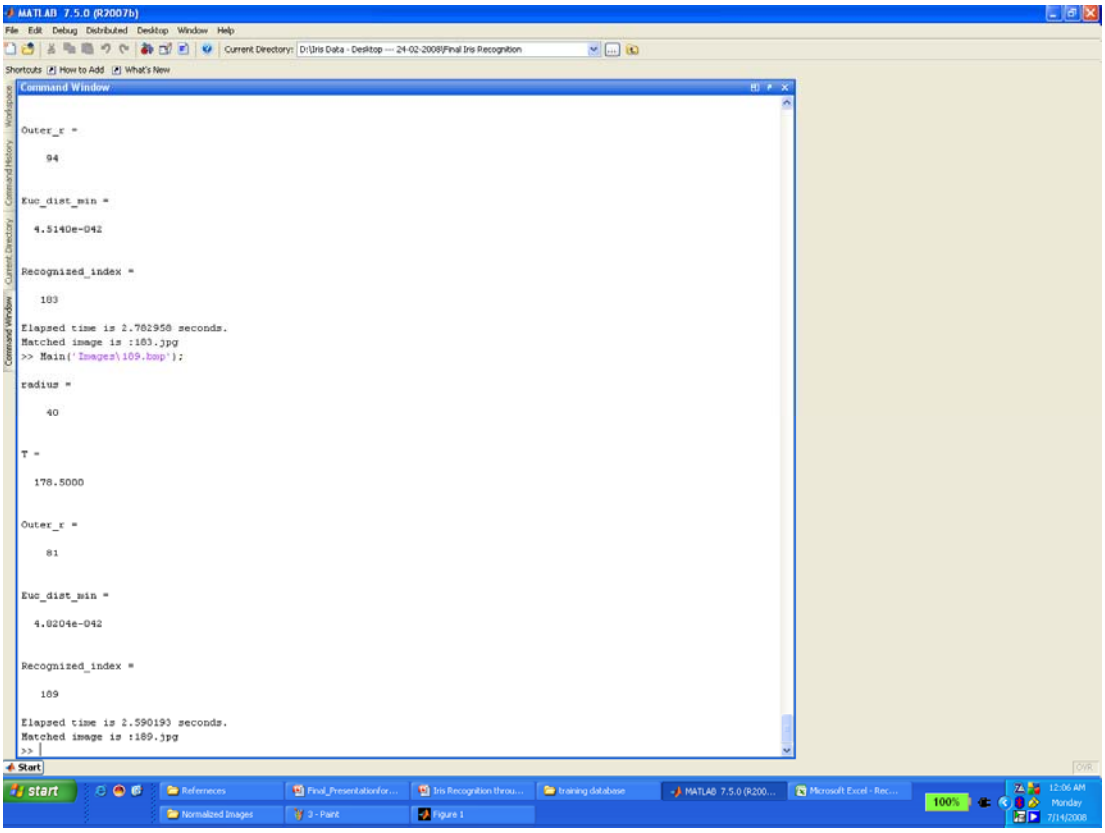


Figure 9.4: Results of testing of an image with 34 subjects

9.5 Scenario – 5

Implemented algorithm takes 45 subjects as input, whereas there are maximum of seven images for each subject and total number of images are 242, after testing I obtained the following results.

Number of Subjects	%age Recognition Testing images
45 (300 Images)	96.5%

Table 5

The following figure 9.5 displays the testing results in which the input image is compared with all the templates in the database;

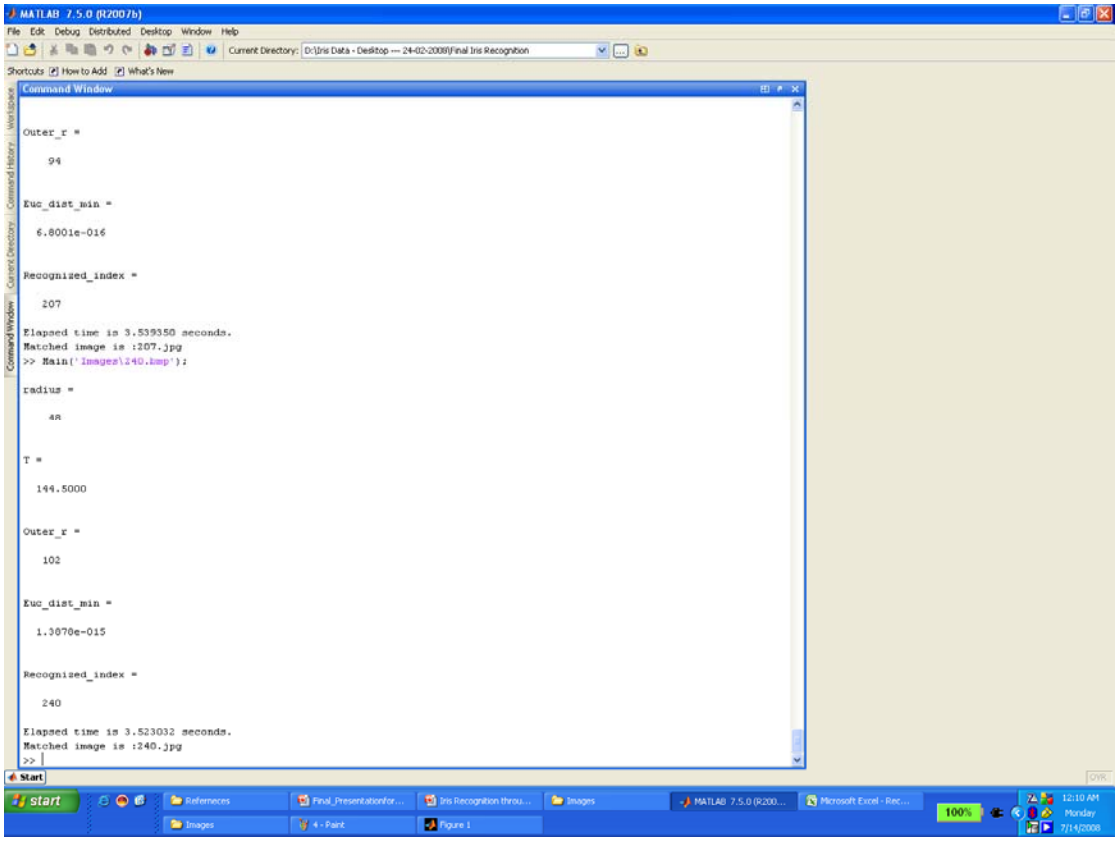


Figure 9.5: Results of testing of an image with 45 subjects

9.6 Scenario – 6

Implemented algorithm takes taken 54 subjects as input, whereas there are maximum of seven images for each subject and total number of images are 292, after testing I obtained the following results.

Number of Subjects	%age Recognition Testing images
54 (360 Images)	97.5 %

Table 6

The following figure 9.6 displays the testing results in which the input image is compared with all the templates in the database;

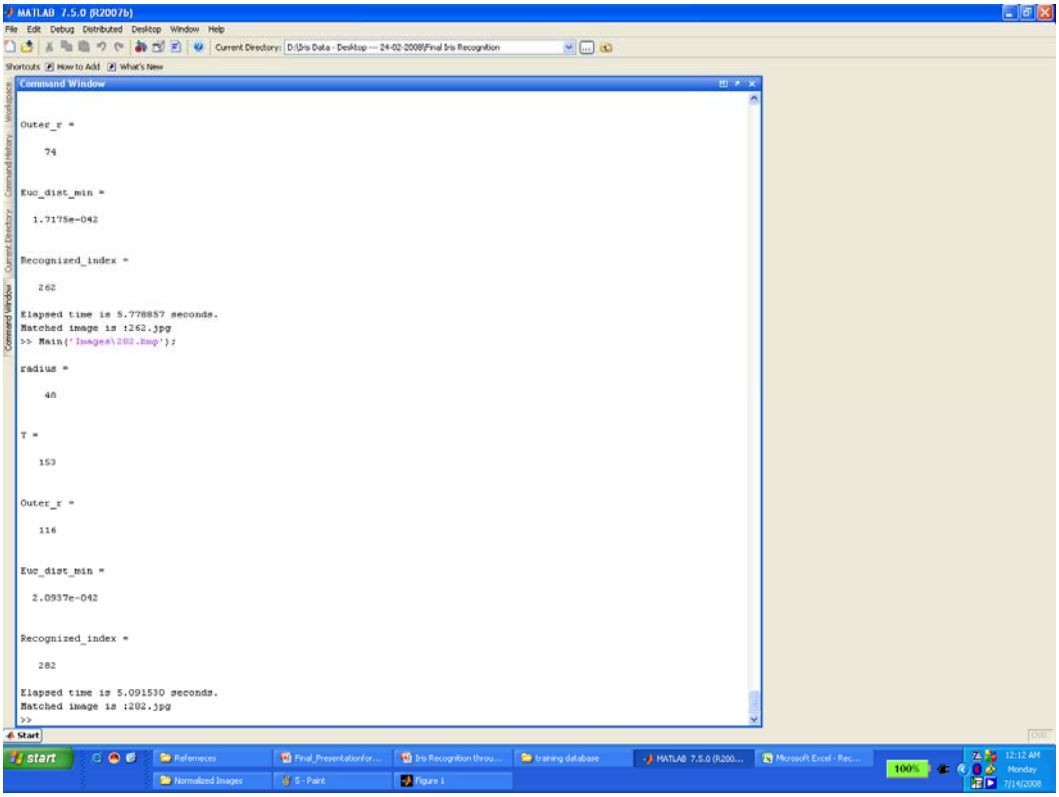


Figure 9.6: Results of testing of an image with 54 subjects

9.7 Scenario – 7

Implemented algorithm takes 64 subjects as input, whereas there are maximum of seven images for each subject and total number of images are 348, after testing I obtained the following results.

Number of Subjects	%age Recognition Testing images
64 (440 Images)	96.5 %

Table 7

The following figure 9.7 displays the testing results in which the input image is compared with all the templates in the database;

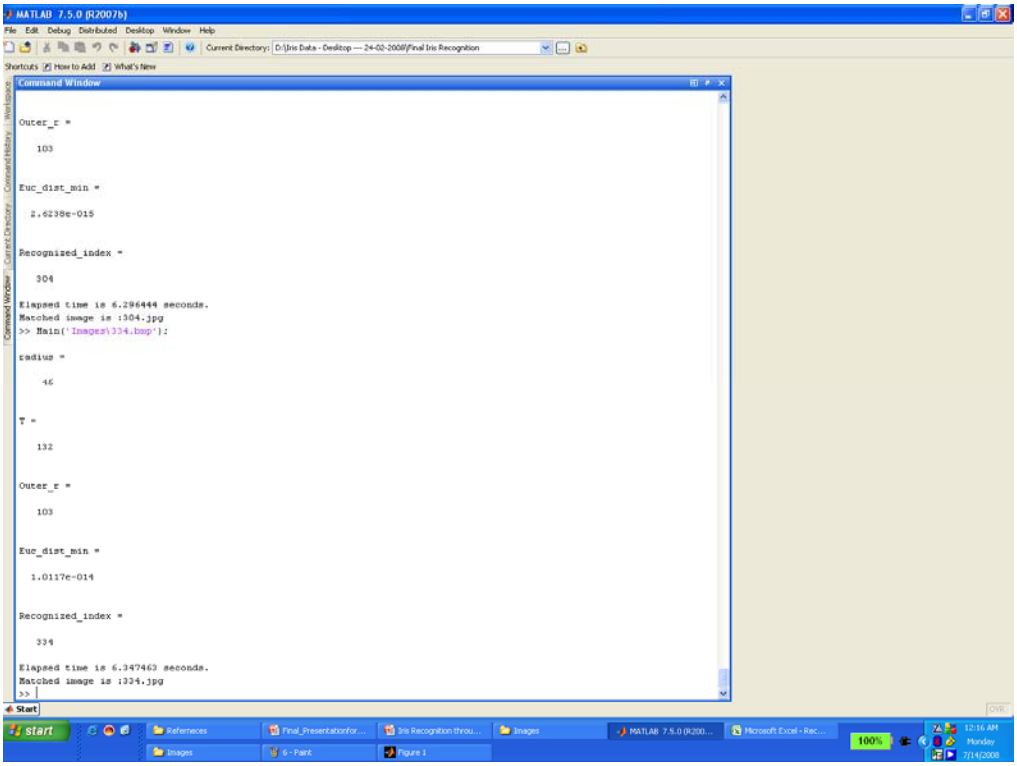


Figure 9.7: Results of testing of an image with 64 subjects

9.8 Scenario – 8

Implemented algorithm takes 71 subjects as input, whereas there are maximum of seven images for each subject and total number of images are 392, after testing I obtained the following results.

Number of Subjects	%age Recognition Testing images
71 (500 Images)	97 %

Table 8

The following figure 9.8 displays the testing results in which the input image is compared with all the templates in the database;

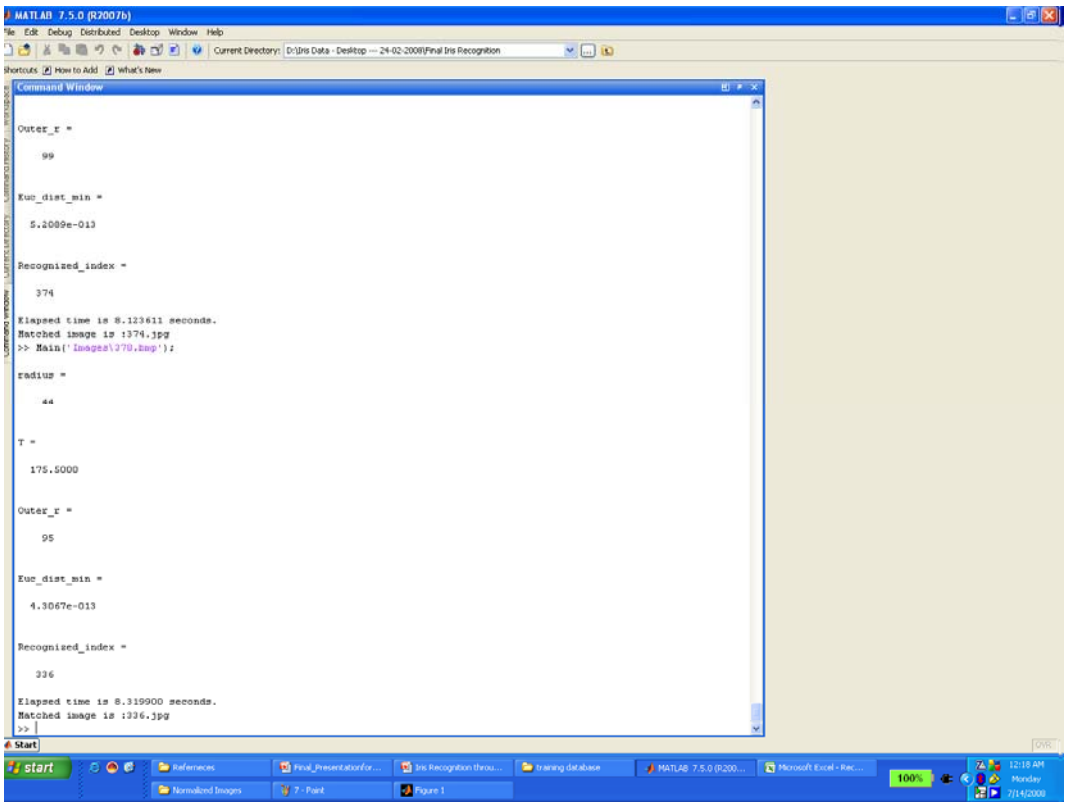


Figure 9.8: Results of testing of an image with 71 subjects

9.9 Scenario – 9:

Implemented algorithm takes 80 subjects as input, whereas there are maximum of seven images for each subject and total number of images are 444, after testing I obtained the following results.

Number of Subjects	%age Recognition Testing images
80 (550 Images)	97.5 %

Table 9

The following figure 9.9 displays the testing results in which the input image is compared with all the templates in the database;

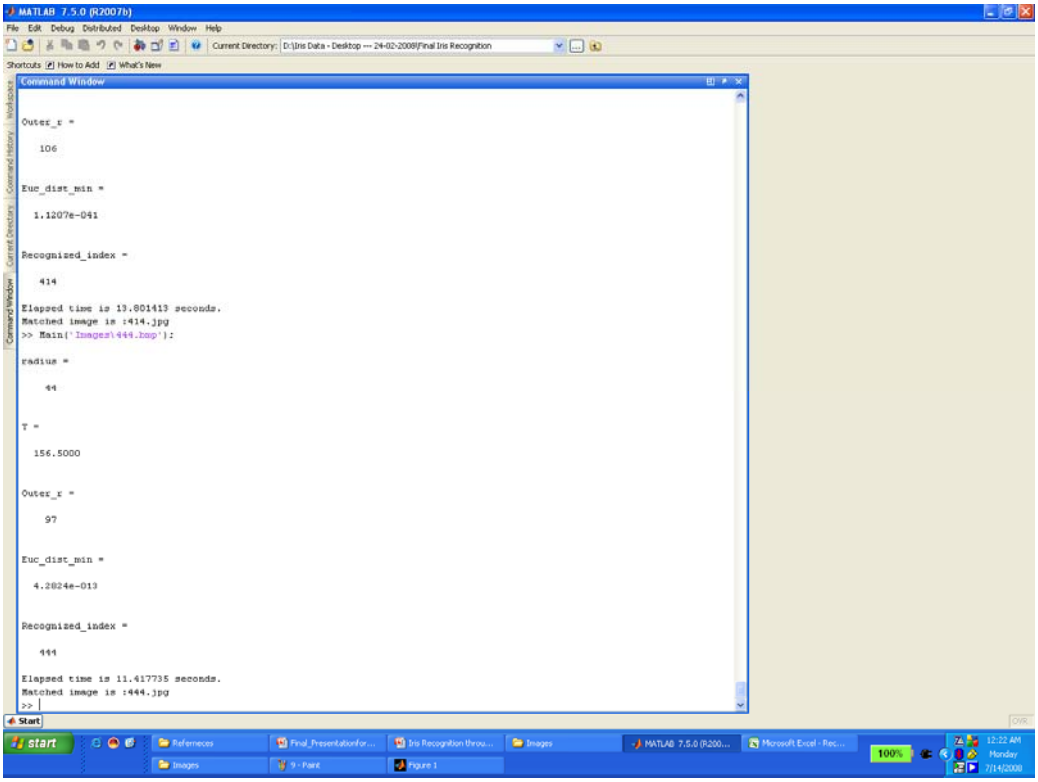


Figure 9.9: Results of testing of an image with 80 subjects

9.10 Scenario – 10

Implemented algorithm takes taken 92 subjects as input, whereas there are maximum of seven images for each subject and total number of images are 500, after testing I obtained the following results.

Number of Subjects	%age Recognition Testing images
92 (600 Images)	97 %

Table 10

The following figure 9.10 displays the testing results in which the input image is compared with all the templates in the database;

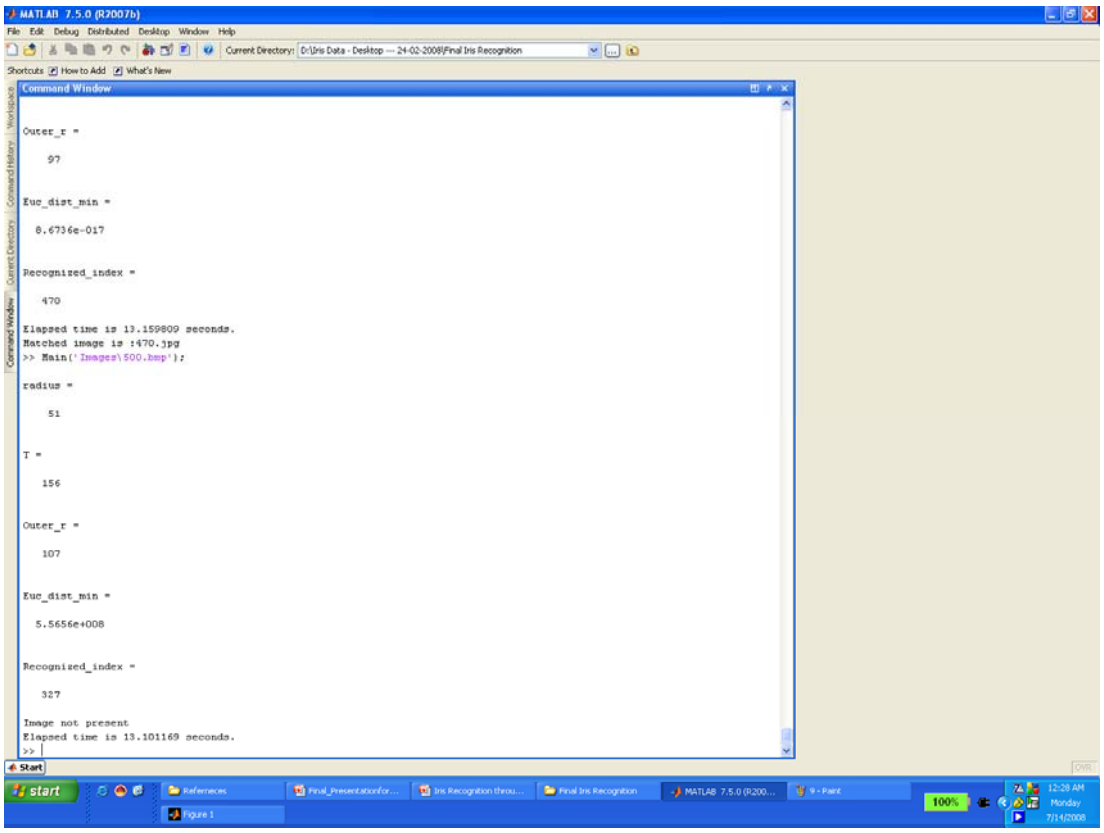


Figure 9.10: Results of testing of an image with 92 subjects

9.11 Comparison of Results with other Iris Recognition Methods

The algorithm was implemented using 2.8 GHZ Pentium 4 machine with Windows XP and MATLAB 7.5 as the development tool.

This section compares results of my proposed technique with libor masek's implementation of iris recognition techniques and iris recognition using neural networks. All the images are stored on hard drive to acquire the memory saving.

Firstly comparison for percentage of recognition with various Iris Recognition (IR) techniques is shown. The time and results of the proposed work and other techniques is displayed. Then the time requires for testing 10, 19, 29, 34, 45, 54, 64, 71, 88 and 92 subjects images e.g. 70, 120, 180, 240, 300, 360, 440, 500, 550 and 600 images respectively is calculated. Also the time per image testing is also displayed. 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.

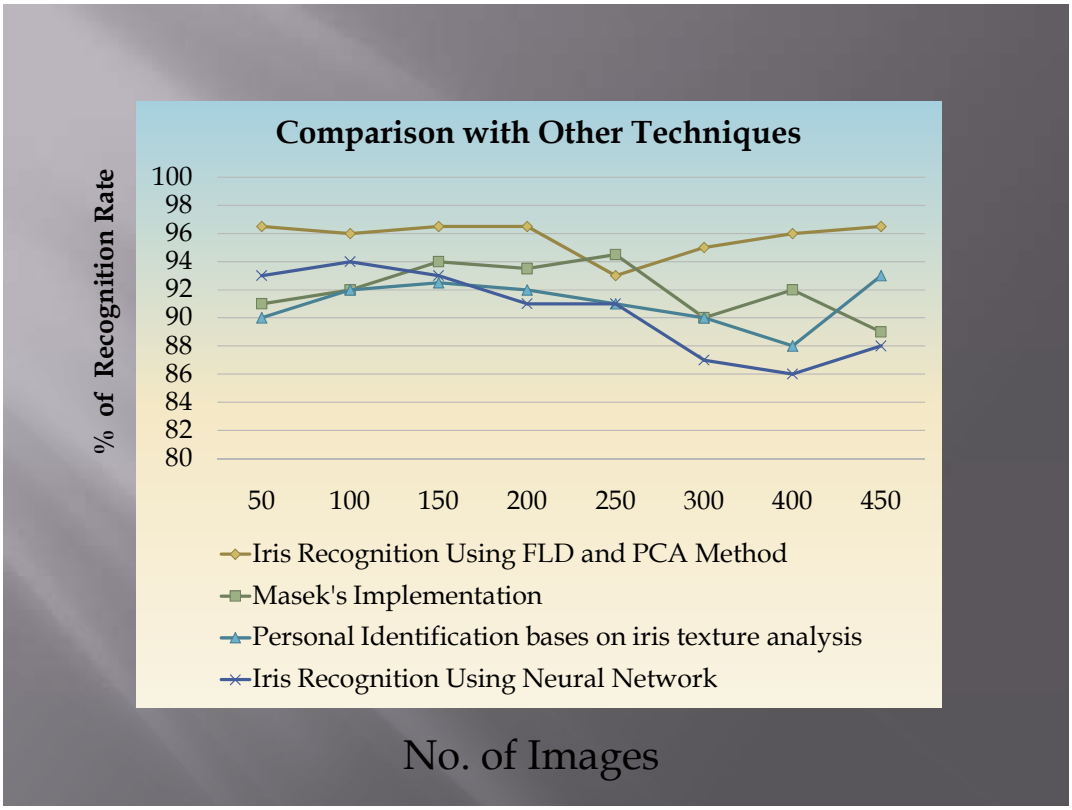


Figure 9.11: Percentage of correct decisions

The graph in figure 9.11 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.

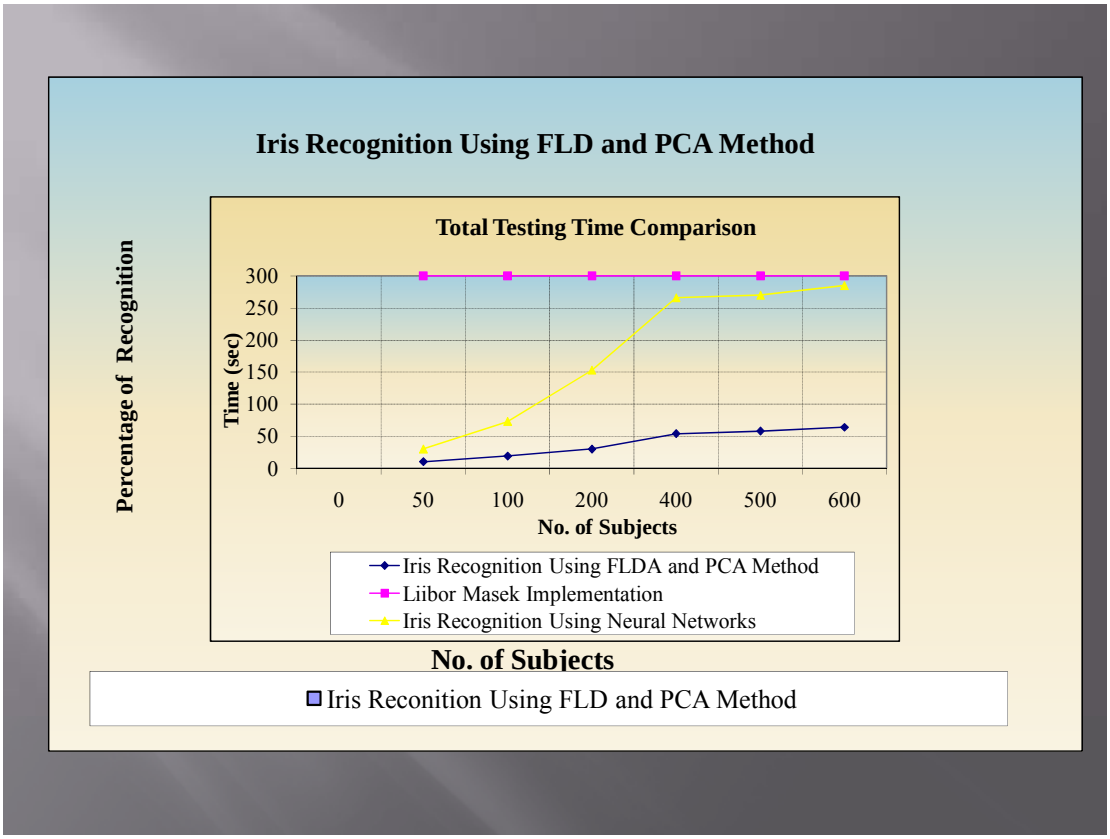


Figure 9.12: Total testing time comparison

The graph in figure 9.12 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.

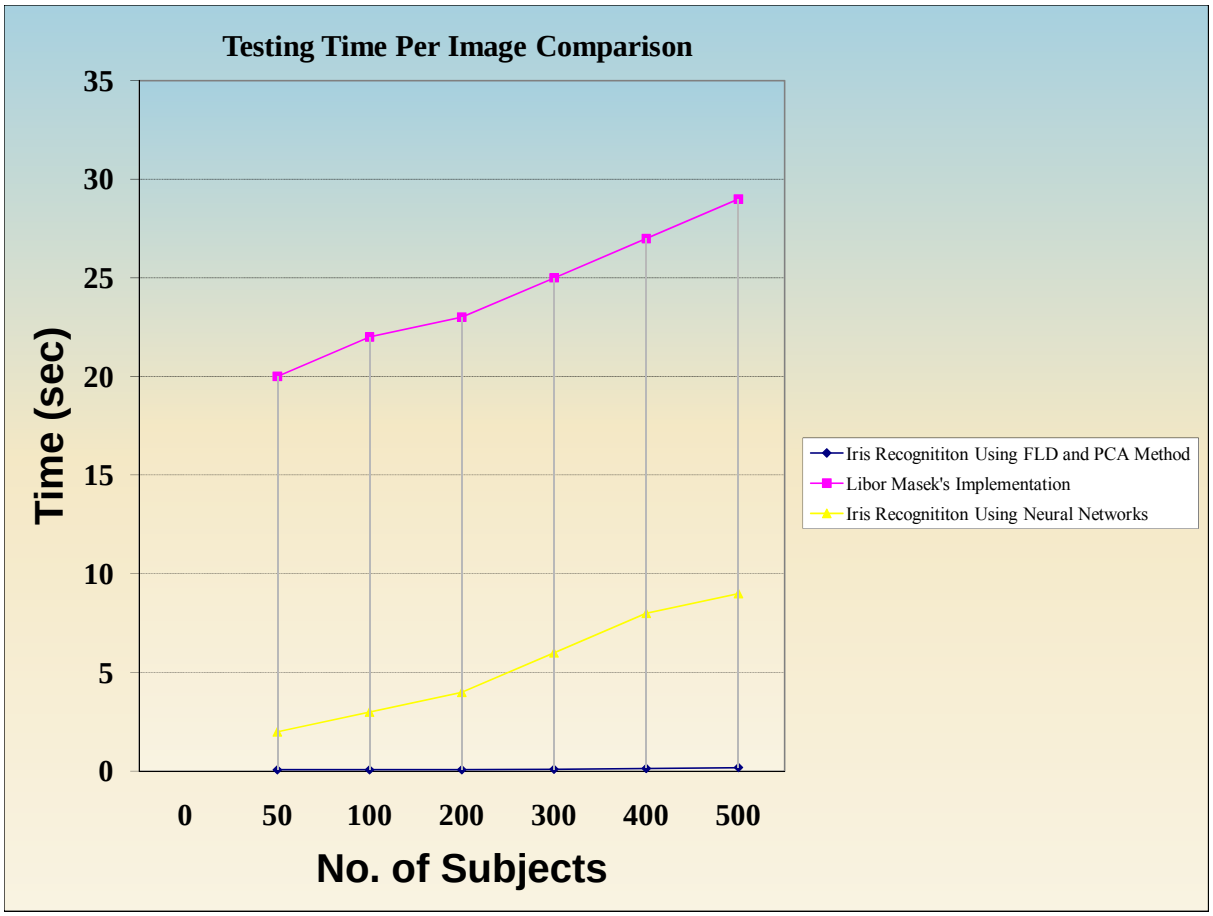


Figure 9.13: Testing time per image comparison

The graph in figure 9.13 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.

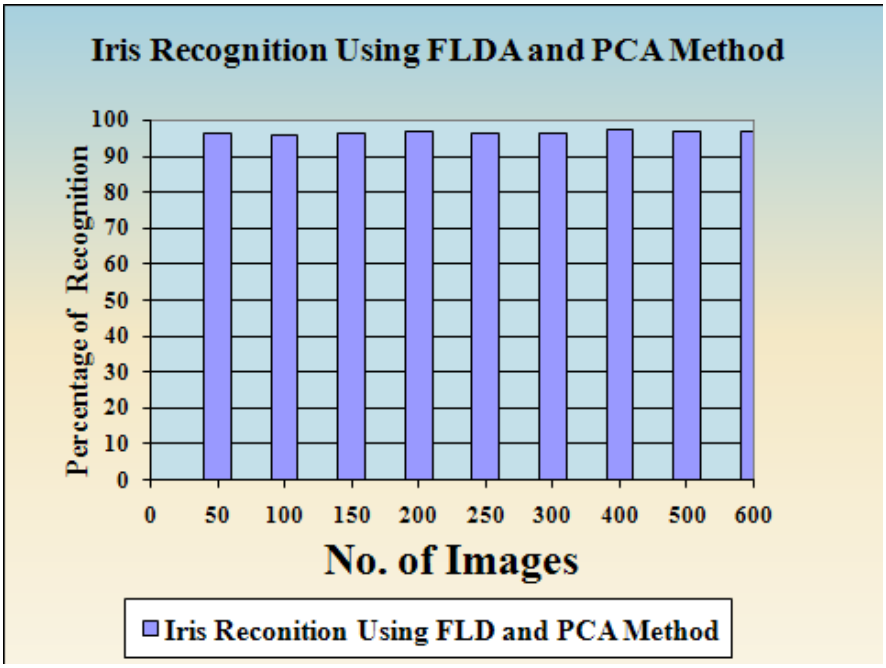


Figure: 9.14: percentage of recognition at various number of subjects

Figure 9.14 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.

9.12 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	97%	70%	65%

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 10: Research Achievements and Future Issues to Examine

10.1 Research Achievements

1. Time effective Pre-processing is done to incorporate faster processing times.
2. System Architecture is designed and optimized in such a way to achieve desired results
3. The effect of increasing subjects with recognition rate is analyzed

10.2 Future Issues to Examine.

Matlab is powerful software when it comes to mathematical operations but it uses a lot of vectors and matrix and these matrices and vectors uses too much memory, hard-disk and slowdown the processor unit of the computer. Thus implementing our project by using C#.net or Visual Basic.net to save memory and time. Also program will work independent from the Matlab.

Since we used a stand alone PC to control any gate, if we construct a Local Area Network (LAN) by keeping irises, user and log information on main server, ever PC may reach these files and may given access any gate on the LAN.

Also it may add specific time groups to indicate which user will be given access at which time user get access.

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.

Also following issues can be taken in consideration;

1. Can image capture technologies be improved?
2. Can iris color be utilized?
3. Larger database!

Chapter 11: Conclusion

For at least a century, it has been suggested that the iris can subserve 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.

The main functional components of extant iris-recognition systems consist of image acquisition, iris localization and pattern matching. In evaluating designs for these components, one must consider a wide range of technical issues. The most important among these are the physical nature of the iris, optics, image processing/analysis, and human factors. All these considerations must be combined to yield robust solutions even while incurring modest computational expense and compact design.

In conclusion, this report is giving detailed information about FLD analysis & PCA method and how to implement it by explaining on figures. Also it gives test result for iris recognition.

The FLD Analysis method appears to be the best at extrapolating and interpolating.

The thesis has presented an iris recognition system, which was tested using a database of grayscale eye images in order to verify the claimed performance of Iris Recognition Technology.

A new iris recognition system is provided in this paper that's implements Fisher Linear Discriminant Analysis Method in a new optimized manner which resulted in time reduction to perform the comparisons as well as provide more efficiency. In first step, we found the center of the pupil, after we have cut eye image according to center of the pupil. Therefore we have got clearer picture. Secondly, an automatic segmentation algorithm was presented, which would localize the iris region from unwanted parts of the

eye. Automatic segmentation was achieved, normalized template is calculated. From normalized template, the features are extracted.

Next, we applied FLD analysis and PCA method on the segmented iris image. So it gives the most accurate results. As a result, FLD analysis and PCA method is applicable for iris recognition system.

The iris is an ideal biometric feature for human identification. Although relatively young, the field of iris recognition has seen some great successes. Commercial implementations could become much more common in the future.

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.

References

[1] 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.

[2] 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.

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

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

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

[6] 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.

[7] 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.

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

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

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

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

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

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

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

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

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

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

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

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

[20] 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.

[21] 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.

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

[23] 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.

[24] 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.

[25] N. Ritter. Location of the pupil-iris border in slit-lamp images of the cornea. Proceedings of the International Conference on Image Analysis and Processing, 1999.

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

[27] A tutorial on Principal Component Analysis, Lindsey Smith (2002): http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

[28] Cooley, W.W. and Lohnes, P.R. Multivariate Data Analysis (John Wiley & Sons, Inc., New York, 1971).

[29] Rao, C.R. The Use and Interpretation of Principal Component Analysis in Applied Research. Sankhya A 26, 329 -358 (1964).

[30] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, \Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection", in European Conf. on Computer Vision, 1996, pp. 45-58.

[31] Moghaddam and Pentland, \Probabilistic visual learning for object detection", in Int. Conf. on Computer Vision, 1995, pp. 786{793.

[32] Pattern Classification (2nd ed.), R.O. Duda, P.E. Hart, D.H. Stork, Wiley Interscience, (2000). ISBN 0471056693

[33] Fisher, R.A. The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics, 7: 179-188 (1936) pdf file (http://www.library.adelaide.edu.au/digitised/fisher/138.pdf)

[34] Friedman, J.H. Regularized Discriminant Analysis. Journal of the American Statistical Association, (1989) pdf file (http://www.slac.stanford.edu/cgi-wrap/getdoc/slac-pub-4389.pdf)

[35] Mika, S. et al. Fisher Discriminant Analysis with Kernels. IEEE Conference on Neural Networks for SignalProcessing IX, (1999) gzipped ps file (http://www.first.gmd.de/~mika/MikRaeWesSchMue99.ps.gz)

[36] Tutorial about LDA from msstate.edu (http://www.isip.msstate.edu/publications/reports/isip_internal/1998/linear_discrim_analysis/lda_theory.pdf)

[37] LDA tutorial using MS Excel (http://people.revoledu.com/kardi/tutorial/LDA/index.html)

Appendix - A

Detailed Experimental Results:

Results Database								
Image No.	Euclidean Distance	Image Data Set Size	Time Elapsed Per Image Data Set Size (Seconds)	Time Elapsed Per Single Image (Seconds)	Threshold 'T' Value	Inner Radius	Outer Radius	Status
1	5.73E-28	**50**	0.531	**0.01062**	152.5	36	100	**Match**
2	1.77E-28	**50**	0.75	**0.015**	152.5	39.5	99	**Match**
3	5.02E-29	**50**	0.532	**0.01064**	150.5	38	87	**Match**
4	3.48E-27	**50**	0.531	**0.01062**	145.5	40	98	**Match**
5	4.56E-28	**50**	0.875	**0.0175**	160.5	39	109	**Match**
6	8.81E-29	**50**	0.532	**0.01064**	158.5	36	101	**Match**
7	1.23E-27	**50**	1.375	**0.0275**	122	50	105	**Match**
8	1.48E-27	**50**	0.547	**0.01094**	119.5	48.5	119	**Match**
9	5.63E-28	**50**	1.063	**0.02126**	115.5	46	97	**Match**
10	1.19E-27	**50**	0.547	**0.01094**	129	50	103	**Match**
11	9.29E-29	**50**	0.531	**0.01062**	117	50	101	**Match**
12	7.59E-28	**50**	0.531	**0.01062**	116	51	104	**Match**
13	1.08E-27	**50**	0.516	**0.01032**	120	52	105	**Match**
14	4.19E-27	**50**	0.5	**0.01**	139	42	95	**Match**
15	5.87E-29	**50**	0.563	**0.01126**	145.5	42.5	87	**Match**
16	2.04E-28	**50**	0.546	**0.01092**	162	39.5	79	**Match**
17	5.65E-28	**50**	0.515	**0.0103**	150.5	48	115	**Match**
18	5.87E-29	**50**	0.531	**0.01062**	148	51	103	**Match**
19	3.04E-27	**50**	0.516	**0.01032**	147.5	49	103	**Match**
20	9.31E-28	**50**	0.531	**0.01062**	149.5	50	104	**Match**
21	1.27E-28	**50**	0.547	**0.01094**	168.5	45	107	**Match**

22	6.95E-29	**50**	0.516	**0.01032**	168.5	43	105	**Match**
23	2.23E-29	**50**	0.547	**0.01094**	162.5	45	109	**Match**
24	4.05E-28	**50**	0.516	**0.01032**	167.5	46	133	**Match**
25	2.61E-28	**50**	0.562	**0.01124**	172	41	88	**Match**
26	2.38E-28	**50**	0.531	**0.01062**	169.5	43	107	**Match**
27	2.33E-28	**50**	0.516	**0.01032**	169	42	118	**Match**
28	2.57E-28	**50**	0.516	**0.01032**	169	42.5	109	**Match**
29	1.81E-28	**50**	0.5	**0.01**	158.5	43	101	**Match**
30	5.43E-28	**50**	0.5	**0.01**	161.5	42	103	**Match**
31	2.97E-28	**50**	0.516	**0.01032**	161.5	44	104	**Match**
32	4.23E-29	**50**	0.516	**0.01032**	161.5	42.5	95	**Match**
33	1.80E-28	**50**	0.515	**0.0103**	165.5	43	94	**Match**
34	5.75E-29	**50**	0.609	**0.01218**	171.5	39	83	**Match**
35	4.58E-29	**50**	0.531	**0.01062**	186.5	41	90	**Match**
36	1.54E-28	**50**	0.531	**0.01062**	185.5	41	106	**Match**
37	2.84E-28	**50**	0.516	**0.01032**	183.5	41.5	92	**Match**
38	4.05E-28	**50**	0.516	**0.01032**	186.5	38	96	**Match**
39	1.56E-27	**50**	0.546	**0.01092**	182	43	94	**Match**
40	3.32E-28	**50**	0.547	**0.01094**	189.5	40	103	**Match**
41	4.23E-28	**50**	0.516	**0.01032**	181.5	38.5	92	**Match**
42	2.38E-29	**50**	0.5	**0.01**	141.5	50.5	103	**Match**
43	2.70E-28	**50**	0.516	**0.01032**	143.5	47	97	**Match**
44	3.79E-29	**50**	0.5	**0.01**	138.5	49	98	**Match**
45	1.45E-28	**50**	0.515	**0.0103**	145.5	45	97	**Match**
46	5.92E-29	**50**	0.515	**0.0103**	134	45	96	**Match**
47	4.70E-28	**50**	0.531	**0.01062**	149.5	45	95	**Match**
48	1.15E-08	**50**	0.578	**0.01156**	169.5	47	103	**Not Match**
49	4.21E-28	**50**	0.5	**0.01**	145	45.5	111	**Match**
50	5.99E-28	**50**	0.515	**0.0103**	141.5	46	109	**Match**

51	1.87E-27	**102**	1.141	**0.011186275**	156.5	44	101	**Match**
52	1.38E-27	**102**	1.422	**0.013941176**	149.5	46	101	**Match**
53	1.55E-27	**102**	1.281	**0.012558824**	158.5	44	101	**Match**
54	8.97E-28	**102**	1.687	**0.016539216**	155	46	101	**Match**
55	8.96E-27	**102**	0.969	**0.0095**	114	54	115	**Match**
56	8.84E-28	**102**	1.469	**0.014401961**	106.5	53	112	**Match**
57	9.89E-28	**102**	1.375	**0.013480392**	111	50	115	**Match**
58	7.49E-28	**102**	1.141	**0.011186275**	113	51	113	**Match**
59	4.95E-28	**102**	1.531	**0.015009804**	110.5	51	115	**Match**
60	7.26E-27	**102**	1	**0.009803922**	109.5	53	110	**Match**
61	2.22E-27	**102**	1.531	**0.015009804**	105	54	118	**Match**
62	3.06E-28	**102**	1.813	**0.01777451**	151.5	43.5	91	**Match**
63	1.62E-27	**102**	1.844	**0.018078431**	147.5	42	89	**Match**
64	3.25E-27	**102**	1.203	**0.011794118**	149.5	43.5	88	**Match**
65	2.98E-28	**102**	0.969	**0.0095**	148.5	45	107	**Match**
66	5.60E-27	**102**	0.938	**0.009196078**	149.5	40.5	95	**Match**
67	4.46E-27	**102**	0.953	**0.009343137**	176.5	37	77	**Match**
68	1.83E-27	**102**	0.969	**0.0095**	174.5	39	85	**Match**
69	4.65E-28	**102**	0.969	**0.0095**	178	37	107	**Match**
70	4.58E-07	**102**	0.984	**0.009647059**	175	38	108	**Not Match**
71	1.09E-27	**102**	0.953	**0.009343137**	177.5	39	107	**Match**
72	3.43E-27	**102**	1.015	**0.00995098**	175.5	41.5	97	**Match**
73	2.72E-28	**102**	0.984	**0.009647059**	160.5	40	112	**Match**
74	7.54E-28	**102**	0.968	**0.009490196**	163	41	118	**Match**
75	5.51E-28	**102**	0.984	**0.009647059**	161.5	47	107	**Match**
76	9.56E-29	**102**	0.984	**0.009647059**	163	43	117	**Match**
77	9.91E-28	**102**	0.985	**0.009656863**	115.5	49	115	**Match**
78	2.52E-27	**102**	1	**0.009803922**	118	49	115	**Match**
79	2.13E-27	**102**	0.953	**0.009343137**	140.5	49	116	**Match**

80	2.41E-26	**102**	1.312	**0.012862745**	114	47	102	**Match**
81	4.33E-28	**102**	0.984	**0.009647059**	165	45.5	97	**Match**
82	1.13E-27	**102**	1.844	**0.018078431**	160.5	43	104	**Match**
83	6.81E-27	**102**	0.968	**0.009490196**	119.5	42	105	**Match**
84	2.59E-26	**102**	0.969	**0.0095**	117	43.5	106	**Match**
85	7.92E-27	**102**	1.672	**0.016392157**	112.5	48	102	**Match**
86	1.42E-26	**102**	0.984	**0.009647059**	170.5	45	101	**Match**
87	1.33E-27	**102**	2.015	**0.019754902**	164	49.5	101	**Match**
88	1.24E-28	**102**	0.968	**0.009490196**	160.5	44	103	**Match**
89	6.14E-28	**102**	0.984	**0.009647059**	161.5	48	96	**Match**
90	2.25E-28	**102**	0.984	**0.009647059**	175	44.5	92	**Match**
91	2.07E-27	**102**	0.969	**0.0095**	176.5	45	99	**Match**
92	5.51E-28	**102**	0.969	**0.0095**	179.5	43	98	**Match**
93	1.59E-08	**102**	0.969	**0.0095**	179.5	42	98	**Not Match**
94	5.00E-28	**102**	0.984	**0.009647059**	176.5	42.5	94	**Match**
95	1.74E-27	**102**	1.297	**0.012715686**	176.5	43.5	98	**Match**
96	3.47E-28	**102**	0.969	**0.0095**	181.5	40	98	**Match**
97	2.40E-27	**102**	0.953	**0.009343137**	140.5	43	102	**Match**
98	1.16E-27	**102**	0.985	**0.009656863**	127	44.5	98	**Match**
99	2.62E-27	**102**	0.984	**0.009647059**	130	44	103	**Match**
100	1.01E-26	**102**	0.937	**0.009186275**	**143.5**	43	105	**Match**
101	1.99E-27	**102**	0.969	**0.0095**	124	41.5	83	**Match**
102	2.45E-27	**102**	0.984	**0.009647059**	155.5	39	103	**Match**
103	3.56E-16	**158**	1.609	**0.010183544**	144	44	108	**Match**
104	1.48E-15	**158**	1.656	**0.010481013**	129	46.5	107	**Match**
105	3.83E-16	**158**	1.625	**0.01028481**	124.5	47	105	**Match**
106	3.13E-16	**158**	2.735	**0.017310127**	126	47	107	**Match**
107	5.42E-18	**158**	2.172	**0.013746835**	121.5	46	109	**Match**
108	1.06E-17	**158**	2.25	**0.014240506**	129	46	109	**Match**

109	8.67E-15	**158**	2.156	**0.01364557**	172.5	43	109	**Match**
110	1.53E-15	**158**	1.625	**0.01028481**	140	45	108	**Match**
111	3.33E-15	**158**	1.625	**0.01028481**	143.5	43	93	**Match**
112	6.28E+10	**158**	1.625	**0.01028481**	175.5	42	93	**Not Match**
113	8.88E-16	**158**	2.047	**0.012955696**	123	41	115	**Match**
114	2.00E-15	**158**	2.344	**0.014835443**	115	42	105	**Match**
115	4.10E-16	**158**	2.594	**0.016417722**	118	53	114	**Match**
116	3.36E-15	**158**	1.672	**0.010582278**	108	52	110	**Match**
117	1.06E-17	**158**	2.234	**0.014139241**	117	50	112	**Match**
118	1.35E-15	**158**	2.953	**0.018689873**	112	49.5	108	**Match**
119	8.88E-16	**158**	2.5	**0.015822785**	111.5	51	107	**Match**
120	5.00E-16	**158**	2.219	**0.014044304**	106	53	110	**Match**
121	4.50E-15	**158**	1.719	**0.010879747**	113	50	109	**Match**
122	5.55E-17	**158**	1.671	**0.010575949**	127	49	110	**Match**
123	5.55E-17	**158**	1.625	**0.01028481**	139	48	110	**Match**
124	1.25E-15	**158**	2.984	**0.018886076**	116	44.5	113	**Match**
125	2.81E-16	**158**	3.281	**0.020765823**	120	48	115	**Match**
126	2.62E-17	**158**	1.656	**0.010481013**	120	45	116	**Match**
127	3.47E-18	**158**	2.656	**0.016810127**	118	42	104	**Match**
128	1.39E-15	**158**	2.734	**0.017303797**	119	44.5	99	**Match**
129	9.16E-16	**158**	2.781	**0.017601266**	145.5	37.5	76	**Match**
130	1.48E-15	**158**	1.922	**0.012164557**	148.5	37	74	**Match**
131	2.51E-06	**158**	1.703	**0.010778481**	120	43.5	105	**Not Match**
132	5.86E-16	**158**	1.672	**0.010582278**	120	41	83	**Match**
133	6.80E-16	**158**	1.672	**0.010582278**	116	44	96	**Match**
134	4.50E-15	**158**	2.968	**0.01878481**	165.5	36	107	**Match**
135	2.08E-15	**158**	1.656	**0.010481013**	167.5	37	107	**Match**
136	2.08E-15	**158**	2.344	**0.014835443**	165.5	36.5	106	**Match**
137	1.39E-15	**158**	1.828	**0.01156962**	106.5	44	115	**Match**

138	2.00E-15	**158**	2.219	**0.014044304**	110	39	109	**Match**
139	5.00E-16	**158**	2.687	**0.017006329**	108	39.5	108	**Match**
140	3.55E-15	**158**	2.016	**0.012759494**	106	42.5	111	**Match**
141	5.01E-15	**158**	1.672	**0.010582278**	111	41.5	112	**Match**
142	7.99E-15	**158**	1.688	**0.010683544**	113	38	113	**Match**
143	2.17E-17	**158**	2.906	**0.018392405**	115	37.5	111	**Match**
144	4.62E-15	**158**	1.703	**0.010778481**	151.5	40	96	**Match**
145	1.70E-16	**158**	1.625	**0.01028481**	142	41	111	**Match**
146	1.39E-17	**158**	2.156	**0.01364557**	160.5	47	101	**Match**
147	1.46E-15	**158**	2.953	**0.018689873**	164.5	49	100	**Match**
148	1.47E-16	**158**	1.875	**0.011867089**	158.5	47	103	**Match**
149	1.47E-16	**158**	1.672	**0.010582278**	140.5	46	102	**Match**
150	4.20E-16	**158**	1.672	**0.010582278**	154.5	46.5	118	**Match**
151	1.05E-16	**158**	2.812	**0.017797468**	154.5	45.5	97	**Match**
152	1.39E-17	**158**	2.719	**0.017208861**	160.5	36	101	**Match**
153	4.20E-16	**158**	3.188	**0.020177215**	163.5	35	102	**Match**
154	2.87E-15	**158**	3.016	**0.019088608**	162	38	99	**Match**
155	3.47E-18	**158**	1.687	**0.010677215**	172	37.5	97	**Match**
156	2.53E-15	**158**	3.093	**0.019575949**	168.5	34	95	**Match**
157	5.75E-16	**158**	1.703	**0.010778481**	163	33	95	**Match**
158	1.27E-15	**158**	2.828	**0.017898734**	162.5	34	99	**Match**
159	4.01E-15	**192**	2.391	**0.012453125**	157.5	41.5	94	**Match**
160	1.70E-16	**192**	2.516	**0.013104167**	154.5	43	94	**Match**
161	5.01E-15	**192**	2.344	**0.012208333**	156	40	90	**Match**
162	1.00E-15	**192**	2.641	**0.013755208**	151.5	38	92	**Match**
163	8.88E-16	**192**	3.625	**0.018880208**	160.5	43	95	**Match**
164	1.09E-14	**192**	3.547	**0.018473958**	168	41.5	94	**Match**
165	1.84E-15	**192**	3.953	**0.020588542**	167.5	42	96	**Match**
166	2.00E-15	**192**	3.406	**0.017739583**	157.5	48.5	113	**Match**

167	2.17E-17	**192**	3.391	**0.017661458**	159.5	50	104	**Match**
168	3.23E-15	**192**	3.563	**0.018557292**	154.5	51.5	109	**Match**
169	1.95E-16	**192**	3.5	**0.018229167**	162.5	48.5	103	**Match**
170	8.34E-16	**192**	2.828	**0.014729167**	159.5	50.5	102	**Match**
171	1.47E-16	**192**	3.532	**0.018395833**	163.5	49.5	109	**Match**
172	3.47E-18	**192**	2.375	**0.012369792**	163.5	45.5	105	**Match**
173	3.73E-16	**192**	2.344	**0.012208333**	156.5	47	105	**Match**
174	1.44E-15	**192**	3.36	**0.0175**	157	47	95	**Match**
175	1.39E-15	**192**	2.985	**0.015546875**	159.5	41	102	**Match**
176	8.67E-17	**192**	3.312	**0.01725**	158.5	42	102	**Match**
177	7.79E-15	**192**	3.203	**0.016682292**	159.5	41	99	**Match**
178	3.33E-15	**192**	3.11	**0.016197917**	166.5	38	103	**Match**
179	8.88E-06	**192**	3.063	**0.015953125**	175	36.5	101	**Not Match**
180	7.81E-16	**192**	2.313	**0.012046875**	171.5	36.5	108	**Match**
181	8.88E-16	**192**	2.532	**0.0131875**	166.5	37	91	**Match**
182	7.81E-16	**192**	2.344	**0.012208333**	172	37.5	96	**Match**
183	6.80E-16	**192**	2.343	**0.012203125**	171.5	38	94	**Match**
184	1.25E-16	**192**	2.313	**0.012046875**	169.5	40.5	96	**Match**
185	4.50E-15	**192**	2.312	**0.012041667**	171.5	39	79	**Match**
186	7.03E-17	**192**	3.14	**0.016354167**	117	43	111	**Match**
187	5.86E-16	**192**	2.343	**0.012203125**	112	45	92	**Match**
188	1.32E-15	**192**	2.36	**0.012291667**	180	39	88	**Match**
189	1.60E-14	**192**	2.406	**0.01253125**	178.5	40	81	**Match**
190	1.09E-14	**192**	2.344	**0.012208333**	179	39	91	**Match**
191	6.42E-14	**192**	2.375	**0.012369792**	116.5	41	90	**Match**
192	3.20E-14	**192**	2.531	**0.013182292**	122	42	94	**Match**
193	1.36E-16	**242**	3.359	**0.013880165**	162.5	50	105	**Match**
194	7.55E-16	**242**	3.359	**0.013880165**	159.5	50.5	108	**Match**
195	3.12E-17	**242**	3.328	**0.013752066**	162.5	49.5	105	**Match**

196	3.47E-16	**242**	3.344	**0.013818182**	165.5	50	105	**Match**
197	9.45E-16	**242**	3.407	**0.014078512**	164.5	50	105	**Match**
198	1.06E-15	**242**	3.391	**0.014012397**	168.5	49	98	**Match**
199	1.89E-16	**242**	3.328	**0.013752066**	166.5	53.5	107	**Match**
200	8.40E-16	**242**	3.281	**0.013557851**	161	51.5	103	**Match**
201	1.15E-16	**242**	3.438	**0.014206612**	163.5	48	98	**Match**
202	3.47E-16	**242**	3.297	**0.013623967**	168.5	38	86	**Match**
203	1.84E-15	**242**	3.296	**0.013619835**	173.5	51	103	**Match**
204	6.56E-16	**242**	3.265	**0.013491736**	180	45	112	**Match**
205	7.81E-18	**242**	3.734	**0.015429752**	174.5	47.5	100	**Match**
206	3.83E-16	**242**	3.281	**0.013557851**	157.5	44	92	**Match**
207	1.12E-16	**242**	3.312	**0.01368595**	158.5	43	94	**Match**
208	1.61E-16	**242**	3.329	**0.013756198**	157.5	44	115	**Match**
209	1.82E-16	**242**	3.344	**0.013818182**	164	39.5	109	**Match**
210	3.66E-17	**242**	3.391	**0.014012397**	161.5	40	98	**Match**
211	1.72E-05	**242**	3.375	**0.013946281**	156.5	41	108	**Not Match**
212	1.04E-15	**242**	3.375	**0.013946281**	165.5	42.5	86	**Match**
213	3.72E-15	**242**	3.328	**0.013752066**	161.5	43	95	**Match**
214	6.32E-16	**242**	3.39	**0.014008264**	175.5	40.5	101	**Match**
215	2.54E-16	**242**	3.391	**0.014012397**	171.5	42	97	**Match**
216	5.42E-16	**242**	3.265	**0.013491736**	140.5	41	99	**Match**
217	7.05E-16	**242**	3.344	**0.013818182**	170.5	39	98	**Match**
218	2.08E-16	**242**	3.359	**0.013880165**	173	38	96	**Match**
219	8.25E-17	**242**	3.36	**0.013884298**	177.5	37	97	**Match**
220	1.90E-15	**242**	3.391	**0.014012397**	168.5	46	92	**Match**
221	5.86E-16	**242**	3.313	**0.013690083**	123	39.5	110	**Match**
222	1.00E-15	**242**	3.313	**0.013690083**	170.5	37	101	**Match**
223	5.21E-16	**242**	3.437	**0.014202479**	170.5	47.5	103	**Match**
224	2.39E-17	**242**	3.343	**0.01381405**	171.5	45	97	**Match**

225	1.76E-15	**242**	3.609	**0.014913223**	153.5	49	105	**Match**
226	1.58E-16	**242**	3.422	**0.014140496**	159.5	48.5	99	**Match**
227	1.00E-15	**242**	3.312	**0.01368595**	131	47	109	**Match**
228	1.25E-16	**242**	3.187	**0.013169421**	136	46.5	107	**Match**
229	2.00E-15	**242**	3.328	**0.013752066**	155.5	45	109	**Match**
230	1.12E-15	**242**	3.219	**0.013301653**	129	46.5	114	**Match**
231	3.12E-17	**242**	3.328	**0.013752066**	162.5	47.5	107	**Match**
232	5.55E-17	**242**	3.219	**0.013301653**	148.5	49.5	104	**Match**
233	5.21E-16	**242**	3.344	**0.013818182**	149.5	49.5	112	**Match**
234	5.21E-16	**242**	3.313	**0.013690083**	148.5	47	98	**Match**
235	2.62E-17	**242**	3.485	**0.014400826**	150.5	45	101	**Match**
236	2.51E-16	**242**	3.36	**0.013884298**	118	50.5	102	**Match**
237	2.39E-17	**242**	3.282	**0.013561983**	116	51	103	**Match**
238	2.29E-16	**242**	3.281	**0.013557851**	142	49	100	**Match**
239	2.35E-15	**242**	3.438	**0.014206612**	180	46.5	101	**Match**
240	1.39E-17	**242**	3.188	**0.013173554**	144.5	48	102	**Match**
241	6.80E-16	**242**	3.375	**0.013946281**	180.5	47	98	**Match**
242	2.22E-16	**242**	3.391	**0.014012397**	177.5	47	105	**Match**
243	8.67E-19	**292**	8.219	**0.02814726**	122	46	103	**Match**
244	3.30E-16	**292**	8.515	**0.029160959**	136.5	43	101	**Match**
245	1.84E-15	**292**	8.266	**0.028308219**	126	44	116	**Match**
246	1.70E-16	**292**	9.25	**0.031678082**	125	41	103	**Match**
247	2.72E-15	**292**	8.25	**0.028253425**	131	42	109	**Match**
248	3.47E-16	**292**	8.344	**0.028575342**	137	42	112	**Match**
249	2.17E-17	**292**	8.672	**0.02969863**	123	42	106	**Match**
250	1.19E-15	**292**	8.281	**0.028359589**	174.5	45	99	**Match**
251	5.86E-16	**292**	8.219	**0.02814726**	155	46.5	109	**Match**
252	1.39E-17	**292**	8.219	**0.02814726**	156.5	48	98	**Match**
253	1.39E-17	**292**	8.844	**0.030287671**	154.5	49	102	**Match**

254	1.12E-15	**292**	8.485	**0.029058219**	157	50.5	103	**Match**
255	6.80E-16	**292**	8.437	**0.028893836**	159.5	46	102	**Match**
256	1.39E-15	**292**	8.281	**0.028359589**	160.5	45.5	102	**Match**
257	8.34E-16	**292**	8.219	**0.02814726**	159.5	47	105	**Match**
258	2.17E-17	**292**	8.453	**0.02894863**	168	43	106	**Match**
259	5.42E-16	**292**	8.391	**0.028736301**	168.5	39	111	**Match**
260	1.06E-15	**292**	8.422	**0.028842466**	169.5	43.5	89	**Match**
261	3.47E-16	**292**	8.578	**0.029376712**	166	41	106	**Match**
262	1.32E-15	**292**	8.562	**0.029321918**	168.5	37	74	**Match**
263	8.34E-16	**292**	8.375	**0.028681507**	168.5	41	105	**Match**
264	2.35E-15	**292**	8.359	**0.028626712**	169.5	40	104	**Match**
265	4.25E-17	**292**	8.531	**0.029215753**	166.5	34	76	**Match**
266	5.01E-15	**292**	9.969	**0.034140411**	168.5	39	78	**Match**
267	1.12E-15	**292**	8.672	**0.02969863**	174.5	34	72	**Match**
268	1.39E-17	**292**	8.422	**0.028842466**	129	39	78	**Match**
269	5.00E-16	**292**	8.375	**0.028681507**	188.5	39.5	105	**Match**
270	2.22E-16	**292**	8.406	**0.028787671**	172.5	43	101	**Match**
271	1.46E-15	**292**	8.39	**0.028732877**	164	41	91	**Match**
272	7.03E-17	**292**	8.563	**0.029325342**	176.5	43	99	**Match**
273	2.17E-15	**292**	8.406	**0.028787671**	174.5	42	99	**Match**
274	1.39E-07	**292**	8.375	**0.028681507**	172.5	42	87	**Not Match**
275	3.55E-15	**292**	8.797	**0.030126712**	160.5	43.5	99	**Match**
276	1.70E-16	**292**	8.359	**0.028626712**	157.5	39	101	**Match**
277	1.78E-15	**292**	8.266	**0.028308219**	154.5	41.5	99	**Match**
278	4.79E-16	**292**	8.765	**0.030017123**	160	39.5	109	**Match**
279	1.47E-16	**292**	8.312	**0.028465753**	158.5	40.5	99	**Match**
280	5.55E-17	**292**	8.281	**0.028359589**	160.5	41.5	105	**Match**
281	1.00E-15	**292**	8.843	**0.030284247**	167	38.5	77	**Match**
282	1.68E-15	**292**	8.329	**0.028523973**	153	48	116	**Match**

283	5.53E-16	**292**	8.156	**0.027931507**	140.5	47.5	105	**Match**
284	2.19E-16	**292**	8.703	**0.029804795**	140	45	106	**Match**
285	1.39E-17	**292**	8.25	**0.028253425**	142	45	106	**Match**
286	2.26E-15	**292**	8.5	**0.029109589**	165.5	47	112	**Match**
287	3.72E-15	**292**	8.969	**0.030715753**	164.5	46.5	109	**Match**
288	9.85E-16	**292**	8.375	**0.028681507**	166.5	44	109	**Match**
289	1.16E-15	**292**	8.406	**0.028787671**	167.5	41	101	**Match**
290	1.06E-17	**292**	9.453	**0.032373288**	163.5	47.5	109	**Match**
291	2.51E-16	**292**	8.391	**0.028736301**	164.5	44.5	108	**Match**
292	4.62E-15	**292**	8.907	**0.030503425**	167.5	44	111	**Match**
293	6.56E-16	**348**	12	**0.034482759**	149.5	42.5	109	**Match**
294	3.78E-15	**348**	11.875	**0.034123563**	114	57	114	**Match**
295	1.00E-15	**348**	12.062	**0.03466092**	113	54	112	**Match**
296	2.22E-16	**348**	12.031	**0.034571839**	116	53	106	**Match**
297	5.55E-15	**348**	16.844	**0.048402299**	118	54	109	**Match**
298	1.70E-16	**348**	12.031	**0.034571839**	114	53	108	**Match**
299	3.12E-17	**348**	12.125	**0.034841954**	118	52.5	109	**Match**
300	9.45E-16	**348**	12.047	**0.034617816**	125	56.5	119	**Match**
301	1.55E-15	**348**	12.078	**0.034706897**	172	42	103	**Match**
302	1.11E-14	**348**	12.094	**0.034752874**	172	42	97	**Match**
303	2.53E-15	**348**	11.641	**0.033451149**	171.5	39	80	**Match**
304	1.20E-16	**348**	11.859	**0.034077586**	175.5	38	103	**Match**
305	5.45E-15	**348**	12	**0.034482759**	166.5	51	102	**Match**
306	9.16E-16	**348**	11.609	**0.033359195**	137	52	114	**Match**
307	1.06E-17	**348**	11.609	**0.033359195**	162	51	107	**Match**
308	2.39E-17	**348**	11.578	**0.033270115**	161.5	52	107	**Match**
309	8.20E-16	**348**	11.922	**0.034258621**	163.5	51	106	**Match**
310	4.49E-16	**348**	12.219	**0.035112069**	164.5	49	109	**Match**
311	5.00E-16	**348**	12.079	**0.03470977**	162.5	48	109	**Match**

312	1.39E-15	**348**	12.625	**0.036278736**	161.5	47	105	**Match**
313	2.02E-16	**348**	11.75	**0.033764368**	157.5	47	106	**Match**
314	4.15E-16	**348**	12.359	**0.035514368**	162.5	46	104	**Match**
315	5.00E-16	**348**	12.015	**0.034525862**	157.5	45	105	**Match**
316	3.78E-15	**348**	12.375	**0.035560345**	158	47	105	**Match**
317	1.80E-14	**348**	11.985	**0.034439655**	155.5	51	104	**Match**
318	3.47E-18	**348**	12.125	**0.034841954**	156.5	52	105	**Match**
319	1.25E-15	**348**	12.047	**0.034617816**	180.5	45.5	96	**Match**
320	1.39E-15	**348**	11.922	**0.034258621**	179	41	92	**Match**
321	3.12E-07	**348**	12.875	**0.036997126**	178	42	93	**Not Match**
322	6.80E-16	**348**	11.875	**0.034123563**	181.5	40	93	**Match**
323	6.72E-15	**348**	11.937	**0.034301724**	177.5	43	100	**Match**
324	6.57E-15	**348**	11.687	**0.033583333**	181	43	96	**Match**
325	1.80E-14	**348**	12.984	**0.037310345**	169.5	47	105	**Match**
326	3.55E-15	**348**	12.188	**0.035022989**	145.5	45	99	**Match**
327	3.95E-17	**348**	12.047	**0.034617816**	174.5	47	103	**Match**
328	4.90E-15	**348**	11.765	**0.033807471**	172.5	44	103	**Match**
329	1.25E-15	**348**	11.937	**0.034301724**	169.5	42	97	**Match**
330	2.35E-15	**348**	11.704	**0.033632184**	139.5	42	107	**Match**
331	9.45E-16	**348**	11.765	**0.033807471**	150.5	45	104	**Match**
332	3.17E-15	**348**	11.656	**0.033494253**	145.5	48	108	**Match**
333	6.80E-16	**348**	11.656	**0.033494253**	140.5	42	120	**Match**
334	3.23E-15	**348**	11.672	**0.03354023**	132	45	103	**Match**
335	2.88E-15	**348**	11.657	**0.033497126**	140	47	108	**Match**
336	1.54E-15	**348**	11.735	**0.033721264**	153.5	50	109	**Match**
337	1.17E-15	**348**	11.719	**0.033675287**	108	49	119	**Match**
338	1.59E-15	**348**	11.766	**0.033810345**	107	49.5	132	**Match**
339	8.67E-17	**348**	11.75	**0.033764368**	104.5	51	118	**Match**
340	4.20E-16	**348**	11.719	**0.033675287**	104	51.5	118	**Match**

341	1.33E-14	**348**	11.812	**0.033942529**	106	50	114	**Match**
342	3.30E-16	**348**	11.718	**0.033672414**	161.5	41	106	**Match**
343	3.12E-15	**348**	11.75	**0.033764368**	160	43	111	**Match**
344	8.88E-16	**348**	11.672	**0.03354023**	160.5	43	105	**Match**
345	8.88E-16	**348**	11.61	**0.033362069**	160.5	43	103	**Match**
346	1.39E-17	**348**	11.579	**0.033272989**	168	42	106	**Match**
347	1.47E-14	**348**	11.719	**0.033675287**	137	42	105	**Match**
348	1.12E-15	**348**	11.656	**0.033494253**	118	43	104	**Match**
349	6.58E-42	**392**	17.141	**0.043727041**	169.5	36.5	82	**Match**
350	8.40E-42	**392**	18.407	**0.046956633**	172	36	73	**Match**
351	4.11E-42	**392**	12.953	**0.033043367**	168.5	39.5	106	**Match**
352	7.42E-42	**392**	15.656	**0.039938776**	162.5	39	110	**Match**
353	1.12E-41	**392**	17.328	**0.044204082**	172.5	37	76	**Match**
354	5.12E-42	**392**	17.484	**0.044602041**	136.5	52	105	**Match**
355	7.37E-42	**392**	17.562	**0.04480102**	110	51	111	**Match**
356	1.34E-41	**392**	17.454	**0.04452551**	162.5	50	104	**Match**
357	4.05E-42	**392**	17.578	**0.044841837**	108.5	51	111	**Match**
358	5.21E-42	**392**	17.312	**0.044163265**	136.5	50	105	**Match**
359	5.50E-42	**392**	17.594	**0.044882653**	114	50	103	**Match**
360	4.45E-42	**392**	17.656	**0.045040816**	114	51.5	109	**Match**
361	3.97E-42	**392**	17.75	**0.045280612**	156.5	47	103	**Match**
362	3.62E-42	**392**	17.313	**0.044165816**	164.5	45	91	**Match**
363	3.29E-42	**392**	17.625	**0.044961735**	153.5	50	101	**Match**
364	2.24E-42	**392**	17.296	**0.044122449**	142.5	53	106	**Match**
365	2.96E-42	**392**	17.546	**0.044760204**	152.5	45	104	**Match**
366	7.61E-04	**392**	17.312	**0.044163265**	138.5	45.5	101	**Not Match**
367	2.48E-42	**392**	17.516	**0.044683673**	144.5	47	101	**Match**
368	2.70E-42	**392**	17.609	**0.044920918**	173.5	46	107	**Match**
369	5.12E-42	**392**	17.688	**0.045122449**	174.5	47	101	**Match**

370	3.09E-42	**392**	17.328	**0.044204082**	175	45.5	98	**Match**
371	3.25E-42	**392**	17.813	**0.045441327**	174.5	46	95	**Match**
372	5.26E-42	**392**	17.157	**0.043767857**	175	47.5	101	**Match**
373	2.43E-42	**392**	17.5	**0.044642857**	145.5	44	109	**Match**
374	4.09E-42	**392**	17.704	**0.045163265**	145.5	44	99	**Match**
375	3.74E-42	**392**	17.438	**0.044484694**	143.5	46	98	**Match**
376	5.39E-42	**392**	17.953	**0.045798469**	150.5	47	97	**Match**
377	3.81E-42	**392**	19.11	**0.04875**	144.5	45	98	**Match**
378	2.74E-42	**392**	17.328	**0.044204082**	175.5	44	104	**Match**
379	3.39E-42	**392**	17.734	**0.045239796**	140.5	47	98	**Match**
380	2.84E-42	**392**	17.281	**0.044084184**	113	52	105	**Match**
381	3.94E-42	**392**	17.469	**0.044563776**	121	51	104	**Match**
382	7.37E-42	**392**	17.515	**0.044681122**	112	51	105	**Match**
383	5.51E-42	**392**	17.891	**0.045640306**	113.5	50	103	**Match**
384	3.41E-42	**392**	17.594	**0.044882653**	113	51	111	**Match**
385	2.42E-42	**392**	18.094	**0.046158163**	118	51	105	**Match**
386	2.62E-42	**392**	17.609	**0.044920918**	182	44	105	**Match**
387	8.08E-04	**392**	19.188	**0.04894898**	177.5	42	100	**Not Match**
388	1.54E-41	**392**	19.438	**0.049586735**	175	40.5	95	**Match**
389	7.98E-42	**392**	18.109	**0.046196429**	179.5	44	102	**Match**
390	4.14E-42	**392**	17.843	**0.045517857**	178	43	103	**Match**
391	1.63E-41	**392**	17.922	**0.045719388**	135	41	115	**Match**
392	8.05E-42	**392**	17.672	**0.045081633**	120	42	115	**Match**
393	1.06E-41	**444**	21.797	**0.049092342**	139	45	90	**Match**
394	1.84E-41	**444**	21.547	**0.048529279**	125	44.5	98	**Match**
395	4.17E-41	**444**	22.203	**0.050006757**	171.5	45	90	**Match**
396	2.27E-41	**444**	21.829	**0.049164414**	138.5	43	107	**Match**
397	3.63E-41	**444**	22.047	**0.049655405**	167.5	43	88	**Match**
398	2.95E-41	**444**	20.563	**0.046313063**	142.5	47	101	**Match**

399	3.36E-41	**444**	22.328	**0.050288288**	164.5	45	94	**Match**
400	1.44E-41	**444**	21.578	**0.048599099**	143	49.5	111	**Match**
401	1.03E-41	**444**	22.704	**0.051135135**	117	49	119	**Match**
402	1.69E-41	**444**	22.75	**0.051238739**	155.5	48	109	**Match**
403	3.97E-07	**444**	22.156	**0.049900901**	159.5	45	109	**Not Match**
404	4.91E-41	**444**	22.625	**0.050957207**	113	53	109	**Match**
405	2.45E-41	**444**	22.468	**0.050603604**	122	48	97	**Match**
406	2.08E-41	**444**	22.234	**0.050076577**	115	49	105	**Match**
407	1.75E-41	**444**	22.922	**0.051626126**	113.5	49	103	**Match**
408	2.17E-41	**444**	21.625	**0.048704955**	116	49	105	**Match**
409	4.18E-41	**444**	23.328	**0.052540541**	109	48	99	**Match**
410	4.13E-41	**444**	23.219	**0.052295045**	111	47	95	**Match**
411	3.77E-41	**444**	22.062	**0.049689189**	161.5	40	106	**Match**
412	2.72E-41	**444**	22.797	**0.051344595**	166.5	37	109	**Match**
413	3.29E-41	**444**	22.094	**0.049761261**	164	40	107	**Match**
414	4.73E-41	**444**	22.125	**0.049831081**	158.5	39	106	**Match**
415	6.81E-41	**444**	22.844	**0.05145045**	165	37	98	**Match**
416	7.41E-41	**444**	22.25	**0.050112613**	157.5	39	105	**Match**
417	2.01E-41	**444**	22.984	**0.051765766**	168.5	37	108	**Match**
418	6.36E-41	**444**	22.156	**0.049900901**	104	40	109	**Match**
419	4.23E-41	**444**	22.718	**0.051166667**	110	42	92	**Match**
420	3.32E-41	**444**	22.937	**0.05165991**	109.5	40	111	**Match**
421	7.05E-41	**444**	22.25	**0.050112613**	111	42	109	**Match**
422	4.36E-08	**444**	22.282	**0.050184685**	112	41	111	**Not Match**
423	2.81E-41	**444**	23.156	**0.052153153**	107	40	108	**Match**
424	2.19E-41	**444**	22.313	**0.050254505**	174.5	48	105	**Match**
425	1.73E-41	**444**	22.328	**0.050288288**	169.5	46	104	**Match**
426	3.84E-41	**444**	23.094	**0.052013514**	169.5	45	105	**Match**
427	2.33E-41	**444**	22.422	**0.0505**	170.5	46	105	**Match**

428	2.06E-41	**444**	15.562	**0.03504955**	170.5	47	105	**Match**
429	1.67E-41	**444**	22.015	**0.049583333**	135	55	110	**Match**
430	7.49E-42	**444**	24.094	**0.054265766**	153.5	50.5	105	**Match**
431	1.77E-41	**444**	22.156	**0.049900901**	132.5	55	110	**Match**
432	1.31E-41	**444**	22.89	**0.051554054**	149	53	106	**Match**
433	4.89E-41	**444**	24.047	**0.05415991**	120	55	119	**Match**
434	2.62E-41	**444**	22.109	**0.049795045**	153.5	53.5	107	**Match**
435	8.03E-42	**444**	22.937	**0.05165991**	153.5	53.5	107	**Match**
436	1.68E-41	**444**	22.25	**0.050112613**	154.5	42	97	**Match**
437	4.00E-41	**444**	22.406	**0.050463964**	157.5	40	91	**Match**
438	3.85E-41	**444**	23.015	**0.051835586**	160.5	46	101	**Match**
439	2.20E-41	**444**	22.094	**0.049761261**	148.5	45	91	**Match**
440	2.08E-41	**444**	22.219	**0.050042793**	147.5	45	94	**Match**
441	2.39E-41	**444**	23.969	**0.053984234**	165.5	42	97	**Match**
442	4.28E-41	**444**	22.969	**0.051731982**	158.5	42.5	97	**Match**
443	1.92E-41	**444**	22.156	**0.049900901**	163.5	45	98	**Match**
444	4.88E-41	**444**	24.578	**0.055355856**	156.5	44	96	**Match**
445	2.90E-41	**500**	25.953	**0.051906**	168.5	35	97	**Match**
446	1.36E-05	**500**	26.187	**0.052374**	162.5	38	108	**Not Match**
447	5.55E-41	**500**	26.906	**0.053812**	162.5	37	75	**Match**
448	8.15E-41	**500**	27.36	**0.05472**	151.5	35	102	**Match**
449	1.04E-40	**500**	26.265	**0.05253**	138	42.5	119	**Match**
450	8.55E-41	**500**	26.031	**0.052062**	133.5	36.5	105	**Match**
451	1.20E-40	**500**	26.469	**0.052938**	151.5	34	83	**Match**
452	1.37E-40	**500**	26.953	**0.053906**	150	33.5	94	**Match**
453	5.70E-41	**500**	26.125	**0.05225**	163.5	37	97	**Match**
454	1.58E-40	**500**	26.532	**0.053064**	159	37	101	**Match**
455	1.58E-40	**500**	26.656	**0.053312**	161.5	37	100	**Match**
456	6.42E-41	**500**	27.047	**0.054094**	164.5	36.5	102	**Match**

457	5.72E-41	**500**	26.563	**0.053126**	169.5	36	106	**Match**
458	8.82E-41	**500**	26.688	**0.053376**	165	38.5	101	**Match**
459	1.28E-40	**500**	26.969	**0.053938**	152.5	36	91	**Match**
460	7.46E-41	**500**	26.625	**0.05325**	170.5	38.5	96	**Match**
461	1.66E-40	**500**	26.781	**0.053562**	164.5	40	94	**Match**
462	8.02E-41	**500**	27.344	**0.054688**	165.5	38	91	**Match**
463	3.74E-40	**500**	27.172	**0.054344**	157.5	37.5	92	**Match**
464	3.06E-40	**500**	27.204	**0.054408**	163.5	38.5	94	**Match**
465	5.45E-41	**500**	27.141	**0.054282**	166.5	38	93	**Match**
466	8.41E-41	**500**	26.922	**0.053844**	170.5	42	97	**Match**
467	2.41E-41	**500**	27	**0.054**	173.5	43.5	104	**Match**
468	1.61E-40	**500**	27.078	**0.054156**	175.5	39	97	**Match**
469	7.91E-41	**500**	27.046	**0.054092**	172.5	43	93	**Match**
470	4.18E-41	**500**	27.047	**0.054094**	175.5	40	97	**Match**
471	4.24E-41	**500**	26.968	**0.053936**	167.5	43	97	**Match**
472	8.10E-41	**500**	26.89	**0.05378**	173.5	40	84	**Match**
473	6.40E-41	**500**	26.672	**0.053344**	146.5	53	106	**Match**
474	9.35E-41	**500**	30.625	**0.06125**	138.5	52	105	**Match**
475	4.39E-41	**500**	28.937	**0.057874**	157.5	51	109	**Match**
476	4.81E-41	**500**	26.89	**0.05378**	151.5	53.5	108	**Match**
477	5.75E-41	**500**	26.968	**0.053936**	167.5	51	102	**Match**
478	1.12E-40	**500**	26.859	**0.053718**	176.5	41	101	**Match**
479	9.38E-41	**500**	27.375	**0.05475**	169.5	43.5	100	**Match**
480	3.98E-41	**500**	26.969	**0.053938**	148.5	46	109	**Match**
481	7.86E-41	**500**	26.797	**0.053594**	149.5	48	112	**Match**
482	3.87E-41	**500**	26.86	**0.05372**	151.5	44.5	114	**Match**
483	9.63E-41	**500**	26.86	**0.05372**	154.5	43	101	**Match**
484	7.43E-09	**500**	26.828	**0.053656**	128	51	105	**Not Match**
485	7.35E-41	**500**	28.454	**0.056908**	172.5	45	90	**Match**

486	6.59E-41	**500**	28.297	**0.056594**	117	46	109	**Match**
487	1.04E-40	**500**	27.156	**0.054312**	118.5	47	101	**Match**
488	1.11E-40	**500**	27.203	**0.054406**	123	44	91	**Match**
489	1.42E-40	**500**	27	**0.054**	171.5	42.5	86	**Match**
490	1.17E-40	**500**	26.953	**0.053906**	142.5	44	91	**Match**
491	7.47E-41	**500**	27.015	**0.05403**	149.5	34	96	**Match**
492	1.55E-40	**500**	26.922	**0.053844**	162.5	33	70	**Match**
493	4.85E-41	**500**	26.984	**0.053968**	173.5	40.5	83	**Match**
494	1.08E-40	**500**	26.922	**0.053844**	177.5	40	86	**Match**
495	7.56E-41	**500**	27.031	**0.054062**	126	51	105	**Match**
496	2.80E-41	**500**	26.953	**0.053906**	159	50	104	**Match**
497	4.92E-41	**500**	27	**0.054**	155.5	48	98	**Match**
498	1.01E-40	**500**	27.266	**0.054532**	161.5	45	109	**Match**
499	3.83E-41	**500**	26.954	**0.053908**	157.5	48	101	**Match**
500	8.11E-41	**500**	27.359	**0.054718**	156	51	103	**Match**