ECE 738: Advanced Image Processing

Final Project Report

Biometrics Based Identification:
A Survey

By
Nishant Khattar
Statement of Work

The goal of this paper “Biometrics Based Identification: A Survey”, is to present a brief overview of how our body uniquely identifies us and how this fact can be used to provide us with the much needed security in this hi-tech world that we live in. Keeping this aim in mind, the paper takes a look at the different techniques for identification/verification of an individual based on what he is and not what he knows or possesses. As a road map through the content to follow, we shall take a look at the four most widely used modalities, namely: Fingerprints, Face recognition, Iris recognition and hand geometry. Apart from comparing the accuracy of these four modalities, the paper also compares, in the case of each modality, the various algorithms used to implement that modality based system. In some cases, I have also compared the output results when two or more algorithms, for a particular modality, are combined with the outcome of a more accurate system.

The paper commences with the introduction which answers the question of why security systems based on biometrics have become an integral part of our world, following which is given the important traits for something to be considered as a biometric and how there broad classification of parameters to measure the practicality of such a biometric based system.

For the first biometric based system, we take a look at the system based on fingerprints, which are also the most widely used and accepted biometric. I shall be describing a minutiae based fingerprint identification algorithm which makes use of an adaptive bounding box technique in order to alleviate the discrepancy caused due to non linear deformations which may arise during the fingerprint acquisition process and comparing it to a previously used feature extraction algorithm based on ridge flow orientation.

For the face as a biometric, the paper concentrates on addressing of the question “Does the combination of different Face Experts help in reduction of the Total error during classification?” I have considered five different face experts for the same: three based on Linear Discriminant Analysis and two on the Probability Matching Algorithm. Results are shown for the combination of two, three and four face experts from which, it can be clearly concluded that as the number of face experts combined increases, so does the accuracy.

The Iris based identification/verification system is by far the most accurate of the four. In this section, the paper takes a look at a relatively new approach to iris recognition using circularly symmetric filters. This algorithm is based on the fact that the orientation of the features does not play a significant part in the matching process. The performance of this system is compared to that
of Multi-channel Gabor filtering method and the much famous Daugman Methodology. It is seen that the accuracy of the CBF method is at 99.85% which is only 15% below the Daugman method, but this difference appears to be very small when you consider that the Daugman method makes use of a feature vector of size 2048 as compared to the 384 used by CBF.

As the last of the four biometrics considered, we take a look at a prototype hand identification system developed by Recognition systems Inc. Along with a brief description of the image acquisition apparatus, the paper also describes various feature measurements made and the manner in which a feature is selected to participate in matching process. As far as classifiers are concerned, we deal with Euclidean, Hamming distance and Gaussian Mixture Model and compare their performances. The GMM classifier seems to give the best results in both the experiments conducted: the first one with varying the number of enrollments keeping the size of the feature vector constant at 25 and in the second, we varied the feature vector size keeping the number of enrollments at a fixed five.
BIOMETRICS BASED IDENTIFICATION: A SURVEY

1. INTRODUCTION

Traditional personal identification technologies to verify the identity of an individual, made use of “something you know”, such as personal identification number (PIN) or “something you have”, such as a key or ID card etc. Such systems are no longer considered to be reliable enough to satisfy security requirements as all of these techniques suffer from a common problem, the inability to differentiate between an authorized individual and an imposter who fraudulently acquires the access privilege of authorized personal.

Biometrics is a technology that uniquely identifies a person based on his physiological or behavioral characteristics. Since it relies on “something you are”, it can inherently differentiate between authorized personal and an imposter. Theoretically, any human physiological or behavioral characteristic can be used to make a personal identification as long as it satisfies the following requirements [1]:

1) **Universality**: Every person should have the characteristic
2) **Uniqueness**: No two people should be the same in terms of the characteristic
3) **Permanence**: Characteristic should be invariant with time
4) **Collectability**: Characteristic should be able to be measured quantitatively

<table>
<thead>
<tr>
<th>BIOMETRICS</th>
<th>UNIVERSALITY</th>
<th>UNIQUENESS</th>
<th>PERMANANCE</th>
<th>COLLECTABILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face</td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>Fingerprint</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>Hand Geometry</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>Iris</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
</tr>
</tbody>
</table>

For practical purposes, there are some additional requirements:

1) **Performance**: This refers to the achievable identification accuracy, the resource requirements to achieve acceptable accuracy and the working or environmental factors that affect the identification accuracy.

2) **Acceptability**: To what extent people are willing to accept the biometric system
3) **Circumvention:** How easy it is to fool the system

<table>
<thead>
<tr>
<th>BIOMETRICS</th>
<th>PERFORMANCE</th>
<th>ACCEPTABILITY</th>
<th>CIRCUMVENTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Fingerprint</td>
<td>High</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>Hand Geometry</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Iris</td>
<td>High</td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>

Biometric accuracy is measured in two ways; the rate of false acceptance (an impostor is accepted as a match - Type 1 error) and the rate of false rejects (a legitimate match is denied - Type 2 error) [2].

Every biometric technique has a different method of assigning a "score" to the biometric match; a "threshold value" is defined which determines when a match is declared. Scores above the threshold value are designated as a "Hit" and scores below the threshold are designated as "No-Hit." A Type 2 error occurs if a true match does not generate a score above the threshold. A Type 1 error is made when an impostor generates a match score above the threshold. If the Type 1 and Type 2 error rates are plotted as a function of threshold value, they will form curves which intersect at a given threshold value. The point of intersection (where Type 1 error equals Type 2 error) is called the crossover accuracy or Equal Error rate of the system. In general, as the value of the crossover accuracy becomes higher, the inherent accuracy of the biometric increases.

Biometrics is a rapidly evolving technology that has been widely used in forensics (fingerprints), prison security etc. and has the potential to be widely adopted in a broad range of civilian applications such as physical access control, banking security, information system security, voter and driver registration etc. Currently the world market for biometric systems is estimated to be $112 million and growing.

### 2. FINGERPRINT MATCHING

Fingerprints are graphical flow like ridges present on human fingers. Their formations depend on the initial conditions of embryonic mesoderm from which they develop. Fingerprints can be broadly classified into six categories i.e. arch, tented arch, right loop, left loop, whorl and twin loop. These six classifications are given below in figure 2.1.
Fingerprint based identification systems are possible and very efficient because fingerprints have two very important characteristics required for biometric identification i.e. fingerprints are unique for an individual and the configurations and minute details of individual ridges and furrows are permanent and unchanging, i.e. fingerprints do not change over time. It is for this reason that fingerprint identification was the first legally accepted biometric technique for identification. Amongst the identification techniques used for commercial purposes, fingerprinting is the most common and accounts for $ 100 million out of the total $112 million stated before [3].

The most widely accepted automated fingerprint matching is minutiae matching algorithm. In this section, I shall describe an improved minutiae-extraction algorithm, which in addition to previous finger printing algorithms has the ability to compensate adaptively for the nonlinear deformations and inexact and inexact transformations between template and the input. Although this algorithm works well for both, identification and verification, we shall deal only with the later in this section.

2.1 Design of a fingerprint verification system

Automated fingerprint identification consists of mainly four components: acquisition, representation, feature extraction and matching. We shall take a look of each one of these components briefly and then mainly concentrate on the feature extraction and matching part in the sections to follow.

2.1.1 Acquisition

There are primary two methods of capturing a fingerprint image, inked and live scan. The acquisition of inked fingerprints is extremely cumbersome; in the context of an identity-authentication system, it is both infeasible and socially unacceptable. This however, is not the case when it comes to live scans. Images by this method are obtained by direct scanning of the fingerprint. The most popular technology employed in live scans is that based on the frustrated total Internal Reflection concept [4]. When a finger is placed on one side of a glass prism, ridges of the finger are in contact with the plate, while the valleys are not. A laser light source is used to illuminate the glass at a certain angle and a camera is placed such that it can capture the light reflected from the glass. The light incident on the plate at the glass surface in contact with the ridges in scattered randomly, and the glass surface corresponding to the valleys undergoes total internal reflection, resulting in a corresponding fingerprint image at the CCD.
2.1.2 Representation

There are various ways to represent the template and the matching algorithms used depend on these methods. The common hypothesis underlying all of the below mentioned representations is that the individuality of fingerprints is captured by the local ridge structure and their spatial distributions. Representations based on the entire gray-scale profile of a fingerprint image are prevalent among verification systems employing optical matching [5]. The performance of such systems are limited due to factors like brightness variations, image-quality variations etc. Representations that rely on the entire ridge structure have also been proposed. Such representations are largely invariant to brightness variations, but are significantly more sensitive to the quality of the image. An alternative to these two approaches would be to extract landmark features from a binary fingerprint image since the presence of landmarks is easier to verify.

Figure 2.1. Fingerprint schema classification (a) arch, (b) tented, (c) right loop, (d) left loop, (e) whorl, and (f) twin loop. Critical points called core and delta are marked in (c).
Approximately 150 local ridge structures have been identified. It would be extremely difficult to automatically, quickly and reliably extract these different representations from fingerprint images. Automatic fingerprint identification and authentication systems rely on representing two most prominent structures [6]: ridge endings and ridge bifurcations shown in figure 2.2. These two structures are background foreground duals of each other and pressure variations can lead to the conversion of one type of structure into the other. Hence, most systems do not distinguish between the two, i.e. both the structures are treated equivalently and are collectively called minutiae. The simplest minutiae-based representation consist a list of points defined by their spatial coordinates. However, additional features are attached to the basic representation in the form of ridge orientation

![Figure 2.2. Ridge endings and ridge bifurcations](image)

at the minutiae or ridge count the number of ridges visited during the linear traversal between two minutiae, orientation of the finger, locations of core and delta etc.

The representation used in this paper is minutiae based and each minutia is described by its location i.e. (x, y coordinates) and the orientation.

### 2.1.3 Feature Extraction

A feature extractor finds the ridge endings and bifurcations from the input image. If ridges can be perfectly located in the input, then minutiae extraction is just a trivial task if extracting singular points in the thinned ridge map. However, it is not always possible to obtain a perfect ridge map. The performance of currently available algorithms depends heavily on the quality of the fingerprint image. Due to factors such as occupational marks, problems with acquisition device etc. fingerprint images may not be well-defined ridge structures leading to the need of reliable extraction algorithms, which do not assume perfect ridge structures and degrade gracefully with the quality of the fingerprint. The algorithm presented in later sections is an improvement over previously used algorithm [7] in terms of reliability and speed.
2.1.4 Matching

The matching module, given an input and a template, decides whether the two images are of the same finger using a metric of similarity and a threshold. In case of the minutiae based representation, similarity matching reduces to a point pattern-matching problem. In an ideal case (no deformations & each minutia is localized), matching is trivial. But usually the correspondence between the test and the reference image is not known. Inconsistencies are also introduced due to the acquisition system. The reasons of such inconsistencies include non-uniform contact due to dryness of the skin, skin disease, sweat, dirt etc., irreproducible contact due to manual work, accidents and injuries changing the ridge structure permanently or temporarily, feature extraction artifacts due to measurement errors and noise added due to sensing.

In light of the above operational environment, an efficient fingerprint matching algorithm should include properties such as finger may be placed at different locations and orientations on the glass plate while sensing, spurious minutiae may be present in both reference and test images, genuine minutiae may be absent in the reference of test patterns, individual perturbations among the corresponding minutiae could be relatively large (with respect to ridge spacing), but the perturbations among pairs of minutiae are spatially linear or even nonlinear. A matcher may rely on one or more of these assumptions resulting in a wide range of behavior for e.g. at one end we have the Euclidean matchers, which allow only rigid transformations between the test and template and at the other end, topological matchers which may allow the most general transformations. Elastic matchers [8] accommodate a small bounded local perturbation of minutiae from their true locations but cannot handle large displacements from their true locations. For our system, an alignment based matching algorithm is proposed. The algorithm is capable of finding the correspondences between minutiae without resorting to an exhaustive search and has the ability to compensate adaptively for the non-uniform deformations and inexact transformations between the fingerprints.

2.2 Minutiae Extraction

Fingerprint authentication is based on the matching of minutiae patterns. A reliable minutiae extraction algorithm is critical to performance of an automatic identity authentication system using fingerprints. The extraction algorithm consists of mainly three components: 1)
Orientation field estimation, 2) Ridge Extraction and Minutiae extraction and post-processing. A flow chart of the algorithm is given below in figure 2.3.

2.2.1 Orientation Field Estimation

The orientation field of a fingerprint image represents the intrinsic nature of the fingerprint and plays a very important role in fingerprint-image analysis. To determine the orientation field, the first step is to divide the image into blocks of size $W \times W$. Next we compute the gradients $G_x$ and $G_y$ at each pixel in each block. The local orientation at pixel $(i, j)$ is given by

$$
\theta(i, j) = \frac{1}{2} \tan^{-1}\left( \frac{V_x(i, j)}{V_y(i, j)} \right)
$$

where $\theta(i, j)$ is the local ridge orientation at pixel $(i, j)$.

After the orientation field on an input image is estimated, a segmentation algorithm based on the local certainty level of the orientation field, is used to locate the region of interest within the input image. The certainty level is defined as

$$
CL(i, j) = \sqrt{\frac{1}{W \times W} \left( \frac{V_x(i, j)^2 + V_y(i, j)^2}{V_x(i, j)} \right)}
$$

where

$$
V_x(i, j) = \sum_{u=-\frac{W}{2}}^{\frac{W}{2}} \sum_{v=-\frac{W}{2}}^{\frac{W}{2}} (G_x(x, y) + G_y(x, y))
$$

For each pixel, if its certainty level of the orientation is below a certain threshold $T_s$, then the pixel is marked as a background pixel.
2.2.2 Ridge Extraction

An important property of the ridges in a fingerprint image is that the gray-level values on ridges attain their local maxima along a direction normal to the local ridge orientation. Pixels can be identifies to be ridges based on this property. This is in turn done by convolving the image with two masks $h_t(i, j; u, v)$ and $h_b(i, j; u, v)$ of size $L \times H$ (generally $11 \times 7$), respectively. The two masks are capable of accentuating the local maxima gray-level values along a direction normal to the local ridge direction.
Where, $\theta(i,j)$ is the local ridge orientation and $\delta$ is a large constant. If both the gray-level values at pixel $(i,j)$ of the convolved images are larger than a certain threshold, $T_{\text{ridge}}$, then the pixel is labeled as a ridge.

### 2.2.3 Minutiae Detection

Once the ridge points are extracted, 8-connecting the points creates a thinned ridge map. Without any loss generality, it is assumed that if a pixel lies on a thinned ridge, then its value is one, else zero. The minutiae detection is done as follows. Let $(x,y)$ denote a pixel on thinned ridge and $N_0, N_1, \ldots, N_7$ denote its eight neighbors. A pixel is a ridge ending if $\sum_{i=0}^{7} N_i = 1$ and a ridge bifurcation if $\sum_{i=0}^{7} N_i > 2$. However, due to the presence of spikes and breaks in the ridge map, which may lead to many spurious minutiae, we apply a smoothening algorithm before the detection is done. The smoothening algorithm uses the following heuristics:

- If the angle formed by a branch and trunk ridge is larger than $T_{\text{lower}}$ (=70) and less then $T_{\text{upper}}$ (=110) and the length of the branch is less than $T_{\text{branch}}$ (=20 pixels), then the branch is removed.
• If a break in a ridge is shorter than $T_{\text{break}}$ (=15 pixels) and no other ridges pass through it, then the break is connected.

For each detected minutiae, the following parameters are recorded: 1) x-coordinate, 2) y-coordinate, 3) orientation and 4) the associated ridge segment. The recorded ridges are represented as one-dimensional discrete signals, which are normalized by a preset length parameter that is approximately equal to the inter-ridge distance of the finger. About ten locations on the associated ridge are sampled per minutiae. These recorded ridges are used for pattern alignment. The entire representation of the finger, in a compressed form, takes on an average about 250 bytes.

2.3 Minutiae Matching

The use of a point-matching algorithm for verification enables us to design a robust, simple and fast verification algorithm. It also allows us to keep the template size small. A number of point match techniques are known. The relaxation approach [9] for example, is an iterative approach to point matching but due to this nature, time consuming. The generalized Hough Transform is another widely used approach but has a drawback in the sense that if there are a few minutiae points, it is difficult to accumulate enough evidence in the transform space for a reliable match. Then there is the Tree-pruning approach. This approach however, imposes a number or requirements on the input point sets which are not difficult to satisfy in practice, especially for our purpose.

In our system, we make use of alignment based pattern matching. The matching algorithm is as follows. Let $P = ((x_1^P, y_1^P, \theta_1^P), ..., (x_M^P, y_M^P, \theta_M^P))$ denote the M minutiae in the template and $Q = ((x_1^Q, y_1^Q, \theta_1^Q), ..., (x_N^Q, y_N^Q, \theta_N^Q))$ denote the N minutiae in the input image. We begin by estimating the translation and rotation parameters between the corresponding minutiae and their ridges from the template and input image. Next we convert the input and the template pattern into its polar coordinates with respect to the minutiae on which aligning was achieved and represent them as two symbolic strings by concatenating each minutia in an increasing order of radial angles. Consider,

$$P_p = ((r_1^p, e_1^p, \theta_1^p), ..., (r_M^p, e_M^p, \theta_M^p)) \quad \text{and} \quad Q_p = ((r_1^p, e_1^p, \theta_1^p), ..., (r_N^p, e_N^p, \theta_N^p))$$

Where $r$, $e$ and $\theta$ represent the corresponding radius, radius angle and normalized minutiae orientation, with respect to the reference minutia. Match the resulting strings $P_p$ and $Q_p$ using a modified dynamic programming algorithm described below to find the ‘edit distance’ between
the two strings and use it to find the correspondence between them. The matching score is defined by \( S = \frac{100 \times M_{PQ}^2}{MN} \), where \( M_{PQ} \) is the number of minutiae that fall into the bounding boxes of template minutiae. The bounding box specifies the possible positions of the corresponding input minutiae with respect to the template minutiae.

This algorithm decomposes into two stages: 1) alignment stage, where transformations such as translation, rotation etc, between the input and a template in the database are estimated and the two minutiae are aligned, 2) matching stage, where both the input and template minutiae are converted into a polygon and elastic string matching algorithm is used to match the resulting polygons.

![Figure 2.4. Result of Minutiae Extraction algorithm on a finger print image (512x512). (a)Input image, (b) Orientation Field Superimposed, (c) Fingerprint Region, (d) Extracted Ridges, (e)Thinned Ridge map, (f) Extracted minutiae and their orientations superimposed on the input image.](image-url)
2.3.1 Alignment of Point Patterns

The approach we use to achieve this is based on the fact that corresponding curve segments care capable of aligning two point patterns with high accuracy in the presence of noise and deformations. Hence, alignment is achieved by aligning corresponding ridges, which are provided by the ridge recordings in the minutiae extraction stage. By matching these ridges, the relative pose transformation between the input fingerprint and the template can be accurately estimated. This alignment process is shown in figure 2.5.

![Figure 2.5 Alignment of the input ridge and the template ridge](image)

Let $R^d$ and $RD$ denote the sets of ridges associated with the minutiae in the input and the template respectively. Compare each ridge $d \in R^d$, represent it as an one dimensional discrete signal and match it against each ridge $D \in R^D$ in accordance with the formula

$$S = \frac{\sum_{i=0}^{L} d_i D_i}{\sqrt{\sum_{i=0}^{L} d_i^2 D_i^2}}$$

where $L$ is the minimal length of the two ridges and $d_i$ and $D_i$ represent the distances from point $i$ on the ridges $d$ and $D$ to the x-axis, respectively. The sampling interval on a ridge is set to the average inter-ridge distance. If the matching score is $S$ is larger than a certain threshold $T_r (=0.8)$, we proceed to estimating the transformation between the two ridges, else we continue to match the next pair of ridges. For the estimation we make use of the translation vector between the two ridges which is computed as

$$\left( \frac{\Delta x}{\Delta y} \right) = \left( \begin{array}{c} x^d - x^D \\ y^d - y^D \end{array} \right),$$

where $(x^d,y^d)^T$ and $(x^D,y^D)^T$ are the x and y coordinates of the two minutiae, which are called the reference minutiae. The rotation angle between the two ridges is given
by $\Delta \theta = \frac{1}{L} \sum_{i=0}^{L} (\gamma_i - \tau_i)$, where $L$ is the minimal length of the two ridges, and $\gamma_i, \tau_i$ are the radial angles of the $i^{th}$ point on the ridge with respect to the reference minutiae associated with the two ridges. We now denote the minutiae $(x^d, y^d, \theta^d)^T$ as the reference minutiae, based on which the transformation parameters are estimated. Translate and rotate all the N input minutiae with respect to this reference minutia, using the formula:

$$
\begin{pmatrix}
    x_i^A \\
    y_i^A \\
    \theta_i^A
\end{pmatrix}
= \begin{pmatrix}
    \Delta x \\
    \Delta y \\
    \Delta \theta
\end{pmatrix}
+ \begin{pmatrix}
    \cos \Delta \theta & \sin \Delta \theta & 0 \\
    -\sin \Delta \theta & \cos \Delta \theta & 0 \\
    0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
    x_i - x^d \\
    y_i - y^d \\
    \theta_i - \theta^d
\end{pmatrix},
$$

where $(x_i, y_i, \theta_i)^T, (i=1,2,\ldots,N)$, represents the input minutiae and $(x_i^A, y_i^A, \theta_i^A)^T$ represent the corresponding aligned minutiae.

### 2.3.2 Aligned Point Pattern Matching

If two identical point matching patterns are exactly aligned with each other, then each pair of corresponding points are completely coincident, making the point pattern matching trivial. In practice however, this is seldom the case. Besides, our alignment scheme described above does not model nonlinear deformation of fingerprints, which is an inherent property of fingerprint impressions. With the existence of such a nonlinear deformation, it is impossible to recover the position of each minutia exactly and hence the need of the matching algorithm to be elastic, i.e. it should be capable of tolerating, to some extent, the deformations due to inexact extraction of minutiae positions and nonlinear deformations. Therefore, we make use of an adaptive elastic matching algorithm which can be broken up into two main steps: 1) representing minutiae patterns as a string in the polar coordinate system, 2) matching the strings with a dynamic programming algorithm to establish the correspondence.

Not only is it easier to manipulate the representations in polar space, but at the same time it is also easier to formulate rotation, which constitutes the main part of the alignment error between the input and template. The symbolic string generated by concatenating points in increasing order of radial angle in polar coordinates uniquely represents a point pattern. This reveals that point pattern matching can be achieved by string matching algorithm.

Generally, string matching can be thought of as the maximization/minimization of a certain cost function, such as the edit distance. Including an elastic term in the cost function of a string-matching algorithm can achieve a certain amount of error tolerance. Given two strings
P_p and Q_p of lengths M and N, respectively, we define the “edit distance” C(M,N) recursively as shown below.

\[
C(m, n) = \begin{cases} 
0, & \text{if } m = 0 \text{ and } n = 0 \\
\min \left\{ C(m-1, n) + \Omega, \right. \\
\left. C(m, n-1) + \Omega, \right. \\
\left. C(m-1, n-1) + w(m, n) \right\}, & 0 < m \leq M \text{ and } 0 < n \leq N
\end{cases}
\]

\[
w(m, n) = \begin{cases} 
\alpha |r_m^p - r_n^q| + \beta |\Delta c| + \gamma |\Delta \theta|, & \text{if } |r_m^p - r_n^q| < \delta, \Delta c < \epsilon, \text{ and } \Delta \theta < \rho \\
\Omega, & \text{otherwise}
\end{cases}
\]

\[
\Delta c = \begin{cases} 
\alpha, & \text{if } \left( \alpha = (c_m^p - c_n^q + 360 \mod 360) < 180 \right) \\
\alpha - 180, & \text{otherwise}
\end{cases}
\]

\[
\Delta \theta = \begin{cases} 
\alpha, & \text{if } \left( \alpha = (\theta_m^p - \theta_n^q + 360 \mod 360) < 180 \right) \\
\alpha - 180, & \text{otherwise}
\end{cases}
\]

Where, \( \alpha, \beta \) and \( \gamma \) are the weights associated with radius, radial angle and minutiae direction respectively; \( \delta, \epsilon \) and \( \rho \) specify the bounding box shown in figure 2.6. \( \Omega \) is a pre-specified penalty for a mismatch.

The purpose of the adaptive mechanism is to track the local nonlinear deformations, inexact alignment and try to alleviate them during minimization. In this matching algorithm, adaptation is achieved by adjusting the bounding box when an inexact match is found.

---

*Figure 2.6 Bounding Box and its Adjustments*
Mathematically, this can be represented as:

\[
\begin{align*}
  w'(m,n) &= \begin{cases} 
    \alpha |r_{Tn}^P - r_{Tn}^Q| + \beta \Delta \phi + \gamma \Delta \theta, & \text{if} \quad \left\{ \begin{array}{l} 
    \delta_q(m,n) < (r_{Tn}^P - r_{Tn}^Q) < \delta_h(m,n), \\
    \epsilon_q(m,n) < \Delta \phi < \epsilon_h(m,n) 
  \end{array} \right\} \\
    \Omega, & \text{otherwise}
  \end{cases}
\end{align*}
\]

\[
\begin{align*}
  \left( \Delta r_a, \Delta e_a \right) &= \begin{cases} 
    \left( \frac{r_{Tn}^P - r_{Tn}^Q}{\Delta \phi} \right), & \text{if} \quad \left\{ \begin{array}{l} 
    \delta_q(m,n) < (r_{Tn}^P - r_{Tn}^Q) < \delta_h(m,n), \\
    \epsilon_q(m,n) < \Delta \phi < \epsilon_h(m,n) 
  \end{array} \right\} \\
    0, & \text{otherwise}
  \end{cases}
\end{align*}
\]

Where \( w'(m,n) \) represents the penalty for matching a pair of minutiae \((r_n^P, e_n^P, \theta_n^P)\) and \((r_n^Q, e_n^Q, \theta_n^Q)\); \( \delta_q(m,n), \delta_h(m,n), \epsilon_q(m,n) \) and \( \epsilon_h(m,n) \) specify the adaptive bounding box in the polar coordinate system i.e. radius and radial angle and \( \eta \) is the learning rate.

The above parameters have been empirically determined to be \( \delta_h(0,0) = +8 \) pixels, \( \delta_q(0,0) = -8 \) pixels, \( \epsilon_q(0,0) = -7.5 \) pixels, \( \alpha = 1, \beta = 2, \gamma = 0.1, \eta = 0.5, \Omega = 200(\alpha + \beta + \gamma) \) and \( \Omega = 30 \). The results of applying the matching algorithm to an input and a template are shown below.

### 2.4 Experimental Results and Conclusions

Experiments were carried out on mainly two databases: 1) MSU [10] fingerprints database and 2) NIST 9 [11] fingerprint database. The MSU database contains ten images (640 x 480) per finger from 70 individuals for a total of 700 fingerprint images. When these fingerprints were captured, no restriction on the orientation and position of the finger was imposed.

The NIST 9 fingerprint database contains 1350 mated fingerprint card pairs (image size 832 x 768) that approximate a natural distribution of the National Crime and Information Center Fingerprint classes. It is divided into multiple volumes, each volume having 3 CDs. A CD contains 900 images scanned using a rolled method (card 1) and 900 live scans (card 2).
2.4.1 Feature Extraction Performance

The feature extraction performance of the new hierarchical adaptive algorithm was done and compared to the previously used algorithm. We make use of the same matching algorithm for both and therefore, the results should give us a fair account of the performance of the feature extraction part. Verification accuracy was estimated using a Hough Transform based matcher [8]. The results are shown in figure 2.7. The ROC’s resulting from these two experiments clearly shows a significant improvement in accuracy.

![Figure 2.7 ROC’s showing the improvement in performance of verification](image)

2.4.2 Matching Scores

We performed two tests, in the first one; each fingerprint in the MSU database was matched with all other fingerprints in the database. A matching was labeled correct if the matched fingerprint was from the same finger and incorrect otherwise. In test 2, each of the 900 fingerprints of card type 1 in INST 9 (CD1) was matched with all 900 fingerprints of card type 2. Table 2.1 shows the results.
Table 2.1 $d'$ and mean and Std. Deviation of the correct and incorrect Matching scores

Where $d' = \frac{\left\| M_{\text{impostor}} - M_{\text{genuine}} \right\|}{\sqrt{(SD_{\text{impostor}}^2 - SD_{\text{genuine}}^2)/2}}$ is a statistical metric and gives indication of the separation between the genuine and imposter distributions.

The large variance of correct matching scores is mainly due to different numbers of detected minutiae, quality of fingerprint and fingerprint distortion. For example, if two fingerprints are taken of the same finger at different orientations and locations, the region of interest common to both the prints may be very small, which makes it impossible to give a highly confident decision based on the limited number of minutiae detected.

2.4.3 Authentication Test

In test 1 involving the MSU database, we have made use of multiple templates (=3) so as to counter the small region of interest encountered between the test image and template due to the unrestricted acquisition test. An identity is established if at least one of the three matching scores is above a certain threshold, else rejected. In test 2, the 900 fingerprints of card type 2 were taken to be the input and the images from card type 1 as templates. The results are shown in table 2.2 below for different values of threshold.

<table>
<thead>
<tr>
<th>THRESHOLD VALUE</th>
<th>FALSE ACCEPTANCE RATE (MSU)</th>
<th>FALSE REJECT RATE (MSU)</th>
<th>FALSE ACCEPTANCE RATE (NIST-9)</th>
<th>FALSE REJECT RATE (NIST-9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>0.07%</td>
<td>7.1%</td>
<td>0.073%</td>
<td>12.4%</td>
</tr>
<tr>
<td>8</td>
<td>0.02%</td>
<td>9.4%</td>
<td>0.023%</td>
<td>14.6%</td>
</tr>
<tr>
<td>9</td>
<td>0.01%</td>
<td>12.5%</td>
<td>0.012%</td>
<td>16.9%</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>14.3%</td>
<td>0.003%</td>
<td>19.5%</td>
</tr>
</tbody>
</table>

Table 2.2 False acceptance and Reject rates on Test sets with different Threshold Values
The ROC’s of the two tests are given in Figure 2.8 below. The FAR and AAR are plotted in semi log space. If the genuine and imposter distributions are disjoint, i.e., each genuine person is accepted and each imposter rejected, the ROC is a horizontal line segment hovering at the authentic acceptance rate of 100%. On the other hand, if the two distributions are exactly the same, then the ROC would be a line at 45° with one end point at the origin (shown dotted). In practice, ROC is a curve between these two extremes (solid line). The closer the ROC is to the upper boundary, the better.

Figure 2.8. ROC. (a) MSU database, (b) NIST-9 (CD no. 1)
3. BIOMETRIC IDENTIFICATION THROUGH FACE RECOGNITION

In literature, many methods have been introduced to tackle the challenging problem of face recognition/verification. Each one of these methods makes use of different feature extraction and matching algorithms to generate a matching score which is used for authentication. The system performance is primarily based on the error rate and the perennial aim of researchers has always been the attempt to reduce it. Now given the above, one wonders if the error rate can be reduced by the combination of two methods so as to reduce the probability of misclassification of a given image. Since we already have a very good idea of the various face recognition techniques, in this section I wish to take a look at the improvement in verification rates due to the combination of face verification methods.

3.1 Face Experts

In this section we shall describe the five recognition methods (face experts) that we shall use for combination. Among them, three are based on Linear Discriminant Analysis and two on Probabilistic Matching.

3.1.1 LDA-based Experts

In linear discriminant analysis [12], the features are extracted by linearly projecting the face vectors into a subspace where the intraclass facial appearance is minimized while the interclass variation is maximized. Our three face experts that apply this approach for feature extraction are as follows: 1) GDM, 2) NOC and 3) LNC. The first two methods [13] make use of the same LDA Algorithm for feature extraction and differ only in the feature matching phase. The former uses gradient direction metric as the matching scheme, while the latter makes use of normalized correlation for matching. The third method, i.e. LNC is based on LDA and makes use of normalized correlation with different preprocessing parameters.

3.1.2 Probabilistic Matching Experts

In the probabilistic matching approach [14] the algorithm classifies the difference $\Delta$ between the test and template as intrapersonal variations $\Omega_i$ or interpersonal variations $\Omega_e$. The similarity matching is done based on the probability of image variation given as $P(\Omega_i | \Delta)$. Using Bayes formula, this probability can be expressed as
These high dimensional probability densities are obtained from training data using the Eigenface estimation technique. Now, probabilistic matching techniques used for this experiment make use of two different photometric normalization techniques. In the first one, known as PM1, the image pixels are transformed to have zero mean and a unit variance. In the second, PM2, the normalization is done using histogram equalization.

3.2 Combination and Experiments

The performance of a system which employs the combination of more than algorithms, the combination rules used to join the algorithms play a vital role. However, since my primary goal is to assess the variability of combining face verification experts, we can make use of two simple combination rules, namely, the sum rule [15] and the fisher discriminant function. The sum rule takes a weighted average of the individual score values to create a new matching score used for the final decision. Despite its simplicity, the sum rule has been found to outperform a number of other classifier combination schemes. The fisher discriminant function, on the other hand, defines a new score \( s \) using the vector \( x \) of individual expert score values. The score is defined as:

\[
s = x^T \left( \frac{\Sigma_c + \Sigma_I}{2} \right)^{-1} (\mu_c - \mu_I)
\]

Where \( \Sigma_c \) and \( \Sigma_I \) are the class dependent covariance matrices, \( \mu_c \) and \( \mu_I \) are the means for clients and imposters respectively.

3.2.1 Database and Experimental Protocol

The experiments were performed on frontal face images taken from the extended M2VTS database. It contains face images of 259 people. The subjects were recorded in four different sessions uniformly distributed over a period of five months. The Lausanne protocol [16], which is a protocol designed specifically for multimodal fusion experiments, is used as the experimental protocol for our experiments. It divides the database into a set of 200 clients (used for testing false rejection rate) and 95 imposters (used for testing false acceptance rate). The protocol specifies a partitioning of the database into disjoint sets for training, evaluation and testing. The verification performances are measured using the FRR and FAR. The operating point where these two error are equal is referred to as the
Equal error rate. The threshold is found experimentally by computing the ROC on the evaluation set and selecting the EER operating point.

3.3 Experiments and Observations

The experiments were performed in two stages. The first stage consisted of giving the performance comparison of the individual face experts and the second, applying the combination rules and comparing the verification performances of the various combinations.

3.3.1 Individual Expert Performance

Table 3.1 shows the verification results for individual experts. The first column shows the EER obtained on the evaluation set. The three last columns show the FAR, FRR and Total error rate (TER) obtained on the independent test set at the same operating conditions at which the EER was obtained.

It is observed from the above table that the GDM algorithm offers the best verification performance (TER=5.69%). The importance of the photometric normalization for face verification in case of Probalistic methods can be clearly seen by observing the difference in performance between PM1 (TER=9.32%) and PM2 (TER=7.71%).

<table>
<thead>
<tr>
<th>EXPERT USED</th>
<th>EVALUATION EER (%)</th>
<th>TEST FAR (%)</th>
<th>TEST FRR (%)</th>
<th>TEST TER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDM</td>
<td>2.17</td>
<td>2.69</td>
<td>3.00</td>
<td>5.69</td>
</tr>
<tr>
<td>NOC</td>
<td>3.17</td>
<td>3.54</td>
<td>2.50</td>
<td>6.04</td>
</tr>
<tr>
<td>LNC</td>
<td>2.93</td>
<td>3.89</td>
<td>2.75</td>
<td>6.64</td>
</tr>
<tr>
<td>PM1</td>
<td>4.66</td>
<td>5.57</td>
<td>3.75</td>
<td>9.32</td>
</tr>
<tr>
<td>PM2</td>
<td>3.44</td>
<td>4.71</td>
<td>3.00</td>
<td>7.71</td>
</tr>
</tbody>
</table>

Table 3.1. Individual verification results on the XM2VTS database

3.3.2 Combination results

In case of combination using the sum rule, the weights used to form the new matching scores are determined by minimizing the EER on the evaluation set. As the EER varies slowly with respect to the combination weights, an exhaustive search for the minimum is computationally
possible. The weights minimizing the EER and the corresponding threshold are then used to compute verification results on the independent data set.

For the Fisher Discriminant function, the class conditional means and covariance matrices of expert scores are estimated over the evaluation set and the new matching scores are then computed and tested on the test set.

3.3.2.1 Combination of Two Experts

Five of the ten possible combinations of two experts lead to a significant improvement in verification performance with respect to the best individual expert. The result of the experiment showing the six combinations are given in table 3.2.

As a general observation, we can see that the sum rule out performs the fisher discriminant function for each of the five combinations listed above. The best combination of two experts, for either combination rule, is NOC and PM2. The TER has reduced from 6.04% (performance of expert NOC) to 3.67% (performance of the NOC + PM2 using the sum rule).

<table>
<thead>
<tr>
<th>EXPERT COMBINATION</th>
<th>TOTAL ERROR (%)</th>
<th>TOTAL ERROR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SUM RULE</td>
<td>FISHER DISCR. FUNCT.</td>
</tr>
<tr>
<td>GDM and PM2</td>
<td>3.74</td>
<td>4.41</td>
</tr>
<tr>
<td>NOC and PM2</td>
<td>3.67</td>
<td>4.11</td>
</tr>
<tr>
<td>GDM and PM1</td>
<td>4.03</td>
<td>4.83</td>
</tr>
<tr>
<td>NOC and LNC</td>
<td>4.23</td>
<td>4.60</td>
</tr>
<tr>
<td>GDM and LNC</td>
<td>4.33</td>
<td>4.88</td>
</tr>
</tbody>
</table>

*Table 3.2. Total Error when combining two different experts using the sum rule and Fisher Discriminant function on the test set of the XM2VTS database.*

3.3.2.2 Combination of Three Experts

For the sum combination rule in case of three expert’s combination, in some cases, the weight equals to zero. One such case would be when GDM and PM2 are in the combination, adding any of the remaining experts does not improve the EER on the evaluation set. While using fisher discriminant function, in some cases the performance is found to degrade for
e.g. when NOC is added to the GDM-PM1 combination, the TER increases from 4.83% to 5.06%. The results of the five best combinations are given below in table 3.3.

<table>
<thead>
<tr>
<th>EXPERT COMBINATION</th>
<th>TOTAL ERROR (%)</th>
<th>TOTAL ERROR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SUM RULE</td>
<td>FISHER DISCR. FUNCT.</td>
</tr>
<tr>
<td>NOC, LNC and PM2</td>
<td>3.26</td>
<td>4.09</td>
</tr>
<tr>
<td>NOC, PM1 and PM2</td>
<td>3.26</td>
<td>4.04</td>
</tr>
<tr>
<td>GDM, LNC and PM1</td>
<td>3.44</td>
<td>4.51</td>
</tr>
<tr>
<td>NOC, LNC and PM1</td>
<td>3.61</td>
<td>4.53</td>
</tr>
<tr>
<td>GDM, PM1 and PM2</td>
<td>3.74</td>
<td>4.15</td>
</tr>
</tbody>
</table>

Table 3.3. Total Error when combining three different experts using the sum rule and Fisher Discriminant function on the test set of the XM2VTS database.

### 3.3.2.3 Combination of Four Experts

It can be seen from table 3.4 that the there is no improvement in the sum rule method when a fourth expert is added.

<table>
<thead>
<tr>
<th>EXPERT COMBINATION</th>
<th>TOTAL ERROR (%)</th>
<th>TOTAL ERROR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SUM RULE</td>
<td>FISHER DISCR. FUNCT.</td>
</tr>
<tr>
<td>NOC, LNC, PM2, PM1</td>
<td>3.26</td>
<td>4.41</td>
</tr>
<tr>
<td>NOC, PM1 and PM2</td>
<td>3.26</td>
<td>4.11</td>
</tr>
<tr>
<td>GDM, LNC, PM2, PM1</td>
<td>3.74</td>
<td>4.83</td>
</tr>
<tr>
<td>GDM, NOC, PM2, PM1</td>
<td>3.74</td>
<td>4.83</td>
</tr>
<tr>
<td>GDM NOC, LNC, PM2</td>
<td>3.74</td>
<td>4.60</td>
</tr>
<tr>
<td>GDM NOC, LNC, PM1</td>
<td>3.44</td>
<td>4.88</td>
</tr>
</tbody>
</table>

Table 3.4. Total Error when combining four different experts using the sum rule and Fisher Discriminant function on the test set of the XM2VTS database.

Moreover, from the above table, it is evident that adding a third and a fourth expert doesn’t improve the EER on the evaluation set as soon as the experts GDM and PM2 are in combination.
3.4 Conclusions

From the above results, it is evident that the sun rule constantly provides lower error rates than the combination using fisher discriminant function. It also appears that the number of common errors made by the different experts is an important parameter to determine if the combination can improve the performance. For e.g. the two experts that give the best result when combined, i.e. GDM and PM2, turn out to be the ones with least number of common errors. It can be seen that preprocessing techniques also play an important role as shown by table 4.1 and 4.2 in case of PM1 and PM2. It is also evident that combining of two experts that differ only in matching algorithm, i.e. NOC and GDM, does not improve the result.

4. BIOMETRIC IDENTIFICATION THROUGH IRIS RECOGNITION

The human iris has an extraordinary structure and in itself, provides abundant texture information, i.e. the spatial patterns that apparent in the iris are unique to each individual. In particular, biomedical literature [17] goes as far as saying that the iris is as distinct as fingerprints. Steps involved in iris recognition consist of 3 main stages: 1) Iris image processing, 2) Feature Extraction, 3) Classifier design. In this section we will take a look at a relatively new (2002) iris matching algorithm using circularly symmetric filters and compare the results with already existing methods.

4.1 Image processing

An iris image as shown in figure 4.1(a), apart from the region of interest i.e. the iris, also contains the eyelid, pupil etc. The camera to eye distance also plays an important role as it may lead to a variation in the iris size along with the issue of non-uniform illumination. Due to the above mentioned reasons, before the feature extraction stage commences, the iris image has to be localized, normalized and the iris image enhanced in order to nullify the effects of the above possible situations.

4.1.1 Iris Localization

The outer and inner boundaries of a typical iris can be approximated by a pair of non co-centric circles. The iris localization can be performed in two stages: 1) approximate region of iris in an image can be found by projecting iris image in horizontal and vertical direction, 2) the exact parameters of these two circles are obtained by using edge detection and Hough
transform in a certain region determined in the first step. The output of the localization step is shown in figure 4.1(b)

4.1.2 Iris Normalization

The sizes of the iris captured differ from person to person. Even the size of the iris taken from the same eye may differ depending on the amount of illumination. Such elastic deformations in the iris texture affect the results of iris matching. Hence, in order to reduce any error introduced due to pupil dilation, it is necessary to perform iris normalization. This is done by unwrapping the iris ring in the anti clockwise direction into a rectangular block of a fixed size (64 x 512) using piecewise linear mapping. The result of the normalization can be seen in figure 4.1(c)

Figure 4.1. Image pre-processing. (a) Original image, (b) Detected iris region, (c) Unwrapped image, (d) Iris image after enhancement and denoising where the region of interest (ROI) lies above the red line.

4.1.3 Iris Image Enhancement

The iris image, although normalized, still suffers from low contrast and may also have non-uniform illumination caused due to the positioning of the lighting source. In order to obtain a better distributed texture image, we enhance the image by performing local histogram equalization. Noise reduction is performed using low pass Gaussian filter in order to remove the high frequency noise. An enhanced normalized iris image is shown in figure 4.1(d), finer texture characteristics as compared to the normalized image shown in part (c) are clearly evident.
4.2 Feature Extraction

From the viewpoint of texture analysis, the local spatial patterns in an iris mainly involve frequency and orientation information. However, from experimental results it is seen that the orientation is not a crucial factor when analyzing characteristics in a small region, such as a 10 x 10 region. In other words, in a small region, frequency information accounts for the major differences of the irises from different people. Hence, the feature extraction problem can be reduced to applying a frequency band filters which are circularly symmetric in nature for the extraction.

4.2.1 Circular Symmetric Filter

In recent years, the use of Gabor filters has been extensive in computer vision for texture analysis. The use of Multi scale Gabor filters [18] to extract the information of an image at a certain scale and at a particular orientation is also well documented. In this section we introduce circular symmetric filters (CSF) which are very much like the Gabor filters but with one exception: Gabor filters are modulated using an oriented sinusoidal function, whereas CSF with a circular symmetric sinusoidal function. The CSF filter is defined as follows:

\[
G(x, y, f) = \frac{1}{2\pi \delta_x \delta_y} \exp \left[-\frac{1}{2} \left(\frac{x^2}{\delta_x^2} + \frac{y^2}{\delta_y^2}\right)\right] M(x, y, f)
\]

\[
M(x, y, f) = \cos\left[2\pi f \left(\sqrt{x^2 + y^2}\right)\right]
\]

Where \(M(x, y, f)\) is the modulating function, \(f\) is the frequency of the sinusoidal function, \(\delta_x\) and \(\delta_y\) are the space constraints of the Gaussian envelope along the x and y directions respectively. The CDF can capture the frequency information within a specific frequency band but cannot give any information about the orientation as it is circular symmetric.

In experiments, it is found that the top-most 75% section (48 x 512) of an unwrapped iris gives most of the texture information required for iris recognition. Hence, we extract features only in this section called the region of interest shown as region above the red line in figure 4.1(d). For a preprocessed iris image, the texture of the iris becomes coarser from top to down and therefore, we use filters at different frequencies for different regions in the image. So we divided the ROI into three logical regions as filter each local region with a CSF at different frequencies.
4.2.2 Feature Vector

Feature extraction is done in each of the 8 x 8 blocks in the filtered image resulting in a total of 384 feature values. In this particular algorithm, the feature value is calculated as the average absolute deviation of each filtered block defined as:

$$V = \frac{1}{N} \left( \sum_{N} |f(x, y) - m| \right)$$

Where N is the number of pixels in the image block, m is the mean and f(x, y) is the value at point (x, y). These features are arranged to form a 1 D feature vector of length 384 for each input image.

4.3 Classifier Design

In iris matching, we make use of an efficient classification method named the Nearest Feature Line Method (NFL). The original NFL algorithm [19] consisted of a feature line that extended to infinity in the feature space which would lead to false classification. In this algorithm, the NFL was modified by limiting the extent of the feature line. The improved NFL method defines the distance between an unknown point \( p_x \) and a feature line \( \overline{p_j} \) as follows:

$$D(p_x, \overline{p_j}) = \begin{cases} 
\|p_x - \hat{p}_j\| & \beta \leq T_1 \\
\|p_x - \hat{p}_j\| & T_1 \leq \beta \leq T_2 \\
\|p_x - p_j\| & \beta \geq T_2 
\end{cases}$$

Where, \( \hat{p}_x \) is the projection of \( p_x \) on the feature line \( \overline{p_j} \), \( \beta \) is called the position parameter, \( T_1 \) and \( T_2 \) are two thresholds. The original NFL corresponds to the case when \( T_1 = +\infty \) and \( T_2 = -\infty \). In our experiment the threshold values \( T_1 \) and \( T_2 \), were set to -0.5 and 1.5 respectively.

It is desirable to obtain a representation for the iris which is scale, translation and rotation invariant. Scale and translation invariance is achieved in the normalizing step of the preprocessing stage. Approximate rotation invariance is achieved by unwrapping the iris ring at different initial angles. In this algorithm seven initial angles are considered and a template made for each of the seven. When matching the input feature vector with class templates, the minimum of the seven scores is considered to be the final distance.

4.4 Experiments and Conclusions

The algorithm was tested in two modes: 1) Verification and 2) identification. For each iris class, we use three images for training and the rest for testing. In this section we compare the
performance of this algorithm as compared to two already existing iris matching techniques: 1) the famous Daugman method [20], which makes use of phase coding using Gabor filters and 2) Multi channel Gabor Filter [21] method to extract local and global details in an iris.

4.4.1 Database Creation

A database was created containing 1088 iris images taken form 109 volunteers. The new database is created by the combination of two databases. The first database created, consists of 500 images taken from 25 people. Each individual provides 20 images, 10 for each eye. Five images are taken from each individual for each eye is taken in the first week and then the process repeated four weeks later, yielding a total of ten images per eye and 20 per person. The second database consists of 588 images taken from 84 individuals, each individual providing seven images of the left eye. These images are also captured in two stages which are similar to the acquisition process used for the creation of the first database, giving a total of 134 iris classes (2 x 25 + 84)

![Figure 4.2. Iris samples from the database](image)

4.4.2 Experimental results

The verification results as plotted in figure 4.3 as false match rate (FMR) versus false non-match rate (FNMR). The second curve from the horizontal corresponds to the CSF method. Points on this curve denote all possible system operating states. Three typical operating states of this algorithm are given in table 4.1.

<table>
<thead>
<tr>
<th>FALSE MATCH RATE (%)</th>
<th>FALSE NON-MATCH RATE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>3.56</td>
</tr>
<tr>
<td>0.01</td>
<td>2.17</td>
</tr>
<tr>
<td>0.1</td>
<td>0.83</td>
</tr>
</tbody>
</table>

*Table 4.1 Verification Results*
The identification results are given below in table 4.2. It is evident from the data that the method using CSF filters gives a much better performance as compared to the method employing Multi-channel Gabor filters. It can be also seen that Daugman’s Method has only a slightly better performance than CSF in both identification and verification applications. This can be attributed to the high dimensionality used by Daugman. The feature vector used by Daugman consists of 2048 feature values as compared to the 384 in case of CSF.

<table>
<thead>
<tr>
<th>METHOD</th>
<th>CORRECT CLASSIFICATION RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-channel Gabor Filter</td>
<td>93.20 %</td>
</tr>
<tr>
<td>Daugman’s Method</td>
<td>100%</td>
</tr>
<tr>
<td>Circular Symmetric Filter</td>
<td>99.85%</td>
</tr>
</tbody>
</table>

*Table 4.2 Identification Results*
5. BIOMETRIC IDENTIFICATION THROUGH HAND GEOMETRY MEASUREMENTS

Identification based on hand geometry, as compared to other techniques, achieves medium security but has some distinct advantages:

- **Medium Cost**: It needs only a platform and low/medium resolution CCD Camera
- **Fast Results**: This is evident from the fact that this procedure uses low-computational cost algorithms
- **Small Template size**: The average template size ranges from 9 to 25 bytes, hence lesser storage area required.

5.1 Over View of the System

As in the case of any other biometric based system, identification by means of hand geometry also involves two phases: an enrollment phase and a comparison one. In the enrollment phase, several photographs are taken of the user, preprocessed and entered into the feature extraction block, where sets of measurements are performed.

The comparison block can be configured in two ways: as a classifier, where the features extracted are compared to all the users’ templates to determine which is the user photographed, or as a verifier, where the users’ identity is stated and the sample is compared with the pattern of the claimed user present in the database.

5.2 Feature Extraction

The aim of this block is to capture a sample of the user’s biological data, process it, and extract a set of features that represent univocally that user among all who use that system. In order to achieve this, four tasks have to be performed:

- Image capture
- Preprocessing
- Measurements
- Feature Selection
5.2.1 Image Capture

The sample signal is obtained using a CCD color camera, placed above a platform designed to guide the hand to a fixed location. This is done by means of six tops placed in pre-determined positions to guide the hand as shown in figure 5.1 [22]. The tops are equipped with pressure sensors such that when all of them are activated, the camera is triggered. The image captured along with the view of the palm, also contains the lateral view, which is achieved by placing a mirror on the right hand side of the platform so as to enable us to measure heights. The image captured has dimensions 640 x 480 pixels.

Figure 5.1 Different views of the prototype designed: (a) Platform and Camera, (b) placement of the user’s hand, (c) Photograph taken

5.2.2 Preprocessing

The first step in preprocessing is the conversion of the color image into a black and white one where the background is eliminated. This is done by means of an arithmetic operation performed with the different channels

\[ I_{BW} = \langle\langle I_R + I_G \rangle - I_B \rangle \]

Where \(<,>\) operation is the stretching function, \(I_{BW}\) is the Black and White Image, and \(I_R, I_G\) and \(I_B\) are the red green and blue channels. After this, a threshold is used to do away with the spurious pixels in the background. Now, edge detection by means of the Sobel operator is done in order to extract the contour of the hand.

5.2.3 Measurements

Now the processed image is a contour of the palm and the side view of the hand in a fixed position determined by the tops. This simplifies the measurement algorithms and allows us to
make measurements in terms of pixels and angles. The main measurements can be divided into four categories and are represented in Figure 5.2.

1. **Widths:** The width of each of the four fingers is measured at different heights (mainly four), avoiding the pressure points at the top of the fingers. The width of the palm is also measured \((w_0)\) and the distances between the three inter finger points \(P_1, P_2\) and \(P_3\).

2. **Heights:** The heights of the little finger, the middle finger and the palm respectively, are measured using the lateral view.

3. **Deviations:** This is defined as the distance between the middle point of the middle width measurement of the finger and the mid point of the straight line between the inter finger point and the last height where the finger width is measured.

4. **Angles:** The angles between the inter finger points in the horizontal plain.

### 5.2.4 Feature Selection

As seen above, 31 feature vectors have been extracted (21 widths, 3 heights, 4 deviations and 3 angles). Once the features from a relative large number of users with several photographs from each user are obtained, a statistical analysis is performed to determine the discriminability of the features is obtained.

This is analyzed by a ratio \(F\) between the interclass and intraclass \([23]\) variability. The higher the ratio, the more discriminant the feature is. Intraclass variability.
\[ F_j = \frac{\text{Interclass variability}}{\text{Intraclass variability}} = \frac{1}{N} \sum_{i=1}^{N} \left( V \frac{1}{N} \sum_{j=1}^{N} f_{ij} \right) \]

Where \( F_j \) is the ratio for the \( j \)th feature, \( V \) is the Std. Deviation function; \( N \) is the number of classes (users), \( f_{ij} \) is the \( j \)th feature of the \( i \)th class and \( \bar{f}_{ij} \) is the mean of the \( j \)th’s features of the \( i \)th class. After this study, the number of discriminant features was found to be 25.

5.3 Classification and Verification

Now the feature vectors obtained should enter a comparison process to determine the user whose hand photograph was taken. This comparison is made against user templates, which will be calculated depending on the comparison algorithm used which, are as follows.

5.3.1 Euclidean Distance

The Euclidean distance [24], considered the most common of all performs the measurements using the equation:

\[ d = \sqrt{\sum_{i=1}^{L} (x_i - t_i)^2} \]

Where, \( L \) is the dimensions of the feature vector, \( x_i \) the \( i \)th component of the sample feature vector and \( t_i \) the \( i \)th component of the template feature vector. This implies that the template vector dimensions are the same as that of the sample vectors. In order to create such a template, a set of photographs is taken, and the mean of the resulting feature vectors is taken to be the user template.

5.3.2 Hamming Distance

This distance, unlike in the previous case, measures the number of components of the feature vectors that differ [25]. The template generation is based on the assumption that the feature components follow a Gaussian distribution (given that it is typical for all the components of the feature vector to differ between samples of the same user) allowing us not only to gain the mean from the initial few values but also a factor of the Std. Deviation of the samples. Therefore, the template dimensions grow from 25 to 50. In the comparison process, the number of components falling outside the area defined by the template parameters is counted.
\[ d(x_i, t_i^m) = \# \{ i \in \{1, \ldots, L \} | x_i - t_i^m > t_{i}^{v} \} \]

Where \( d \) is the hamming distance, \( L \) is the dimension of the feature vectors, \( x_i \) is the \( i \)th component of the sample vector, \( t_i^m \) is the mean for the \( i \)th component, and \( t_i^{v} \) the factor of the Std. Deviation for the \( i \)th component.

### 5.3.3 Gaussian Mixture Models (GMMs)

This is a pattern recognition technique that uses an approach between statistical methods and neural networks. It is based on modeling with a determined number of Gaussian distributions, giving the probability of the sample belonging to that class [26].

\[
p(x | \theta) = \sum_{i=1}^{M} \frac{c_i}{(2\pi)^{L/2} | R_i |^{1/2}} \exp \left\{ -\frac{1}{2} (\bar{x} - \mu_i)^T R_i^{-1} (\bar{x} - \mu_i) \right\}
\]

Where \( c_i \) is the weight of each of the Gaussian models, \( \mu_i \) is the mean vector, \( R_i \) is the covariance vector, \( M \) the number of models and \( L \) the dimension of the feature vectors. The GMM should be trained, obtaining a GMM for each used, so the template of that user would be the final value of \( c_i, \mu_i \) and \( M \). The disadvantage of this method is that it greatly increases the template size.

### 5.4 Experimental Results and Observations

#### 5.4.1 Classification

The results consist of two main analyses. The first one performs the classification with 25 selected features for different size of the enrollment set (3, 4 and 5). The results are shown in Table 5.1.

<table>
<thead>
<tr>
<th>NO. OF ENROLLMENT VECTORS</th>
<th>EUCLIDEAN</th>
<th>HAMMING</th>
<th>GMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>86%</td>
<td>75%</td>
<td>88%</td>
</tr>
<tr>
<td>4</td>
<td>85%</td>
<td>82%</td>
<td>93%</td>
</tr>
<tr>
<td>5</td>
<td>86%</td>
<td>87%</td>
<td>96%</td>
</tr>
</tbody>
</table>

*Table 5.1 Classification success compared to the enrollment set size*
It can be seen that although the Euclidean distance algorithm does not suffer apparent variation, great improvement is seen in the other two algorithms i.e. Hamming distance and GMMs. GMM shows the best results, achieving a 96% success, while the metric methods are much below.

The second analyses consist of varying the number of feature vectors compared while keeping the size of the enrollment set constant i.e. five. The results are displayed below in Table 5.2.

<table>
<thead>
<tr>
<th>NO. OF FEATURE VECTORS</th>
<th>EUCLIDEAN</th>
<th>HAMMING</th>
<th>GMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>86%</td>
<td>87%</td>
<td>96%</td>
</tr>
<tr>
<td>21</td>
<td>84%</td>
<td>86%</td>
<td>97%</td>
</tr>
<tr>
<td>15</td>
<td>86%</td>
<td>88%</td>
<td>96%</td>
</tr>
<tr>
<td>9</td>
<td>77%</td>
<td>75%</td>
<td>91%</td>
</tr>
</tbody>
</table>

*Table 5.2 Classification success compared to the feature vector dimension*

The numbers of feature vectors vary as follows: 21 (the feature vector without the deviations), 15(with the deviations but without half of the finger widths), and 9 (without deviations, half the finger widths and inter finger points). It can be seen that the number of 15 feature vectors show the best result for Euclidean and Hamming distances while GMM achieve the best result with 21 features.

**5.4.2 Verification**

The same two analyses as mentioned in the classification section were performed. The results are given below in Figure 5.3 and are given in terms of False Acceptance Rate (FAR) and False Rejection Rate (FRR).

As far as the enrollment size is concerned, the results were the same as that in the classification stage i.e. five vectors obtained the best result minute variation in case of the Euclidean system and large improvement in case of Hamming and GMM. In case of the feature vector size, three main results are obtained. The first one being the GMM still shows the best result. The second, the equal error rate remains the similar for each technique, for different vector size. And lastly, the variation in the FAR and
FRR is very acute in case of 9 feature vectors, smoother for 21 and 25 feature vectors respectively. It is seen that results for 15 feature vectors is similar to that obtained for 21, being able to choose between either one, depending on memory availability.

Figure 5.3 the upper ones refer to changes in enrollment set size. The bottom ones refer to the changes in feature vector dimension

6. SUMMARY AND CONCLUSIONS

Looking back, we have taken a look at each of the four modalities, discussed the most popular and wisely used algorithms for each one and compared them with other algorithms for each modality. In the case of fingerprinting, we found that a system based on minutiae detection implementing an adaptive bounding box technique for matching, out performs a previously popular system which was based on ridge flow orientation. In the case of face recognition we compared five different algorithms: GDM, NOC, LNC, PM1 and PM2. We also took a look at effect of a combining these five algorithms on the accuracy of a system and observed that as we increase the number of
algorithms combined, the accuracy too increases. In case of iris, we saw that the circularly
symmetric filter based system, outperforms the Gabor filter implementation and makes up for its
lagging behind to the Daugman method, by being much more feasible and easier to implement than
the later. As far as hand geometry goes, we saw that the Gaussian Mixture Classifier out performs
the Euclidean and hamming distance classifiers in both the experiments conducted, i.e. the first
varying the size of the enrollment vector and the second one varying the number of feature vectors
used in matching.

The comparison [27] of the four biometrics in terms of ease of use, error incidence, user acceptance,
security level provided and long term stability are given in table 6.1 below.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Ease of use</th>
<th>Error incidence</th>
<th>Accuracy</th>
<th>User acceptance</th>
<th>Required security level</th>
<th>Long-term stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fingerprints</td>
<td>High</td>
<td>Dryness, dirt</td>
<td>High</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Hand Geometry</td>
<td>High</td>
<td>Hand injury, age</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Iris</td>
<td>Medium</td>
<td>Poor Lighting</td>
<td>Very High</td>
<td>Medium</td>
<td>Very High</td>
<td>High</td>
</tr>
<tr>
<td>Face</td>
<td>Medium</td>
<td>Lighting, age, glasses, hair</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
</tbody>
</table>

From the above table we can see that face as biometric has the highest error of incidence due to its
dependence on a large number of factors (lighting, age, hair, glasses). The iris was found to provide
the highest accuracy although rated medium when it comes to ease of use. Comparing them on the
basis of the level of security each provides, the iris gives the highest level followed by fingerprints.
The long term stability of fingerprints and iris scanning are high and that of hand geometry and face
are medium owing to their susceptibility to change with time/age.

The most important factor, for a security system would be its accuracy. Keeping that in mind, a
comparison based purely on accuracy [27] has been shown in Table 6.2 as the ratio of the number of
errors per correct matches.
Reiterating what was seen in the previous table, we see that Iris based systems provide the highest level of accuracy. An idea of how accurate iris scanning actually is, can be visualized by the fact that the closest to it are systems based on fingerprint matching, which give an error for every 500 correct matches as compared to an iris system which gives an error for every 131,000 matches. No data on face recognition system accuracy is obtained owing to the large variation in face recognition algorithms and their high dependence on the type of database they are tested on.

For future work on biometrics, identification based on DNA fingerprinting is becoming more and more popular. But such a approach although, finding extensive use in forensics has still a limited scope as far as day to day life is concerned owing to the amount of time required to extract the DNA pattern from say a strand of hair and comparing it with other DNA samples. However, this technique does provide an accuracy of close to a 100 percent and if implemented, would be as close as we can get for an identification/verification system being foolproof.
REFERENCES:


[2] [online] www.biometrics.org


