

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Efficient Fingerprint Recognition Through Improvement of Feature Level Clustering, Indexing and Matching Using Discrete Cosine Transform

Indra Devi

*Indra Ganesan College of Engineering, Tiruchirappalli,  
India*

## 1. Introduction

Fingerprint recognition refers to the automated method of verifying a match between two human fingerprints. Fingerprints are one of many forms of biometrics used to identify an individual and verify the identity. Because of their uniqueness and consistency over time, fingerprints have been used for over a century, more recently becoming automated biometric due to advancement in computing capabilities. Fingerprint identification is popular because of the inherent ease in acquisition, the numerous sources available for collection. In order to design a Fingerprint recognition system, the choice of feature extractor is very crucial and extraction of pertinent features from two-dimensional images of human finger plays an important role. A major challenge in Fingerprint recognition today is to select the low dimensional representative features and to reduce the search space for identification process.

The framework for the Fingerprint recognition system, as shown in Fig.1.1 consists of three phases. i) Feature extraction and representation phase, ii) Feature level Clustering, iii) Indexing and Fingerprint Matching. The three phases of fingerprint recognition framework are detailed in the following sections.

### 1.1 Fingerprint feature extraction

To detect the machine-readable representation completely capture the invariant and discriminatory information in the input measurements is the most challenging problem in representing fingerprint data. This representation issue constitutes the essence of system design and has far reaching implications on the design of the rest of the system. The unprocessed measurement values are typically not invariant over the time of capture and there is a need to determine salient features of the input measurement which both discriminate between the identities as well as remain invariant for a given individual. Thus, the problem of representation is to determine a measurement(feature) space which is invariant for input signals belonging to the same identity and which differ maximally for those belonging to different identities (higher interclass) variation and low interclass variation.

In [1], fingerprint features are classified into three classes. Level 1 features show macro details of the ridge flow shaped, Level 2 features(minutiae point) are discriminative enough for recognition, and Level 3 features(pores) complement the uniqueness of Level 2 features. The popular fingerprint representation scheme have evolved from an intuitive system developed by forensic experts who visually match the fingerprints. These schemes are either based on predominantly local landmarks(e.g. minutiae-based fingerprint matching systems[1] or exclusively global information (fingerprint classification based on the Henry System [2]). The minutiae-based automatic identification techniques first locate the minutiae points and then match their relative placement in a given finger and the stored template. The global representation of fingerprints (e.g. whorl, Left loop, right loop, arch, and tented arch) is typically used for indexing, and does not offer good individual discrimination. The global representation schemes of the fingerprint used for classification can be broadly categorized into four main categories: (i) knowledge-based, (ii) structure-based, (iii) frequency-based, and (iv) syntactic. The different type of fingerprint patterns are described below.

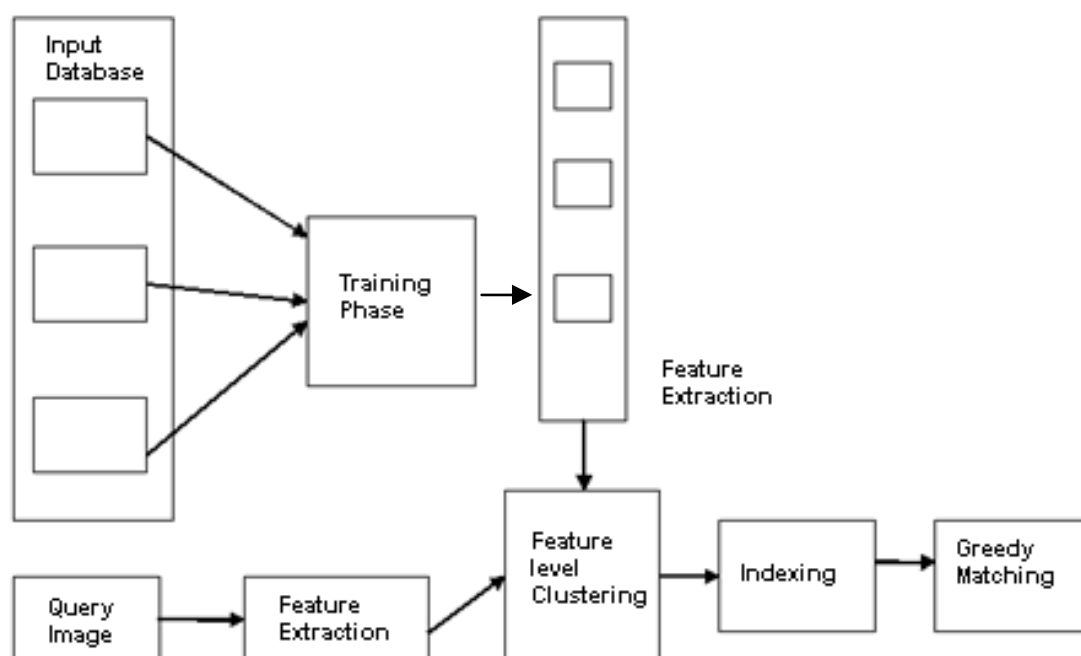


Fig. 1.1. Fingerprint Recognition System.

### 1.1.1 Global ridge pattern

A fingerprint is a pattern of alternating convex skin called ridges and concave skin called valleys with a spiral-curve-like line shape (Fig. 1.2). There are two types of ridge flows: the pseudo-parallel ridge flows and high-curvature ridge flows which are located around the core point and/or delta points. This representation relies on the ridge structure, global landmarks and ridge pattern characteristics. The commonly used global fingerprint features are:

- Singular points - discontinuities in the orientation field. There are two types of singular point. A core is the uppermost of the innermost curving ridge[3], and a delta point is the junction point where three ridge flows meet. They are usually used for fingerprint registration, fingerprint classification.

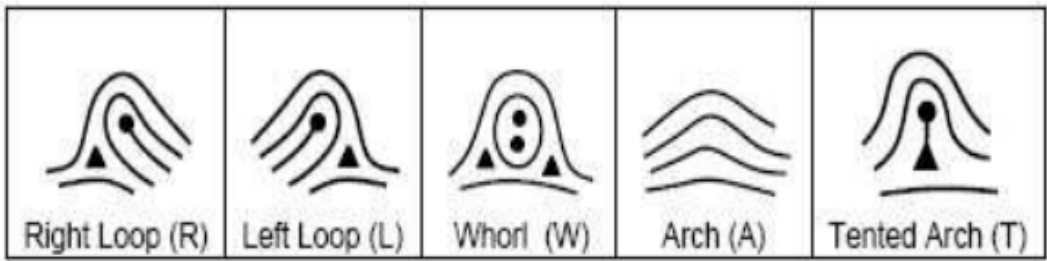


Fig. 1.2. Global Fingerprint Ridge Flow Patterns.

- Ridge orientation map - local direction of the ridge-valley structure. It is commonly utilized for classification, image enhancement, minutia feature verification and filtering.
- Ridge frequency map - the reciprocal of the ridge distance in the direction perpendicular to local ridge orientation. It is formally defined in [4] and is extensively utilized for contextual filtering of fingerprint images.

This representation is sensitive to the quality of the fingerprint images. However, the discriminative abilities of this representation are limited due to absence of singular points.

1.1.2 Local ridge pattern

This is the most widely used and studied fingerprint representation. Local ridge details are the discontinuities of local ridge structure referred to as minutiae. Sir Francis Galton was the first person who observed the structures and permanence of minutiae. Therefore, minutiae are also called „Galton details“. They are used by forensic experts to match two fingerprints. There are about 150 different types of minutiae categorized based on their configuration. Among these minutiae types, ridge ending and ridge bifurcation are the most commonly used, since all the other types of minutiae can be seen as combinations of ridge ending and ridge bifurcations. Some minutiae are illustrated in Fig.1.3.

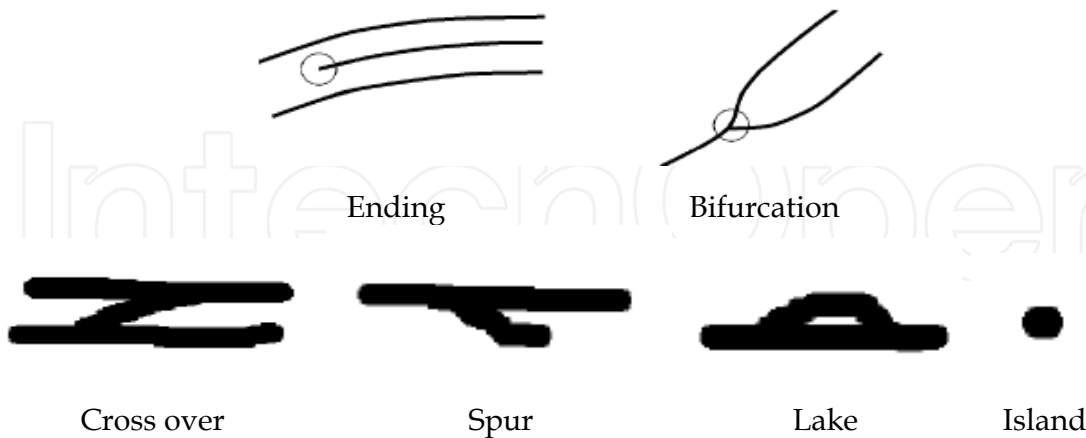


Fig. 1.3. Common Minutiae Types.

1.1.3 Intra-ridge detail

On every ridge of the finger epidermis, there are many tiny sweat pores and other permanent details. Pores are considered to be highly distinctive in terms of their number, position, and shape. However, extracting pores is feasible only in high-resolution fingerprint

images and with very high image quality. Therefore, this kind of representation is not adopted by currently depolyed automatic fingerprint identification systems. Some fingerprint identification algorithm(such as FFT) may require so much computation. Discrete Cosine Transform based algorithm may be the key to making a low cost fingerprint identification system.

### 1.2 Feature level clustering

With the increase in the size of the Fingerprint database, reliability and scalability issues become the bottleneck for low response time, high search and retrieval efficiency in addition to accuracy. Fingerprint classification refers to the problem of assigning fingerprints to one of several prespecified classes. It is an important stage in automatic fingerprint identification systems(AIFS) because it significantly reduces the time taken in identification of fingerprints, especially where accuracy and speed are critical. Traditionally Fingerprint identification systems claims identity of an individual by searching templates of all users enrolled in the database. These comparisons increase the data retrieval time along with the error rates. Thus a size reduction technique must be applied to reduce the search space and thus improve the efficiency. Conventionally databases are indexed numerically or alphabetically to increase the efficiency of retrieval. However, Fingerprint databases do not possess a natural order of arrangement which negates the idea to index them alphabetically/numerically. Reduction of search space in databases thus remains a challenging problem. Considering the classification issues [5], several methods have been proposed in the past couple of years to address the fingerprint classification issues. Most of these methods classify the images based on the ridges, local feature (i.e. minutiae) and global features (i.e. singular points). Model based approaches based on the global features (singular points) of the fingerprints have been found more effective in classifying the fingerprints into different known classes. Structure-based approaches based on the estimated orientation field in a fingerprint image can be found capable to classify the images into one of the five classes. The role of the estimated orientation field for fingerprint classification is generic. However, if the images are of poor quality then the orientation field estimation could not be done properly. Also, in such case difficulties encountered during extraction of other features like minutiae, finger code, Poincare index for singular points detection etc. Exclusive fingerprint classification is a traditional approach that has been widely investigated in the literature [5-7]. It classifies each fingerprint exclusively into one of the predefined classes such as Henry classes. Although it has some advantages such as human-interpretability, fast retrieval and rigid database partitioning, most automated classification algorithms are able to classify fingerprints into only four or five classes. Moreover, fingerprints are not evenly distributed in these classes. Thus, the exclusive classification cannot sufficiently narrow down the search of database. Most of the existing fingerprint classification approaches make use of the orientation image [8]. The main drawback of classification is that it is the supervised method where number of classes has to be known in advance. Further the data within each class is not uniformly distributed so the time required to search some classes is comparatively large. The limitations of classification can be addressed with unsupervised approach known as Clustering. It involves the task of dividing data points into homogeneous classes or clusters so that items in the same class are as similar as possible and items in different classes are as dissimilar as possible. Intuitively it can be visualized as a form of data compression, where a large number of samples are converted into a small number of representative prototypes.

### 1.3 Fingerprint database indexing

Identification refers to determine the identity of an individual from the database of persons available to the system. With the increase in the size of database, the number of false acceptances grows geometrically. Further, the time required to claim an identification is directly proportional to the size of the database. Thus efficiency of such systems can be improved either by minimising the error rates or by reducing the search space. The former is dependent on the efficiency of the algorithm and cannot be reduced considerably. Hence more emphasis can be laid on reducing the search space for improving performance as it is not possible to retrieve each element from probe set and compare with all elements of gallery set to determine the identity. There already exist few indexing schemes to partition the biometric database.

Rajiv Mukherjee et al [9] reported that iris database indexing generated from mean vector constructed from each row of the Iris Code and iris feature vector. Unsupervised clustering is performed on these index vectors. Puan et al [10] proposed iris indexing based on the iris color with chrominance components is generated as index code. Amit et al. [11] reported Indexing hand geometry database by pyramid technique. Gupta et al. [12] proposed an efficient indexing scheme for binary feature template with the help of B+ tree. Umarani et al. [13] employed a modified B+ tree biometric database indexing. The higher dimensional feature vector is projected to lower dimension and the reduced dimensional feature vector is used to index the database by forming B+ tree. All of the aforementioned tree-like data structures lead to "curse of dimensionality". While they work reasonably well in a 2-D or 3-D space, as the dimensionality of the data increases, the query time and data storage would exhibit an exponential increase.

### 1.4 Fingerprint matching

In order to improve fingerprint recognition performance, many techniques have been designed. The most popular matching strategy for fingerprint verification is minutiae matching. The simplest pattern of the minutiae-based representation consists of a set of minutiae, including ridge endings and bifurcation defined by their spatial coordinates. Each minutiae is described by its spatial location associated with the direction and minutiae type. In correlation based matching, correlation between the corresponding pixels is computed for different alignments. In feature based matching features of the fingerprint ridge pattern, texture information may be extracted more reliably than minutiae.

## 2. Significant fingerprint feature extraction with discrete cosine transform

### 2.1 Discrete cosine transform

In this section the discrete cosine transform that can be used for analyzing the Fingerprint image and to extract the local features in the transform domain is described. The Discrete cosine transform that has already been well established by Chaur-Heh for image compression [14] is extended in this model to extract the deterministic fingerprint feature. In order to extract the features of fingerprint image, block DCT-based transformation is employed. Each image is divided into sub blocks with  $(N \times N)$  size. There are  $n \times n$  coefficients in each block after DCT is applied. Only some of the DCT coefficients are to be computed for feature extraction, and they are enough to represent the information that are needed from the block. The equation used for the DCT calculation for each pixel is given as follows:



$$F(u,v) = \alpha(u)\alpha(v) \frac{2}{n} \sum_{y=0}^{n-1} \sum_{x=0}^{n-1} f(x,y) \cos \frac{(2x+1)u\pi}{2n} \cos \frac{(2y+1)v\pi}{2n} \quad (2.1)$$

$$\alpha(u) = \frac{1}{\sqrt{2}} \quad \text{if } u=0 \quad (2.2)$$

$$\alpha(u) = 1 \quad \text{otherwise}$$

$F(u,v)$  is the DCT domain representation of  $f(x,y)$  image and  $u,v$  represent vertical and horizontal frequencies that have values ranges from 0 to 7. The DCT coefficients reflect the compact energy of different frequencies. The first coefficient  $(0, 0)$ , called DC, is the mean of visual gray scale value of pixels of a block. And the other values denoted as AC coefficients, representing each spatial characteristic including vertical, horizontal and diagonal patterns. Having described the discrete cosine transform, the feature extraction technique is presented in the next section.

## 2.2 Statistical feature extraction

The feature set is based on a measure of DCT coefficients excluding DC  $(0, 0)$ . The features derived from the DCT computation are limited to an array of summed spectral energies within a block in frequency domain. The given input image is applied to DCT by considering non-overlapping  $N$  by  $N$  blocks (where  $N=8$ ).

To ensure adequate representation of the image, each block non-overlaps horizontally and vertically with the neighboring blocks, thus for an image which has  $N_R$  rows and  $N_C$  columns, there are  $N_B$  blocks found by following formula:

$$N_B = \left\lceil \frac{N_R}{N} \times \frac{N_C}{N} \right\rceil \quad (2.3)$$

The input image is divided into sub-image blocks:  $F(u, v)$ ,  $u = 1$  to  $N$  and  $v = 1$  to  $N$ , and then the DCT is performed independently on the sub-image blocks. For each sub-block considering AC coefficients, the DCT feature set is extracted. The proposed feature descriptor  $F$  consisting of the following features are denoted as feature set.

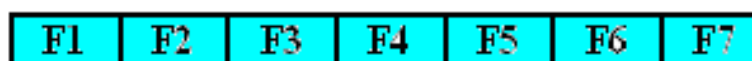


Fig. 2.1. Feature Set.

The features are denoted as Signal Energy, Edge Orientation, Ridge Frequency Estimation, Ridge Orientation Estimation, Non-Coherence Factor, Angular Bandwidth and Directional Strength Orientation respectively.

### 2.2.1 Signal energy

DCT is a reversible transform which obey the "Energy Preservation Theorem" - Total energy in pre-transform domain is equal to total energy in post transform domain [15]. The energy is one of the image properties using signal processing technique, and it means

characteristics of images. From the result of DCT transformed from image  $F(u,v)$  of size  $N \times N$ , the energy using DCT coefficients is defined as following equation [16].

$$F1 = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} |F(u,v)|^2 \tag{2.4}$$

2.2.2 Edge orientation

Edge is a strong feature for characterizing an image, which can be used to construct an important clus to understand the content of fingerprint image. The edge information from an image can be directly extracted by some simple measurements on the AC coefficients of each block in the compressed domain. Thus the following relationship determine the presence of edge in a block. Accurate edge orientation information is computed as follows:

$$F2 = \tan \theta = \frac{\sum_{v=1}^{N-1} F_{(0,v)}}{\sum_{u=1}^{N-1} F_{(0,u)}} \tag{2.5}$$

2.2.3 Ridge frequency estimation

The ridge frequency estimates the curve dominant flow in a block. The highest DCT peak of highest frequency spectrum is given by [16]:

$$F3 = \sqrt{u_0^2 + v_0^2} \tag{2.6}$$

where  $u_0, v_0$  is the coordinate of the highest DCT peak value.

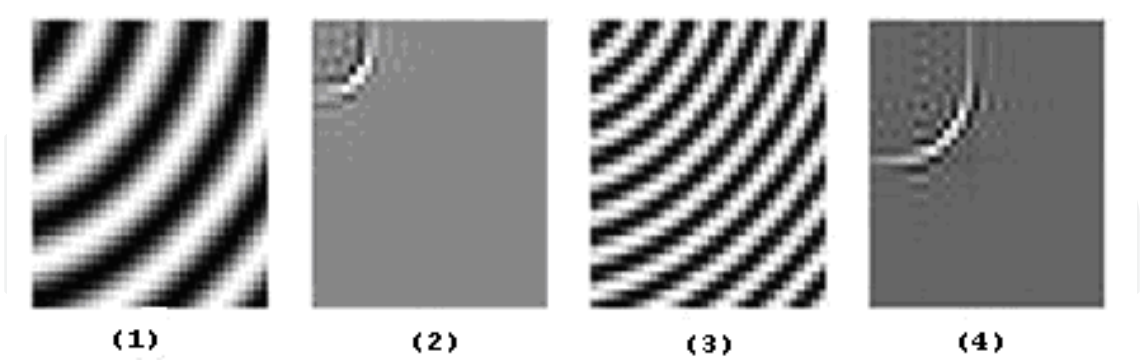


Fig. 2.2. 1) and 3) Represents blocks of a fingerprint image with different frequency, 2) and 4) DCT coefficients of (1) and (3) respectively.

2.2.4 Ridge orientation estimation

The dominant orientation of parallel ridges, are closely related to a peak-angle in DCT coefficients, where  $\phi$  is measured counterclockwise (if  $\phi > 0$ ) from the horizontal axis to the terminal side of the highest spectrum peak of highest frequency (DC is excluded). However  $\phi$  and  $\theta_0$  relationship is not one-to-one mapping. The ridge orientation which



varies in the range of 0 to  $\pi/2$  is projected into the peak-angle  $\theta$  which varies in the range of 0 to  $\pi$ . Relationship between ridge orientation  $\phi$  in spatial domain and peak angle  $\phi_0$  in frequency domain is given by:

$$F4 = \tan^{-1} \frac{v_0}{u_0} \quad (2.7)$$

Let F4 be  $\phi_0$ , then  $\phi_0 = \left| \frac{\pi}{2} - \theta_0 \right|$  where  $0 \leq \theta_0 \leq \pi$

### 2.2.5 Non-coherence factor

The factor represents how wide ridge orientation can be in the block that has more than one dominant orientation. This factor is in the range of 0 to 1, where 1 represents highly non-coherence or highly curved region and 0 represents orientation region. The non-coherence factor is given by:

$$F5 = \frac{\sum_{i,j \in N} \left| \sin(\theta_{u_c, v_c} - \theta_{u_i, v_j}) \right|}{N \times N} \quad (2.8)$$

where  $u_c, v_c$  is the center position of the block,  $u_i, v_j$  is the  $i^{th}$  and  $j^{th}$  position of neighbor blocks within  $N \times N$ .

### 2.2.6 Angular bandwidth

High ridge curvature occurs where the local ridge orientation changes rapidly, i.e. near cores and deltas. Away from these singular points, ridge curvature tends asymptotically to zero. In regions of higher curvature, a wider range of orientations is present. Away from singular points, the angular bandwidth is  $\pi/8$ . The angular bandwidth must equal to  $\pi$  is calculated by the following equation:

$$F6 = \sin^{-1}(F5) \quad (2.9)$$

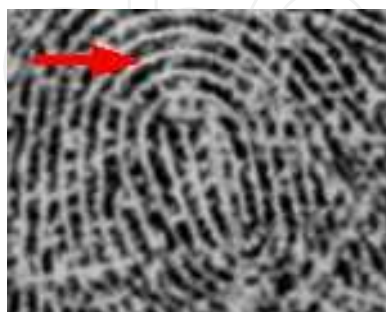


Fig. 2.3. Core Region.

### 2.2.7 Directional strength orientation

To identify the quadrant and avoid influence of interference, two 2-D perpendiculars diagonal vectors  $D_1$  and  $D_2$  are formed with size of  $5 \times 3$  pixels center at the peak position.

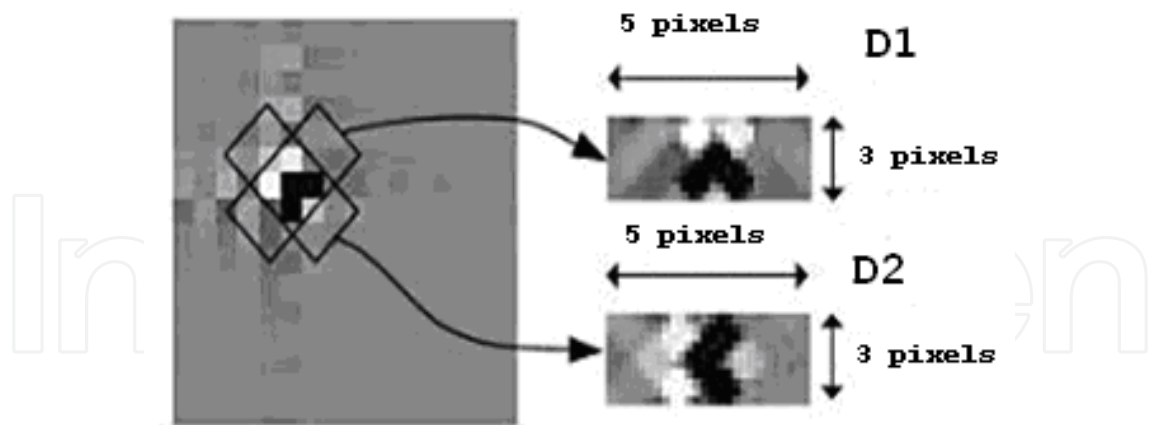


Fig. 2.4. 2-D perpendicular diagonal vectors at 45° and 135° .

The average directional strengths of each vector  $DS_i$

$$Max_{n=-1,0,1} \left| \frac{\sum_{m=-2}^2 D_{u_0+m,v_0+n}}{5} \right| = DS_i, \text{ where } i=1,2 \tag{2.10}$$

Then the quadrant can be classified and the actual fingerprint ridge orientation can be identified as [16]:

$$F7 = \begin{cases} \pi / 2 - F4 & \text{where } DS_1 \geq DS_2 \\ \pi - (\pi / 2 - F4) & \text{otherwise} \end{cases} \tag{2.11}$$

where  $u_o, v_o$  is the coordinate of the highest DCT peak value,  $u_c, v_c$  is the center position of the block,  $u_i, v_j$  is the  $i^{th}$  and  $j^{th}$  position of neighbor blocks within  $N \times N$ , perpendicular diagonal set  $D_1, D_2$  with size  $5 \times 3$  pixels and  $DS_1, DS_2$  average directional strength respectively.

The feature is extracted for each  $N \times N$  block and its average of the total number of blocks of that image is finally derived. The Non-Coherence Factor represents ridge orientation of how wide can be in that block. The maximum value occurred on a block represents highly curved region which is considered as a reference block for indexing. The significant statistical feature values are summerized in the following table.

Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7	IndexCode
680	60	83	2.0965	43	0.6839	21	9200002114
1205	62	92	1.9354	42	0.7269	20	1400000711
1089	65	83	2.2276	37	0.6879	20	1896304005
1116	64	87	2.4107	41	0.6758	21	2120284181
1127	64	88	2.2014	43	0.6962	19	4783545190

Table 2.1. Significant Statistical Feature Set.

3. Clustering categorical fingerprint features

3.1 Robust clustering (ROCK)

The main aim of cluster analysis is to assign objects into groups (clusters) in such a way that two objects from the same cluster are more similar than two objects from different clusters. There are different ways of classifying the cluster analysis methods. It can distinguish partitioning methods (denoted as flat) which optimize assignment of the objects into a certain number of clusters, and methods of hierarchical cluster analysis with graphical outputs which make assignment of objects into different numbers of clusters possible. The partitioning method includes k-means, fuzzy c mean, k-medians, k-medoids. The hierarchical method can be agglomerative (step-by-step clustering of objects and groups to larger groups) or divisive (step-by-step splitting of the whole set of objects into the smaller subsets and individual objects).

The ROCK [17], algorithm is an agglomerative hierarchical clustering algorithm for clustering categorical data. ROCK is based on links between pairs of data objects and the agglomerative process of merging clusters terminates either when there is no pair of clusters with links between them or when the required number clusters is obtained. Informally, the number of links between two tuples is the number of common neighbors they have in the dataset. After an initial computation of the number of links between the data objects, the algorithm starts with each cluster being a single object and keeps merging clusters based on a goodness measure for merging. The merging is continued till one of the following two criteria is reached.

- A specified number of clusters are obtained (or)
- No links remain between the clusters.

Instead of working on the whole dataset, clusters a sample randomly drawn from the dataset and then partitions the entire dataset based on the clusters from the sample.

3.2 Categorical fingerprint feature representation

In the cluster analysis phase, a cluster representative is generated to characterize the clustering result. However, in the categorical domain, there is no common way to decide cluster representative. Based on the assumption, the numerical data are converted into categorical data [22] as described below:

Orientation Range (in degrees)	Angle (in degrees)	Label
0 to 22.5 & 157.5 to 180	0	Q1
22.5 to 67.5	45	Q2
67.5 to 112.5	90	Q3
112.5 to 157.5	135	Q4

Table 3.1. Orientation Label.

Range	1	2	3	4	5	6	7	8	9
Label	F1	F2	F3	F4	F5	F6	F7	F8	F9

Table 3.2. Frequency Label.

Feature 1: Signal Energy  
By considering the first 5 minimum values from the training test, image is set to low else high.

$F1= \{high, low\}$

Feature 2: Edge Orientation

$F2= \{Q1, Q2, Q3, Q4\}$

Feature 3: Ridge Frequency

$F3= \{F1, F2, F3, F4, F5, F6, F7, F8, F9\}$

Feature 4: Ridge Orientation

$F4= \{Q1, Q2, Q3\}$

Algorithm: **Categorical Data**

Input: Numerical Data  $\{ N_1, N_2, N_3.....N_n \}$

Output: Categorical Data  $\{ C_1, C_2, C_3.....C_n \}$

- Step 1: Read the dataset N
- Step 2: Convert Numerical Data into Categorical Data based on condition
- Step 3: Assign Label to each numerical data

IMAGE NAME	SIGNAL ENERGY	EDGE ORIENTATION	RIDGE FREQUENCY	RIDGE ORIENTATION
Image 1	High	Q3	F1	Q2
Image 2	High	Q2	F2	Q3
Image 3	High	Q4	F2	Q1
Image 4	High	Q1	F3	Q1
Image 5	High	Q2	F4	Q2

Table 3.3. Categorical Data.

3.3 Feature level clustering algorithm

3.3.1 Robust clustering

Algorithm 1: ROCK

Procedure: Cluster(S,k)

Let S = {set of n sampled feature}

k = {number of clusters}

Input: Feature vector Fv

Output: Clusters C1, C2....., Cn

- 
- Step1: Read Fv
  - Step2: *Compute links*  
link:= compute\_links(S)
  - Step3: for each s ε S do  
q[s]:=build\_local\_heap(link,s)
-

---

**Step4:** Q:=build\_global\_heap(S,q)

**Step5:** Find the best cluster to merge

while size(Q)>k

do

u:=max(Q)

v:=max(q[u])

delete(Q,v)

w:=merge(u,v)

for each x ∈ q(u) ∪ q(v)

do

link[x,w]:=link[x,u]+link[x,v]

delete(q[x],u,v)

insert(q[x],w,g[x,w])

insert(q[w],x,g[x,w])

update(Q,x,q[x])

insert(Q,w,q[w])

deallocate()

**Step 6:** End

---

### 3.3.2 K-Mean clustering for categorical data

Let  $X = (F_1, F_2, F_3, F_4, F_5)$  be a feature vector where the scalars  $F_1, F_2, F_3, F_4, F_5$  represent the individual features of each object. Let  $V = (v_1, v_2, \dots, v_k)$  be the set of  $k$  clusters where  $v_1, v_2, \dots, v_k$  represent the centroid of each cluster [18].

The centroid of a cluster is defined as:

$$v_k = \frac{\sum_{i=1}^N F_i \in v_k}{|v_k|} \quad (3.1)$$

The above formula simply computes the mean or the average of the values of features belonging to the particular cluster.

The objective function to be minimized is defined below:

$$J(X, V) = \min \sum_{k=1}^K \sum_{i=1}^{N_k} d(F_i, v_k) \quad (3.2)$$

where Euclidean Distance:

$$d(v_k, F_i) = \sqrt{\sum_{j=1}^N (F_{ij} - v_{kj})^2}$$

For categorical attributes, the similarity measure is defined as:

$$d(F_i, v_k) = \sum_{i=1}^N (d_{ij}^n - v_{kj}^n)^2 + wt \times \sum_{i=1}^C \delta(d_{ij}^c, v_{kj}^c) \quad (3.3)$$

where  $\delta(a, b) = 0$  for  $a=b$  and  $\delta(a, b) = 1$  for  $a \neq b$ .

$d_{ij}^n$  and  $v_{kj}^n$  are values of numeric attributes and  $d_{ij}^c$  and  $v_{kj}^c$  are values of categorical attributes for the object  $o$  and cluster  $k$ .  $N_k$  is the number of elements in cluster  $k$ . The equation can be rewritten as:

$$J(X, V) = \sum_{k=1}^p \sum_{i=1}^{N_k} \left( \sum_{j=1}^N (d_{ij}^{na} - v_{kj}^{na})^2 + wt \times \sum_{j=1}^C \delta(d_{ij}^c, v_{kj}^c) \right) \quad (3.4)$$

### Algorithm 2: K Means- for Categorical Data

**Inputs:**

$I = \{i_1, i_2, \dots, i_k\}$  (Instances to be clustered)

$N$  (Number of clusters)

**Outputs:**

$C = \{c_1, \dots, c_n\}$  (Cluster Centroids)

$m: I \rightarrow C$  (Cluster Membership)

---

**Step 1:** Set  $C$  to initial value (e.g. random selection of  $I$ )

**Step 2:** For each  $i_j \in I$

$$m(i_j) = \arg \min_{k \in \{1..n\}} d(i_j, c_k) + wt \times \sum_{k=1}^C \delta(d_{ij}^c, C_k^c)$$

End

**Step 3:** While  $m$  has changed

For each  $j \in \{1..n\}$

Recompute  $i_j$  as the centroid of  $\{i \mid m(i) = j\}$

End

Return  $C$

---

### 3.3.3 Fuzzy C-means clustering for categorical data

A variant of K-Means that performs fuzzy clustering is called Fuzzy C-Means (FCM). The output of a fuzzy clustering is not a partition but still a clustering with each object having a certain degree of membership to a particular cluster [20]. The objective function being minimized in FCM is given below:

$$J(X, V) = \min \sum_{k=1}^K \sum_{i=1}^{N_k} (U_{ki}^m) d(F_i, v_k) \quad (3.5)$$

where  $d(v_k, F_i)$  is the distance function similar to the one discussed above and  $u_{ki}$  is the degree of membership of object  $i$  to cluster  $k$  subject to the following constraints:

$$u_{ki} \in [0, 1] \text{ and } \sum_{k=1}^K u_{ki} = 1, \forall i,$$

which means that the sum of the membership values of the objects to all of the fuzzy clusters must be one.  $m$  is the fuzzifier which determines the degree of fuzziness of the resulting



clusters. The objective function can be minimized by using Lagrange multipliers. By taking the first derivatives of  $J$  with respect to  $u_{ik}$  and  $v_k$  and setting them to zero results in two necessary but not sufficient conditions for  $J$  to be at the local extrema.

The result of the derivation is given below:

$$v_k = \frac{\sum_{i=1}^N u_{ki}^m x_i}{\sum_{i=1}^N u_{ki}^m} \quad (3.6)$$

which computes the centroid of cluster  $k$ .

$$u_{ki} = \frac{1}{\sum_{j=1}^K (d(v_k, x_i) / d(v_k, x_j))^{m-1}} \quad (3.7)$$

computes the membership of object  $i$  to cluster  $k$ . For categorical data, Let  $X = (F_1, F_2, F_3, F_4, F_5)$  be a set of categorical objects.

Let  $F_k = [F_{k1}, F_{k2}, \dots, F_{kp}]$  and  $F_l = [F_{l1}, F_{l2}, \dots, F_{lp}]$  be two categorical objects.

The matching dissimilarity between them is defined as:

$$D(F_k, F)_l = \sum_{j=1}^p \delta(F_{kj}, F_{lj}) \quad (3.8)$$

where  $\delta(a, b) = 0$  for  $a=b$  and  $\delta(a, b) = 1$  for  $a \neq b$ .

### Algorithm 3: Fuzzy C Mean- for Categorical Data

#### Inputs:

$I = \{i_1, i_2, \dots, i_k\}$  (Instances to be clustered)

$N$  (Number of clusters)

#### Outputs:

$C = \{c_1, \dots, c_n\}$  (Cluster Centroids)

---

Step 1: Select  $m$  and  $\varepsilon$  to initial value.

Step 2: Initialize the modes  $v_i$  ( $1 \leq i \leq c$ )

Step 3: Calculate the membership degrees  $u_{ik}$  and determine  $J_m(U, V)$

Step 4: Set  $J_m^{old}(U, V) = J_m(U, V)$

Step 5: Update the cluster centers  $v_i$  ( $1 \leq i \leq c$ )

Step 6: Calculate the membership degrees  $u_{ik}$  ( $1 \leq i \leq c, 1 \leq k \leq n$ )

Step 7: Update the  $J_m(U, V)$

Step 8: If  $|J_m(U, V) - J_m^{old}(U, V)| \leq \varepsilon$  Go to Step 4

---

3.4 Similarity measures

The algorithm finds all the combinations of feature values in an object, which represent a subset of all the attribute values, and then groups the database using [21] the similarity of these combinations by constructing the Jaccard coefficient matrix. Objects in a cluster have not only similar attribute value sets but also strongly associated attribute values (feature). The similarity measure of the ROCK algorithm is found using Jaccard Coefficient

$$sim(o_i,o_j)=\frac{o_i\cap o_j}{o_i\cup o_j}$$

The measure is computed using Single, Complete and Average Linkage.

	I1	I2	I3	I4	I5	I6
I1	1	0.4	0.4	0.6	0.8	0.4
I2	0.4	1	0.2	0.4	0.4	0.8
I3	0.4	0.2	1	0.6	0.4	0.8
I4	0.6	0.4	0.6	1	0.2	0.4
I5	0.8	0.4	0.4	0.2	1	0.6
I6	0.4	0.8	0.8	0.4	0.6	1

	I1	I2/I3	I4/I5	I6
I1	1	0.4	0.6	0.4
I2/I3	0.4	1	0.2	0.4
I4/I5	0.6	0.2	1	0.4
I6	0.4	0.4	0.4	1

	I1	I2/I3/ I4/I5	I6
I1	1	0.2	0.4
I2/I3/I4/I5	0.2	1	0.4
I6	0.4	0.4	1

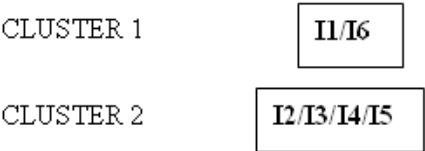


Table 3.4. Jaccard Coefficient Matrix: Single Linkage.

An exhaustive analysis of the proposed method in terms of false acceptance rate (FAR) vs classification accuracy was also carried out FVC datasets. The results and its comparison with its other counterparts are reported in Fig. 3.1.

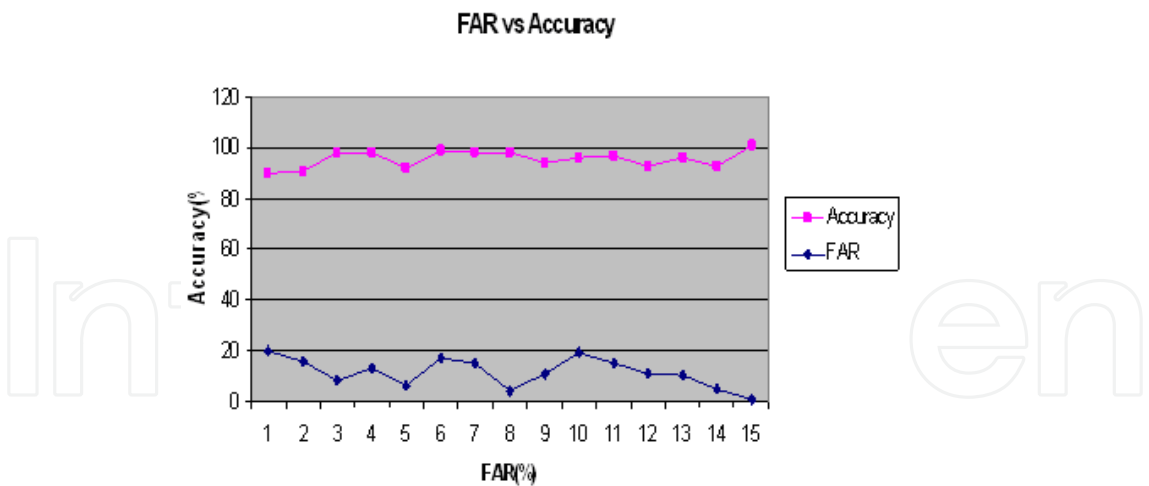


Fig. 3.1. FAR Vs Accuracy for fingerprint classification.

An algorithm performs well compared to another one has higher intra cluster similarity and lower inter cluster similarity compared to other algorithm. From the graphs (Fig 3.2 & 3.3) it is obvious that ROCK algorithm consistently performs better than other algorithms in both intra cluster and inter cluster similarity measure. For  $k = 2$  and 3 bisecting K Means performs slightly better than ROCK algorithm in intra cluster similarity measure but overall ROCK algorithm outperforms other algorithms.

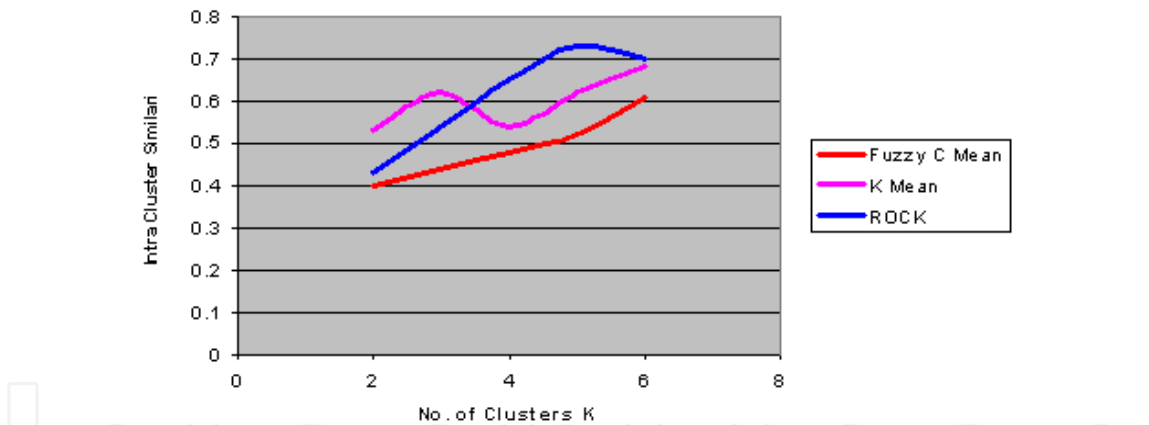


Fig 3.2. Intra Cluster Similarity for various algorithms on fingerprint database.



Fig 3.3. Inter Cluster Similarity for various algorithms on fingerprint database.

The FAR and FRR curve as claimed by the algorithm is shown in (Fig. 3.4). To evaluate the matching performance of the algorithm on database of FVC 2004, experiments have been conducted. The experiment considers all fingerprints in the database, leading to 95 matches and 5 non-matches. In this case, FAR and FRR values were 30-35% approximately and the equal error rate is 0.36.

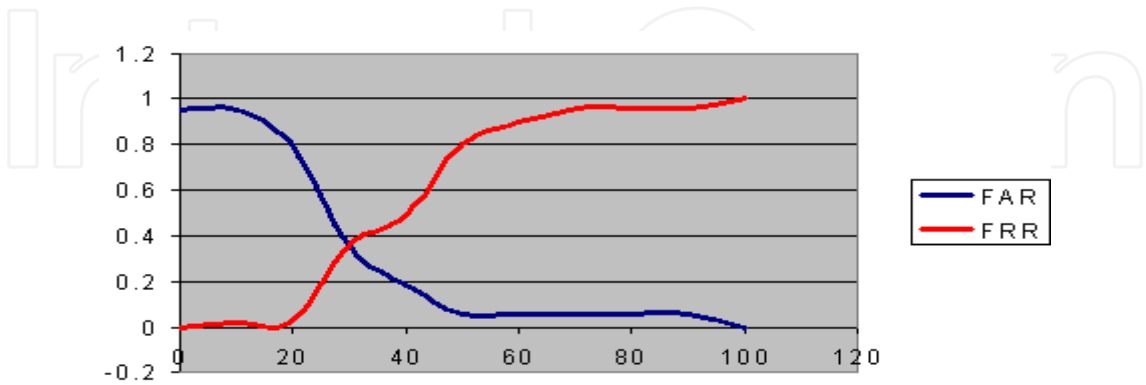


Fig. 3.4. FAR and FRR Curve.

4. Indexing and matching

4.1 Indexing scheme

It is expected that the query response time should depend upon the templates similar to the query template and not the total number of templates in the database. Thus the database should be logically partitioned such that images having similar patterns are indexed together. To search the large visual databases, content based image indexing and retrieval mechanism based on sub block of DCT coefficients are used. The scheme provides fast image retrievals. In this indexing technique feature vector which comprises of global and local features extracted from offline fingerprint databases are used by Robust clustering technique to partition the database. As biometric features posses no natural order of sorting, thus it is difficult to index them alphabetically or numerically. Hence, indexing is required to partition the search space. At the time of identification the fuzziness criterion is introduced to find the nearest clusters for declaring the identity of query sample. The system is tested using bin-miss rate and performs better in comparison to traditional k-means approach.

This method relies on the use of a small set of DCT coefficients for indexing a fingerprint image. By considering a reference block extracted using non-coherence factor, we only extract  $F_{0,0}, F_{0,1}, F_{1,0}, F_{2,0}, F_{1,1}, F_{0,2}, F_{0,3}, F_{1,2}, F_{3,0}$  for indexing and retrieval. All these coefficients are quantized finally.

The fingerprint image associated with an identity is subjected to the procedure above in order to generate the index code. The index code is stored in the fingerprint database along with the identity of the associated fingerprint. When a query image is presented to the system, initially image get categorized to which cluster belong to, in turn search with index code reference. During training phase, the images taken are grouped under various bins based on their index key. During testing phase, the index key is generated for the query image and the respective bin is located. The fingerprint template of the test image is

compared only with the fingerprint templates matched bin. The accuracy analysis of the recognition system with and without indexing (Exhaustive search) is presented in Table 4.1.

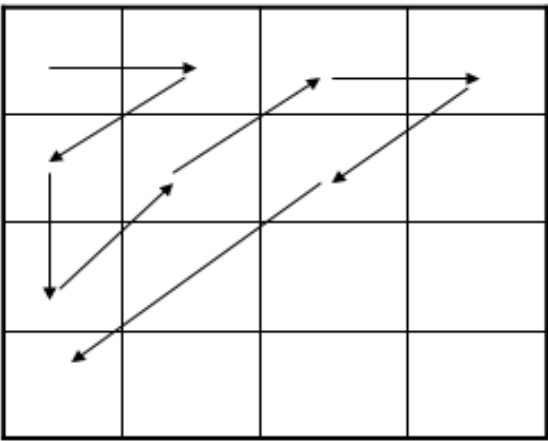


Fig. 4.1. Index Code.

Method	Without indexing	With indexing
Accuracy (%)	91	97

Table 4.1. Accuracy of recognition system with and without indexing scheme.

Table 4.1 shows that the system with indexing provides 97% accuracy; whereas the system without indexing scheme provides only 91% of accuracy. This indexing mechanism considerably increases the recognition accuracy of the proposed feature extraction scheme. Also the number of comparisons made during the testing phase is reduced in the proposed recognition system with the help of indexing than the system using exhaustive search.

4.2 Greedy matching

The approach used is based on the general combination technique [22]. The function called greedy matching algorithm which integrates different matching criteria based on heterogeneous features. Different representations are often related to different features of the fingerprints in determining appropriate matching criteria. Generally, matching algorithm has to solve two problems: correspondence and similarity computation. For the correspondence problem, assign two descriptors: edge-based and ridge-based, and use an alignment-based greedy matching algorithm to establish the correspondences between blocks. From the similarity computation, a matching score is computed.

4.2.1 Alignment

Alignment is a crucial step for the proposed algorithm, as misalignment of two fingerprints of the same finger certainly produces a false matching result. In the proposed approach, two fingerprints are aligned using the top *n* most similar orientation pair. If none of the *n* pairs is correct, a misalignment occurs. The test is conducted using orientation-based, frequency-based and combined descriptors. It can be concluded that alignment based on combined descriptors is very reliable.

#### 4.2.2 Block pairing

For matching, the fingerprint should be cropped to  $64 \times 64$  from the obtained reference block. With the registered orientation field with respect to ridge and edge, the procedure identifies the corresponding orientation block pairs is straightforward. Let denote the corresponding orientation block pair,  $p_k$  from test fingerprint, block  $q_k$  from template fingerprint respectively. The similarity degree  $S(p_k, q_k)$  of the two blocks is calculated as follows:

To compute the similarity between two sampling blocks, first compute the similarity of orientation and the similarity of frequency as follows:

$$s_o(p_k, q_k) = e^{-|\lambda(\alpha_k - \beta_k)|/(\pi/16)}$$

$$s_f(p_k, q_k) = e^{-|v_k - w_k|/3}$$

Then the similarity is defined as

$$s_t(p, q) = \omega s_o(p, q) + (1 - \omega) s_f(p, q) \quad (4.1)$$

where  $\omega$  is set to 0.5.

To compute the similarity between two sampling blocks, the similarity of non-coherence as follows:

$$s_m = \frac{m_p + 1}{M_p + 1} \cdot \frac{m_q + 1}{M_q + 1} \quad (4.2)$$

where  $m_p$  and  $m_q$  represent the number of matching factor of  $N(p)$  and  $N(q)$ , respectively, and  $M_p$  and  $M_q$  represent the total number of non-coherence of  $N(p)$  and  $N(q)$  that should be matching respectively. All terms plus 1 means that two central blocks  $p$  and  $q$  are regarded as matching. Here  $m_p$  and  $m_q$  are different because do not establish one-to-one correspondence. Since orientation-based descriptors and frequency-based descriptors capture contemporary information, further improve the discriminating ability of descriptors by combining two descriptors using the product rule:

$$S_c = S_t \cdot S_m \quad (4.3)$$

$$L = \{(S_c(p, q))\}_{p=1 \dots N1; j=1 \dots N2}$$

A list is used to store the normalized similarity degrees and indices of all block pairs. Elements in  $L$  are sorted in decreasing order with respect to  $S_c(p, q)$ . The first block pair  $(p1, q1)$  in  $L$  is used as the initial block pair, and two blocks are aligned using the initial pair. A pair of block is said to be matchable, if they are close in position and the difference of direction is small. The greedy matching algorithm is used to find match. Two arrays  $flag1$  and  $flag2$  are used to mark block that have been matched, in order that no block can be matched to more than one block. As the initial pair is crucial to the matching algorithm, the block pairs of the top  $Na$  elements in  $L$  are used as the initial pairs, and for each of them, a matching attempt is made. Totally  $Na$  attempts are performed,  $Na$  scores are computed and the highest one is used as the matching score between two fingerprints.



### 4.3 Matching process

**Input:** Orientation Value, Frequency Value, Non-coherence Factor ( $i_0, i_1, \dots, i_n$ ), ( $j_1, j_2, \dots, j_n$ )

**Output:** Matching Score (MS)

**function GreedyMatch**( $i_0, j_0$ )

**Step1:** Initialize  $flag1$  and  $flag2$  with 0;

**Step 2:**  $flag1[i_0] = 1$ ;

$flag2[j_0] = 1$ ;

**Step 3:** for  $m = 1$  to  $N_1 \times N_2$

$i = L(m).i$ ;  $j = L(m).j$ ;

if ( $flag1[i]=0$ ) & ( $flag2[j]=0$ ) & ( $p_i$  and  $q_j$  are matchable)

**Step 4:** Insert ( $i, j$ ) into MP;

**Step 5:**  $flag1[i] = 1$ ;

$flag2[j] = 1$ ;

end if

end for

### 4.4 Conclusion

Fingerprint identification is one of the most well-known and publicized biometrics. Because of their uniqueness and consistency over time, fingerprints have been used for identification for over a century, more recently becoming automated due to advancements in computing capabilities. The fingerprint image is divided into sub-blocks and allow evaluating the statistical features from the DCT Coefficients. The feature space emphasizes meaningful global information by considering AC coefficients. The presented approach requires only a small number of parameters for global ridge pattern. The numerical data transformed into categorical data to produce the discrimination between the intra and inter class cluster with ROCK based algoirhtm. The system is performing comparatively superior as compared to K-Means and Fuzzy C Mean clustering technique for categorical attributes. The performance of the ROCK method for classification based on complete linkage is better than the other linkage measures. The matching results based on the FVC database using testing data sets for images with different orientations and translation shows the system outperforms better in recognition rate and requires short time for both the training and querying process.

### 5. References

- [1] A. K. Jain, L. Hong, S. Pankanti, and R. Bolle. "An identity authentication system using fingerprints". Proc. IEEE, 85(9):1365–1388, 1997.
- [2] A. K. Jain, S. Prabhakar, and L. Hong, "A Multichannel Approach to Fingerprint Classification", IEEE Trans. Pattern Anal. and Machine Intell., Vol. 21, No. 4, pp. 348-359, 1999.
- [3] L. Hong, Y. Wan, and A. K. Jain. "Fingerprint image enhancement: Algorithms and performance evaluation". IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8):777–789, 1998.
- [4] J. I.T. "Principle Component Analysis". Springer-Verlag, New York, 1986.

- [5] Ballan Meltem, Sakarya Ayhan, and Evans Brian, "A Fingerprint Classification Technique Using Directional Images," *Mathematical and Computational Applications*, pp. 91-97, 1998.
- [6] Cho Byoung-Ho, Kim Jeung-Seop, Bae Jae-Hyung, Bae In-Gu, and Yoo Kee-Young. "Fingerprint Image Classification by Core Analysis," *Proceedings of ICSP*, 2000.
- [7] Jain Anil, Prabhakar Salil, and Hong Lin, "A Multichannel Approach to Fingerprint Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 348-359, 1999.
- [8] Ji Luping, and Yi Zhang, "SVM-based Fingerprint Classification Using Orientation Field," *3<sup>rd</sup> International conference on Natural Computation*, vol. 2, pp. 724- 727, 2007.
- [9] Rajiv Mukherjee and Arun Ross, "Indexing iris database," *International Conference on Pattern Recognition*, Vol.54, pp. 45-50, 2008.
- [10] Puan N B and Sudha, "A novel iris database indexing method using the iris color," *IEEE Conference on Industrial Electronics and Applications*, Vol.4, pp. 1886-1891, 2008.
- [11] Amit Mhatre, Sharat Chikkerur and Venu Govindaraju, "Indexing Biometric Databases Using Pyramid Technique," *International Conference on Audio-Video Biometric Person Authentication*, Vol. 354, pp.39-46, 2005.
- [12] Gupta P, Sana A, Mehrotra H and Jinshong Hwang C, "An efficient indexing scheme for binary feature based biometric database," *SPIE Biometric Technology for Human Identification*, Vol. 2, pp. 200-208, 2007.
- [13] Umarani Jayaraman, Surya Prakash and Phalguni Gupta, "An efficient technique for indexing multimodal biometric databases," *International Journal of Biometrics*, Vol.1, No.4, pp.418-441, 2009.
- [14] Chaur-Heh Hsieh, "DCT - Based codebook design for vector quantization of images," *IEEE Transaction on circuits and systems for video technology*, vol.2, pp. 456- 470, 1992.
- [15] Tienwei Tsai, Yo-Ping Huang and Te-Wei Chiang, "Dominant Feature Extraction in Block-DCT Domain", *IEEE Trans.*, 2006.
- [16] Suksan Jirachaweng and Vuctipong Areekul, "Fingerprint Enhancement Based on Discrete Cosine Transform", Springer Berlin / Heidelberg, Volume 4642/2007
- [17] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim, "ROCK: A Robust Clustering Algorithm for Categorical Attributes" *Proc. Of the 15th Int'nl Conference on Data Engineering*, 2000.
- [18] S K Gupta, K.Sambasiva and Vasuda Bhatnagar, "K-means clustering algorithm for categorical attributes", Springer-Verlag
- [19] Kudov.P, Rezankova.H, Husek.D and Snasel.V, "Categorical Data Clustering Using Statistical Methods and Neural Networks"
- [20] George E. Tsekouras, Dimitris Papageorgiou, Sotiris Kotsiantis, Christos Kalloniatis, and Panagiotis Pintelas, "Fuzzy Clustering of Categorical Attributes and its Use in Analyzing Cultural Data", *World Academy of Science, Engineering and Technology* 2005.
- [21] Shyam Boriah, Varun Chandola and Vipin Kumar, "Similarity Measures for Categorical Data: A Comparative Evaluation", In *Proceedings of 2008 SIAM Data Mining Conference*, April 2008.

- [22] W. Sheng, G.Howell, M.C.Fairhur, F.Derav, K.Harmer, "Consensus fingerprint matching with genetically optimised approach", Pattern Recognition, Elsevier 2009.

IntechOpen

IntechOpen



## **Biometric Systems, Design and Applications**

Edited by Mr Zahid Riaz

ISBN 978-953-307-542-6

Hard cover, 262 pages

**Publisher** InTech

**Published online** 21, October, 2011

**Published in print edition** October, 2011

Biometric authentication has been widely used for access control and security systems over the past few years. The purpose of this book is to provide the readers with life cycle of different biometric authentication systems from their design and development to qualification and final application. The major systems discussed in this book include fingerprint identification, face recognition, iris segmentation and classification, signature verification and other miscellaneous systems which describe management policies of biometrics, reliability measures, pressure based typing and signature verification, bio-chemical systems and behavioral characteristics. In summary, this book provides the students and the researchers with different approaches to develop biometric authentication systems and at the same time includes state-of-the-art approaches in their design and development. The approaches have been thoroughly tested on standard databases and in real world applications.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

D. Indradevi (2011). Efficient Fingerprint Recognition Through Improvement of Feature Level Clustering, Indexing and Matching Using Discrete Cosine Transform, *Biometric Systems, Design and Applications*, Mr Zahid Riaz (Ed.), ISBN: 978-953-307-542-6, InTech, Available from:  
<http://www.intechopen.com/books/biometric-systems-design-and-applications/efficient-fingerprint-recognition-through-improvement-of-feature-level-clustering-indexing-and-match>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen