CHAPTER 7

Indexing of Fingerprint Database

7.1. Fingerprint Database

In the modern digital world, biometrics occupy most of the part of the system. When we access a record from a big database, it may take more time compared to the small size of the database. Record accessing time depends on the performance of the indexing technique. For big biometric database, an effective indexing technique is required for fast processing. When the system needs fingerprint comparison, the input fingerprint indexed value will compare with the indexing value of the whole fingerprint database and select set of candidates. The final comparison will be done using a fingerprint matching algorithm between the fingerprint and the set of selected candidates. For this, the indexing has the responsibility to access all the similar set of candidates from the database. Many studies of fingerprint database are found in the literature [72, 73, 74, 75, 76]. Still, there is a need for fast indexing technique for fingerprint comparison.

The proposed fingerprint recognition system used distance feature (numerical values) which are stored in the database. The model of the template database is shown in table 7.1. From the table, it shows that each record contains 22 tuples. The first twenty tuples represent the distance feature and the last two tuples are the indexing values.

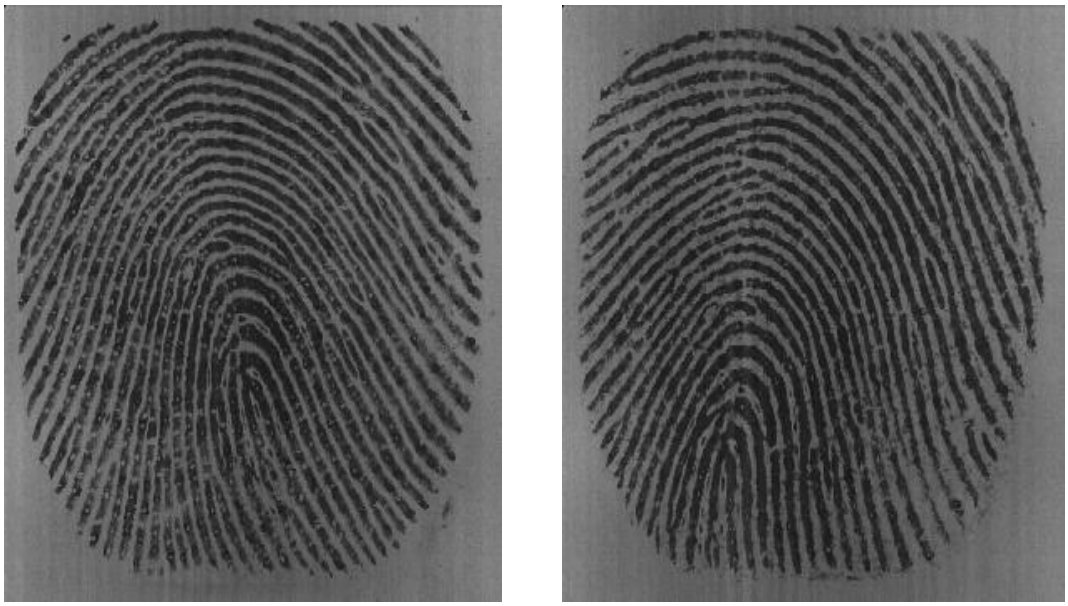
Sl no	Distance Feature																				Indexed value	
	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22
1	29.4	31.9	32	33.3	34	34.5	36.4	42.6	43.6	46	46.3	54.2	56.8	58.1	58.5	62.2	68	68.9	73.3	82.2	***31.1***	***74.8***
2	24.7	25.9	26	28.2	29.5	29.7	30.4	30.6	30.8	32.5	32.6	38	39	45	47.8	52.3	63.2	65.3	76.2	76.4	***25.5***	***72.6***
3	20.8	40.4	41	43.5	44.8	46.3	47.3	49.3	49.6	52.3	54.4	57.2	57.8	58.1	58.1	58.8	59.6	60	60.4	61.5	***34***	***60.6***
4	12.8	17.8	28.8	30.4	30.5	31.8	34.6	37.8	38.2	39.9	40.3	49.6	52.4	53.6	53.9	54.7	55.5	55.9	56.6	58.3	***59.4***	***56.9***
5	21.9	25.2	41.7	51.3	58.3	60.6	64.1	64.3	64.8	65	65.2	65.7	66.4	67.2	68.4	69.4	70.8	70.8	72.4	73.4	***29.3***	***72.2***
6	23	24.6	26.9	29.6	31.1	32.3	32.3	33.8	34.4	36.3	36.8	37.4	39.2	39.2	41.2	42	42.7	45.1	45.6	46.3	***24.8***	***45.6***
7	15.8	17	18.8	19.2	25.5	27.8	35.8	57.7	58.5	63.5	64.8	66.4	73.6	81.5	86.3	87	89.6	91.5	93.4	94.1	***17.2***	***93***
8	13.6	19.8	28.3	62.7	65	65.7	68.1	70.6	70.9	71.5	71.6	73.3	74.2	74.3	75.6	77	77.8	78.5	79.4	82.8	***20.5***	***80.2***
9	13.4	20.6	45.4	47.4	50.5	55.5	56.5	60.9	62.9	75.9	77.4	84.3	86.1	86.8	91.5	91.6	91.9	92.1	93.9	94.1	***26.5***	***93.3***
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
n^{th}	20.2	57.2	59.6	62.7	64.2	64.4	67.5	68	70.7	74	75.2	77.8	77.8	80.1	80.6	80.6	81.3	82.9	83.8	85.4	***45.5***	***84***

Table 7.1. Proposed template database model for fingerprint recognition system

Where T1, T2,T3…T20 are distance feature and T21 and T22 are the indexed values.

7.2 Fingerprint Distance Feature

The distance features are the Euclidean distance from the core to minutiae point. Minutiae are either ridge ending or ridge bifurcation. The distance features are arranged in an ascending order, starting from the nearest to the core. The table 7.2 is the examples of distance feature extract from the fingerprint image shown in figure 7.1 which shows two fingerprint images from the same finger. Due to the different position and scaling, the distance features taken from the same fingers at different time, shown in table 7.2 are not similar. The difference value of the distance feature is varies from 0.5 to 0.8. The proposed algorithm considers an error tolerance 'K' for this reason.



(i) (ii)

Fig 7.1. The two fingerprint images from same finger

Table 7.2: Distance feature

(i) Distance feature from fig. 7.1(i)

Sl no	Distance Feature
1	29.4108823397055
2	31.9530906173409
3	32.0156211871642
4	33.3016516106934
5	34.0587727318528
6	34.5398320783411
7	36.4005494464026
8	42.6380112106557
9	43.6004587131833
10	46.0977222864644
11	46.3249392876019
12	54.2309874518250
13	56.8858506133116
14	58.1893461039046
15	58.5491246732178
16	62.2896460095897
17	68.0000000000000
18	68.9637585982667
19	73.3348484691964
20	82.2009732302483

(ii) Distance feature from fig. 7.1(ii)

Sl no	Distance Feature
1	28.6530975637888
2	28.6836784620308
3	29.2061637330205
4	29.7321374946370
5	30.0832179129826
6	34.5253530032641
7	35.1710107901380
8	40.4474968323134
9	43.3243580448689
10	46.5295604965274
11	46.6904701197150
12	52.7730992078350
13	54.3323108288245
14	54.4058820349418
15	56.5685424949238
16	61.6846820531645
17	72.6223106214612
18	75.2927619363243
19	76.5310394545899
20	77.2010362624751

7.3 Distance Feature Indexing Technique

The proposed indexing technique is a very simple, fast and powerful indexing technique. In this technique, fingerprint distance feature are arrange in ascending order. Different fingerprints have different numbers of distance feature, but stored only the first 20 because in worst case (distorted fingerprint images), according to the study, the minimum number of distance feature of a fingerprint is around 20 to 30. So, the study selected the smallest one. In the stored record, there will be 22 tuples including the indexing values of each records. The first three average values of the tuples (T1,T2 and T3) are stored at 21st tuples and the last three

average values tuples (T18,T19 and T20) are stored at 22nd tuples. The proposed algorithm will select the smaller one among the last two tuples for indexing.

7.3.1 Indexing value

Let I_a be a 2D fingerprint image with a core point. Let M_{ij}^{a} be the minutiae point of Image I_a, where i is the i^{th} minutiae points and j=1,2,3,4. A row comprises of four points $< d_{i,1}^{a}, X_{i,2}^{a}, Y_{i,3}^{a}, b_{i,4}^{a} >$, where d_i is a distance of i^{th} minutiae point from one core point (X_c^a, Y_c^a) and it is calculated using Euclidean distance shown in equation (6). A set (first minimum 20) of distance, d_i^a from I_a is shown in equation (8).

$$T(i) = d_1^a, d_2^a, d_3^a .....d_{20}^a ----(8)$$

Where T(i) is the tuples which contain distance feature in ascending order and further two more tuples i.e., T(21) and T(22) are added by using Equations 9 and 10.

$$T(21) = average(T(1), T(2), T(3)) ----(9)$$

$$T(22) = average(T(18), T(19), T(20)) ----(10)$$

7.3.2 Candidate selection

Among these two equation 9 and 10, the smaller average value are picking up for indexing. The below figure describes how T is constructed.

T(1)	T(2)	T(3)	…	T(18)	T(19)	T(20)	T_{fa}	T_{la}

Where T_{fa} =T(1)+T(2)+T(3)/3 and T_{la} = T(18)+T(19)+T(20)/3.

As the fingerprint has a flexible property, the proposed indexing algorithm consider an error tolerance ‘K’. It is used in two conditions:

1.When the first two average values of two different images such as absolute of {[T1(1)+T1(2)+T1(3)/3]- [T2(1)+T2(2)+T2(3)/3]} is less than or equal to the error ‘K’.

2.When the second two average values of two different images such as absolute of {[T1(18)+T1(19)+T1(20)/3]- [T2(18)+T2(19)+T2(20)/3]} is less than or equal to the error 'K'.

3. If one of the above conditions is true than the candidate is selected.

Algorithm 2: *Indexing of fingerprint database*

Input: Input image I_a and set of images *S*.

Output: Set of matched records

Procedure *T_tuple(* I_a ***)***

$M_{ij}^{a} \leftarrow$ calculate minutiae points of I_a.

$[X_c, Y_c] \leftarrow$ calculate corepoint of I_a.

for i=1 to n // where n is the number of minutiae points.

$$d_i^a = \sqrt{\left(X_i^a - X_c\right) + \left(Y_i^a - Y_c\right)}$$

end

$T_a \leftarrow$ sort d_i^a and store first twenty distance.

$T_a(21) = average\left(T_a(1), T_a(2), T_a(3)\right)$.

$T_a(22) = average\left(T_a(18), T_a(19), T_a(20)\right)$.

end *T_tuple* *// T(21) and T(22) is used for indexing.*

Procedure *candidate_selection(* T_a, T_b ***)***

$T_{fa/fb} \leftarrow$ first three distance of T_a or T_b.

$T_{la/lb} \leftarrow$ last three distance of T_a or T_b.

If $abs(average(T_{fa}) - average(T_{fb})) \le K \parallel abs(average(T_{la}) - average(T_{lb})) \le K$

select I_b ;

else

discard I_b ;

end

end *candidate_selection*

Procedure *main*(I_a, S) // *for indexing*

for i=1 to length of S

$T_i = T\_tuple(S_i)$;

index T_{i21} *and* T_{i22};

end // *for input image* I_a

for i=1 to length of T

$c\_sel = candidate\_selection(I_a, T_i)$;

if c_sel *is not empty*

$hFPM(T_a, T_{c\_sel})$ //Heap base fingerprint matching (hFPM).

and

end

end *main*

7.4. Chapter Summary

In this Chapter, a description of the proposed model of fingerprint database is presented. It also shows the proposed indexing technique for fingerprint database using distance feature.