CHAPTER 6

Fingerprint Matching based on MinHeap

6.1 Introduction

A novel technique for fingerprint matching algorithm is proposed using MinHeap. The Crossing number [33] is used to extract minutiae from the fingerprint and the core point [68] is determined. MinHeap is constructed using the distance values after calculating the Euclidean distance of the minutiae points from the core. Then the heap-based Fingerprint Matching comparison starts with nearest distance from core point, compare the second nearest distance from core point and so on. There may be a slight variation of two same fingerprint images, which results mismatch, due to this, hFPM algorithm considers the error tolerance rate ϵ for distance difference for accuracy in the matching. There is no requirement of comparing entire two different images as well as the early detection is possible as the hFPM is the fast matching algorithm which is implemented based on MinHeap. Moreover, the hFPM algorithm considers every possible case for matching two fingerprint images, for instance, size difference of the same fingerprint.

The hFPM algorithm does not suffer from rotation of an image, as it is based on the distance from minutiae to core points. The hFPM algorithm may fail if an image is skewed or deteriorated and we do not consider such cases. The varieties of fingerprint image quality from FVC2002 database such as wet images, normal image and dry images are examined by hFPM

algorithm. The novel contribution of the hFPM algorithm to fingerprint matching is enlisted below:

i. Two different fingerprint images are early detected.
ii. hFPM algorithm does not effect by image size.
iii. Accuracy in fingerprint matching is more in comparing to the existing matching algorithms found in literature.
iv. A fast fingerprint matching technique with time complexity of O(n) where n is the total number of minutiae points (number of nodes) in heap.
v. For comparison of the fingerprint, most of the algorithms found in the literature use ridge alignment, whereas the proposed hFPM algorithm deals with the distance between the core point and the minutiae.
vi. As hFPM does not consider the ridge alignment, the rotation of a fingerprint image with respect to original image does not affect in the comparison process.

6.2 Heap Based Fingerprint Matching

The heap-based Fingerprint Matching (hFPM) algorithm is based on heap tree property which allows rotation of an image. The hFPM algorithm is very simple, yet robust and powerful. Let I_a be a two dimensional fingerprint image with a core point. Let M_{ij}^a be the minutiae point of Image I_a, where i is the number of 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 distance of i^{th} minutiae point from one core point (X_c^a, Y_c^a) and it is calculated using Euclidean distance is shown in equation (6).

$$d_i^a = \sqrt{(X_i^a - X_c^a)^2 + (Y_i^a - Y_c^a)^2} ----(6)$$

(X_i^a, Y_i^a) is the coordination of i^{th} minutiae point and b_i^a is a Boolean value of minutiae i.e. either ridge ending or ridge bifurcation. The H_a is a MinHeap for M_{ij}^a with respect to the distance d_i^a from core point (X_c^a, Y_c^a). Let us consider two MinHeap H_a and H_b and compare the top and delete it after comparison as shown in the Algorithm 1. The hFPM takes a core point of the

images and calculates the distances from minutiae. The hFPM constructs H_a and H_b based on the distance of every minutiae point from the core point (X_c^a, Y_c^a) and (X_c^b, Y_c^b) respectively which is explained in Algorithm 1.

Algorithm 1: *Heap based fingerprint matching algorithm*

Input Data: I_b is a image of fingerprint and H_a is a preprocessed MinHeap of image I_a.

Output Result: Hit values

Get the direction of I_b as I_b^d;

Get the ROI of I_b^d as I_b^r;

Get the core of I_b^r as ;

The minutiae points of I_b^r are calculated and stored in M_{ij}^a;

for *i=1 to n* **do**

$d_i^b = \sqrt{(X_i^b - X_c^b)^2 + (Y_i^b - Y_c^b)^2}$;

end

for *i=1 to n* **do**

for *j=1 to 4* **do**

$M_{ij}^b = < d_{i,1}^b, X_{i,2}^b, Y_{i,3}^b, b_{i,4}^b >$;

end

end

hit = 0;

$H_{a/b}$=BuildMinHeap M_{ij}^b;

while *H_a and H_b not empty* **do**

Extract Minimum from $Root_a$=H_a and $Root_b$=H_b;

If $Root_a \rightarrow d == Root_b \rightarrow d$ *or* $| Root_a \rightarrow d - Root_b \rightarrow d | \leq \epsilon$ **then**

If *Type of minutiae is not same* **then**

delete $Root_a$ and $Root_b$;

heapify $H_{a/b}$;

else

hit = hit+1;

delete $Root_a$ and $Root_b$;

heapify $H_{a/b}$;

end

end

end

After applying the above algorithm, a hit value is determined which gives the number of matching distance feature. A matching score is calculated using these hit values and the matching parameter as given in Equation 7.

$$Score = \frac{hit}{Number\ of\ nodes\ in\ H_a\ or\ H_b\ whichever\ is\ \min imum} \quad ---(7)$$

The main task of the proposed algorithm is the construction of minimum Heap. In this construction, minutiae and core points are detected and the distance between the core point and the minutiae points (red and green circle for ridge ending and bifurcation respectively) are shown in the Fig. 6.1. Using the distances, d, minimum heap is constructed where T1<T2<….. ....<T13<T14 and T1, T2,..., T14 are the tuples.

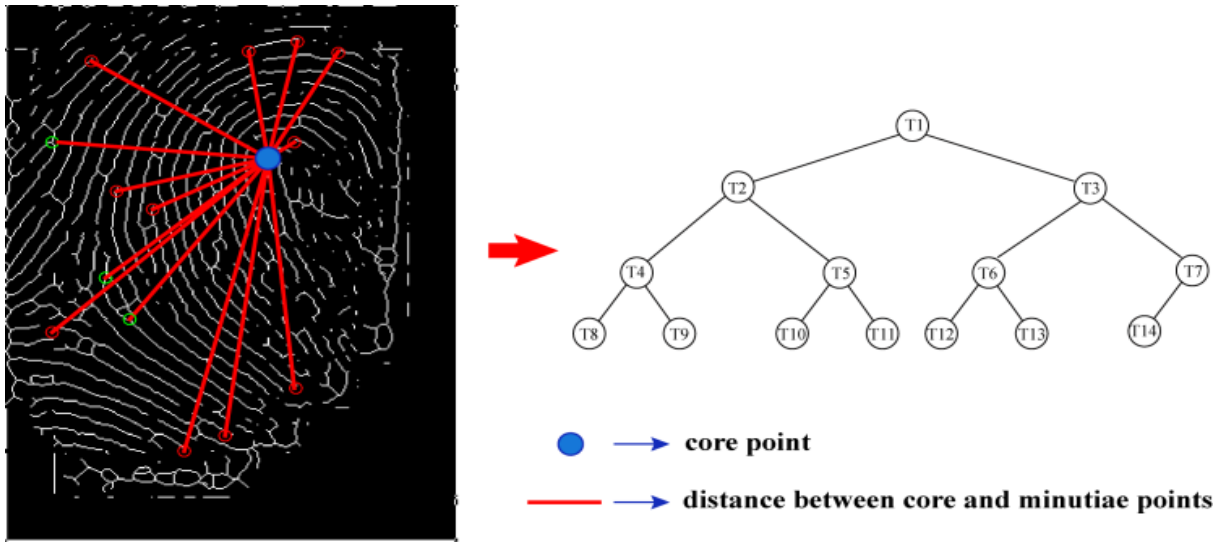

Fig 6.1 Showing Construction of Minimum heap from minutiae points.

6.3. Orientation-Independent

The matching algorithm which the two fingerprint image can match in any degree of rotation is a rotation independent algorithm. The orientation independent fingerprint matching algorithm gives better performance than orientation dependent fingerprint matching algorithm as human finger can have a certain degree of rotation while scanning their finger image on the scanner. The proposed algorithm is an orientation independent algorithm. It is tested in four degrees of rotation viz. -10^o, -5^o, 5^o, 10^o.

6.4. hFPM Test Cases

There are some test cases for hFPM which is significant in proving the robustness of the algorithm.

i. There is a possibility of having same distance between two different images, but the hFPM algorithm takes care of four column, and these are $< d^a_{i,1}, X^a_{i,2}, Y^a_{i,3}, b^a_{i,4} >$. The hFPM assumes that these four columns have at least a difference with two different images.

ii. Same two images can have different distances of same minutiae due to different pressure applied during the acquisition of fingerprint image. For these reason, the hFPM algorithm considers an error tolerance ϵ for matching which is shown in fig. 6.2.

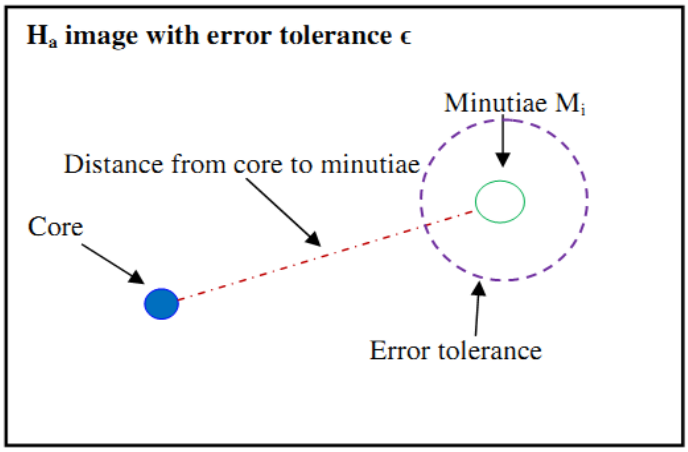

Fig 6.2 Showing error tolerance ϵ

6.4. Chapter Summary

In this Chapter, the heap based fingerprint matching algorithm is proposed. It is an orientation-independent algorithm which is simple but powerful than the orientation based fingerprint matching algorithm as it can match in many possible angles of a fingerprint image.