Indian Journal of Science and Technology, *Vol 10(18), DOI: 10.17485/ijst/2017/v10i18/110325, May 2017*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Fast Fingerprint Indexing Technique with Least Information using Distance Feature

Raju Rajkumar* and K. Hemachandran

Department of Computer Science, Assam University, Silchar - 788011, Assam, India; rajurajkumar.phd@gmail.com, khchandran407@gmail.com

Abstract

Objectives: To develop a novel fingerprint indexing system which is robust against rotation, scaling and noise? **Method/Analysis:** Indexing in large database is a challenging problem as it reduces the number of comparisons. This paper presents a new way of fingerprint database indexing technique using the Euclidian distance from core to minutiae points. Then a set of tuples is created by selecting twenty nearest minutiae points from the core. It performs consistently from image distortion to rotation invariance. It requires less space as it deals only with numerical value and also considers error tolerance "k" as the fingerprint has elastic property. The experimentation is done on FVC 2002 dataset. The high hit rate achieved at low penetration rate indicates that the proposed distance feature indexing technique is satisfactory. **Findings:** The proposed system can able to retrieve fingerprint records with high hit rate at low penetration rate. **Novelity/Improvement:** The proposed system uses the average values of the first three tuples and the last three tuples from the selected set of tuples for indexing.

Keywords: Core Point, Distance Feature, Hit Rate, Minutiae, Penetration Rate

1. Introduction

In the modern digital world, biometrics occupies 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[1–5]. Still, there is a need for fast indexing technique for fingerprint comparison.

Minutiae are the most common feature for fingerprint identification system[6]. Minutiae are the ridge ending and ridge bifurcation. Ridge ending is the point where the ridge terminates and ridge bifurcation is the point where ridge is diverted into two branches as shown in Figure 1. Fingerprint core point is the global feature. The proposed indexing technique feature uses Euclidian distance from core to minutiae.

Figure 2 shows the fingerprint skeleton image from which minutiae and core points are detected and the distance between core point and the minutiae points are calculated for distance feature.

As the fingerprint has elastic property, the distance from core to minutiae may have a slightly different. The proposed indexing algorithm considers error tolerance, K. It is tested in every possible attack such as rotation test, scaling test, noise test and precision test. The dataset FVC2002 is used for fingerprint testing. The novel contribution of this indexing algorithm is listed below:

- A fast and robust candidate selection technique.
- Consistency performance of proposed indexing algorithm from image scaling, rotation and noise attack.

**Author for correspondence*

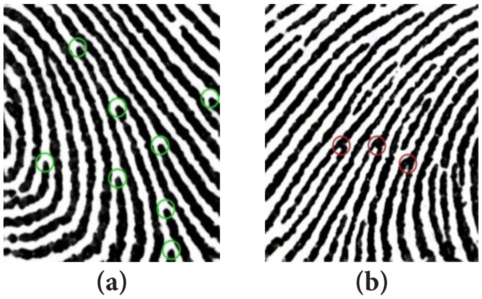


Figure 1. **(a)** Ridge ending **(b)** Ridge bifurcation (in green circle) (in red circle)

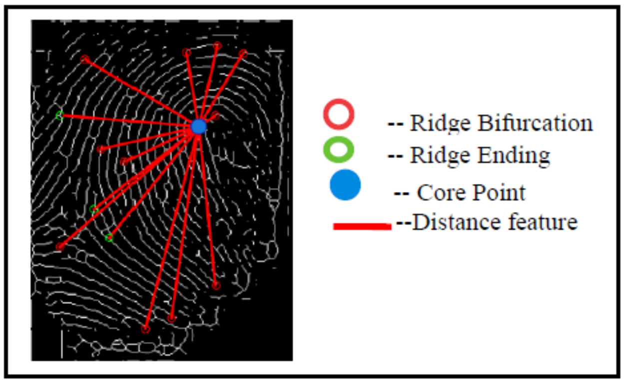


Figure 2. Core point, minutiae point and distance features.

- Least information is required for indexing.
- The paper is organized as follows: Section 2 gives an outline for distance feature, Section 3 explains the proposed indexi algorithm, Section 4 provides experimental results followed by conclusion in Section 5.

In[1] proposed a feature extraction method based on minutia information to create a binary template. By using this binary template and Locality Sensitive Hashing indexing algorithm, a fingerprint indexing is designed. In[2] proposed an indexing method called Local Axial Symmetry (LAS) based on fingerprint registration with a novel feature. After the LAS field is achieved, the location and direction estimation of reference point are achieved in a straightforward way. Then the registered orientation field is utilized as a feature vector to perform the following indexing. In[3] proposed an invariant-based fingerprint indexing scheme. A substructure is formed by combining Minutia and surrounding ridges. The binary relations between substructures are described by the invariants. In[4] proposed continuous fingerprint singular points. By applying a T-shape model to directional field of fingerprint images, location and direction estimation are achieved simultaneously. Homocentric sectors around the candidate singular points are analyzed by the T-shape model, to find the lateral-axes and further main-axes. In[5] proposed a hash-based indexing method to speed up fingerprint identification in large databases. For each minutia, the features defined based on the geometric arrangements of its neighboring minutiae points are computed with its local neighborhood information. The features used are provably invariant to translation, rotation, scale and shear. These features are used to create an arrangement vector (affine invariant local descriptor) for each minutia. In[7] proposed level-2 and level-3 feature based fingerprint indexing algorithm to improve the speed and accuracy of identification using the minutia and pore features, indexing parameters are computed. By incorporating Dempster Shafer theory based match score fusion algorithm, the identification performance is further improved. In[8] proposed an algorithm which used novel features, formed by the Delaunay triangulation of minutiae set as the representation unit, which are insensitive to distortion. These novel features include minutia detail and Delaunay triangle (its handedness, angles, maximum edges, and related angle between orientation field and edges). In[9] proposed a method for fingerprint indexing by using ridge pattern. The ridge pattern within a triangular area is suitable to use as an index and very robust to elastic distortion and is represented by a numerical value. In[10] proposed indexing method which extracts features that have the smallest feature distance to the query fingerprint. Modern systems are able to search databases up to a few hundred fingerprints by using this indexing method. The Three possible fingerprint indexing features viz, Registered directional field estimate, FingerCode and Minutiae triplets, are discussed in this method. In[11] proposed a method for indexing fingerprints named as "minutiae quadruplet". It is used to filter a fingerprint database by combining with a clustering technique. In[12] proposed a fingerprint indexing method based on graph information of minutiae, fingerprint classification and verification which use hierarchical agglomerative clustering technique. This fingerprint indexing is invariant under translation and rotation. In[13] proposed an indexing technique, primarily for latent, that combines multiple level 1 and level 2 features to filter out a large portion of the background database while maintaining the latent matching accuracy.

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 1 is the examples of distance feature extract from the fingerprint image shown in Figure 3 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 1 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.

3. Proposed 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 record. 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.

Table 1. Distance feature from core to minutiae

(i) Distance feature from figure 3(a)

Sl no	Distance
1	29.4108823397055
2	31.9530906173409
3	32.0156211871642
: .	: .
18	68.9637585982667
19	73.3348484691964
20	82.2009732302483

(ii) Distance feature from figure 3(b)

Sl no	Distance
1	28.6530975637888
2	28.6836784620308
3	29.2061637330205
: .	: .
18	75.2927619363243
19	76.5310394545899
20	77.2010362624751

(a) (b)

Figure 3. The two fingerprint images from same finger.

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 ith 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 ith minutiae point from one core point (X_c^a, Y_c^a) and it is calculated using Euclidean distance shown in Equation (1).

$$d_i^a = \sqrt{(X_i^a - X_c^a)^2 + (Y_i^a - Y_c^a)^2} ----(1)$$

(X_i^a, Y_i^a) is the coordination of ith minutiae point and b_i^a is a Boolean value of minutiae i.e., either ridge ending or ridge bifurcation.

A set (first minimum 20) of distance, d_i^a from I_a is shown in Equation (2).

$$T(i) = d_1^a, d_2^a, d_3^a \ldots\ldots d_{20}^a ----(2)$$

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 (3) and (4).

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

$$T(22) = average\big(T(18), T(19), T(20)\big) ----(4)$$

3.2 Candidate Selection

Among these two Equations (3) and (4), 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 considers an error tolerance 'K'. It is used in two conditions:

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

Algorithm: *Indexing of fingerprint database*

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

Output: Set of matched records

Procedure *T_tuple*(I_a)

$M_j^a \leftarrow$ calculate minutiae points of I_a.

$[X_c, Y_c] \leftarrow$ calculatecorepoint 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*

After candidate selection using distance features, fingerprint matching is done based on Heap base fingerprint matching (hFPM)[14].

4. Results and Discussion

For experimentation, the proposed algorithm used fingerprint standard database DB1, DB2, DB3 and DB4 of FVC2002. Each database contains 800 fingerprints from 100 different fingers. The description of the FVC2002 database is given in Table 2.

The proposed algorithm is tested in three possible attacks, namely rotation, noise and scaling:

Table 2. Description of the FVC 2002 fingerprint dataset

Dataset	Scanner type	Image size	Resolution
DB1	Low-cost Optical Sensor	388x374	500 dpi
DB2	Low-cost Capacitive Sensor	296x560	500 dpi
DB3	Capacitive Sensor	300x300	500 dpi
DB4	SFinGe v2.51	288x384	500 dpi

Rotation: It is tested in four degrees, namely -10 degree, -5 degree, 5 degree and 10 degree.

Gaussian noise: It is also tested in four different levels: 0.1, 0.3, 0.5 and 0.7 respectively.

Scaling: For scaling test 1.1, 1.2, 1.3 and 1.4 values were used.

The indexing algorithm also considers error tolerance which is observed by using two parameters i.e., False Matching Rate (FMR) and False Non Matching Rate (FNMR). The intersection point of the two parameters is obtained at the threshold value of 18, which is the error tolerance and is shown in Figure 4. Equation (5) and (6) provides FMR and FNMR.

$$FMR = \frac{Number\ of\ imposter\ fingerprint\ accepted}{Total\ number\ of\ imposter\ test} \quad (5)$$

$$FNMR = \frac{Number\ of\ genuine\ fingerprint\ rejected}{Total\ number\ of\ genuine\ test} \quad (6)$$

4.1 Evaluation

The performance of database indexing can be measured by hit rate and penetration rate. The hit rate represents the correct selection candidate list from the fraction of the set of selected candidates and the penetration rate represents the average length of the candidate list retrieved for each probe.

4.1.1 Experimentation on DB1

The sample fingerprint images of DB1 contain three quality viz, dry images, normal images and dark images as shown in Figure 5. The experimental results are shown in Figure 6.

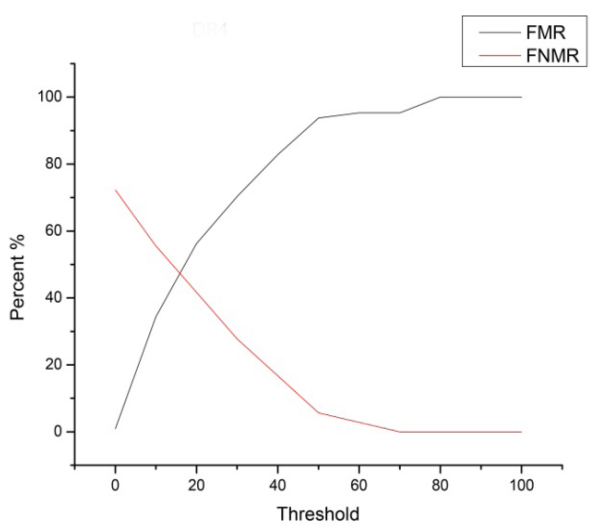


Figure 4. Error tolerance 'K'.

From the Figure 6 it is observed that, in rotation test, the penetration rate at the hit rate of 100% varies from 25% to 45 %. In noise test, the penetration rate at the hit rate of 100% varies from 15% to 43%. In Scaling test, the penetration rate at the hit rate of 100% varies from 20% to 52%.

4.1.2 Experimentation on DB2

The sample fingerprint images of DB2 are shown in Figure 7. The experimental results of rotation test, noise test and scaling test are shown in Figure 8.

From the Figure 8 it is observed that the rotation test of penetration rate is at a hit rate of 100% which varies from 25% to 45%. In noise test, the penetration rate is at the hit rate of 100%, which varies from 35% to 52%. In

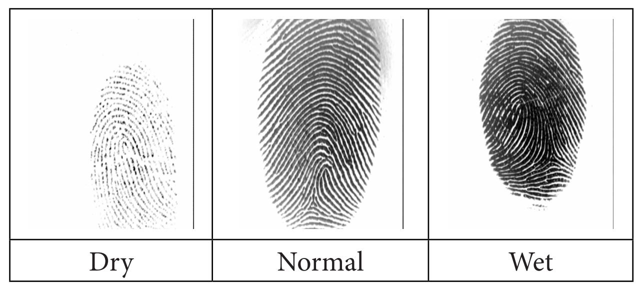
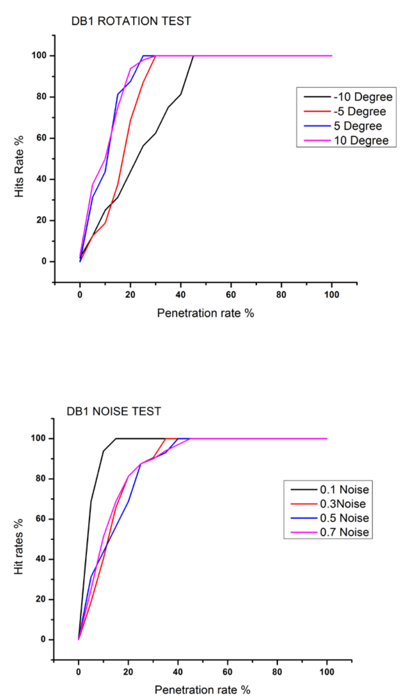


Figure 5. DB1 sample images.



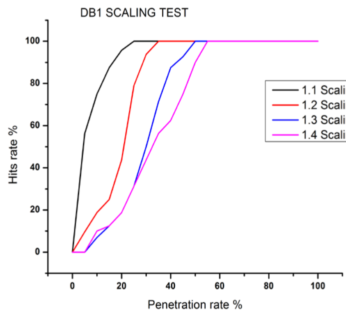


Figure 6. Performance on rotation, noise and scaling test using DB1 FVC 2002 dataset.

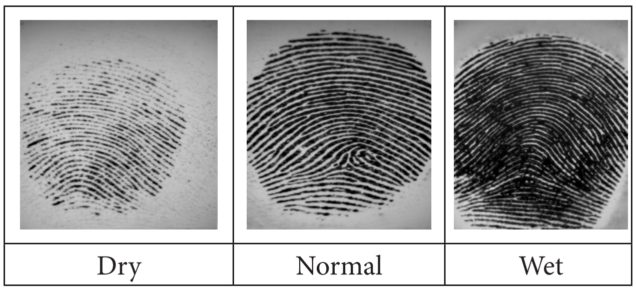
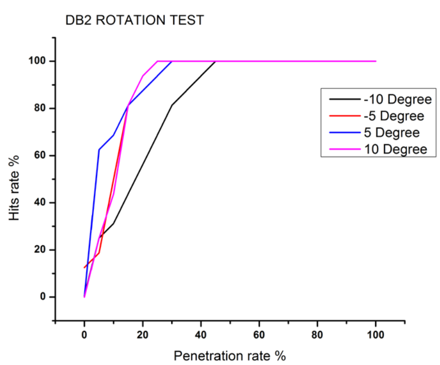
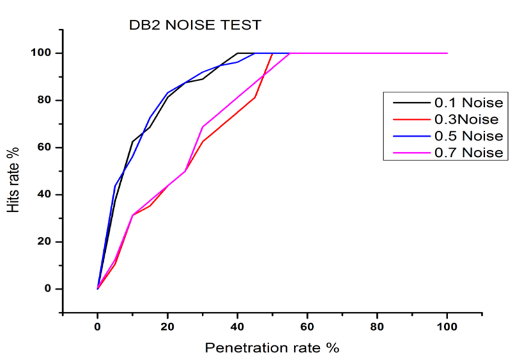


Figure 7. DB2 sample images.





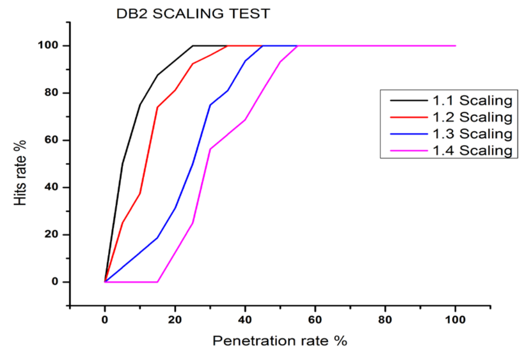


Figure 8. Performance on rotation, noise and scaling test using DB2 FVC 2002 Dataset.

scaling test, the penetration rate is at the hit rate of 100%, which varies from 25% to 50%.

4.1.3 Experimentation on DB3

The sample fingerprint images of DB3 and their different quality is shown in Figure 9. The experimental results of rotation test, noise test and scaling test are shown in Figure 10.

Figure 10 shows the penetration rate on rotation test at a hit rate of 100%, which varies from 25% to 35%. In noise test, the penetration rate is at a hit rate of 100%, which varies from 15% to 30%. In scaling test, the penetration rate is at a hit rate of 100%, which varies from 15% to 30%.

4.1.4 Experimentation on DB4

The sample fingerprint images of DB4 and their different qualities are shown in Figure 11. The experimental results of rotation test, noise test and scaling test are shown in Figure 12.

Here, penetration rate of rotation test at a hit rates of 100% are in the range from 30% to 45%. Penetration rates of noise test and scaling test at a hit rate of 100% are in the range from 20% to 40% and 20% to 38% respectively.

From the experimentation of all the database of FVC2002, it shows that the proposed algorithm has a robustness property to some certain degree of rotation, noise and scaling. And it also performs consistently on different quality of fingerprint images, i.e, dry, dark and normal fingerprint images. The average penetration rate and hit rate for the proposed system using FVC2002 dataset is shown in Figure 13.

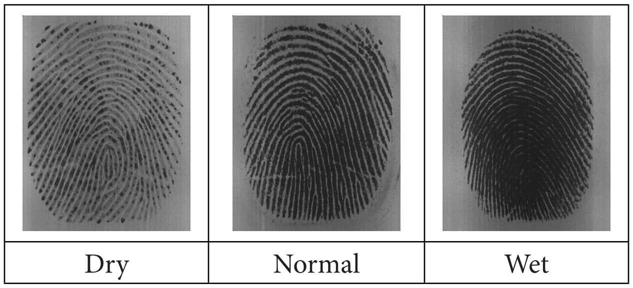


Figure 9. DB3 sample image

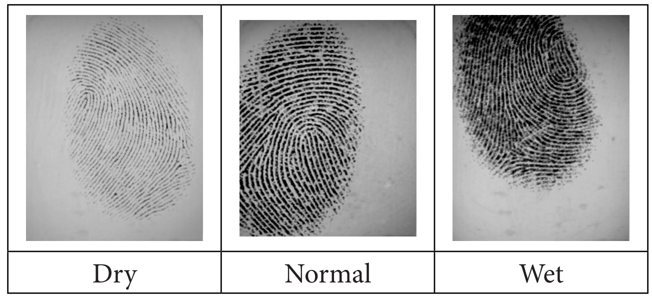


Figure 11. DB4 sample images.

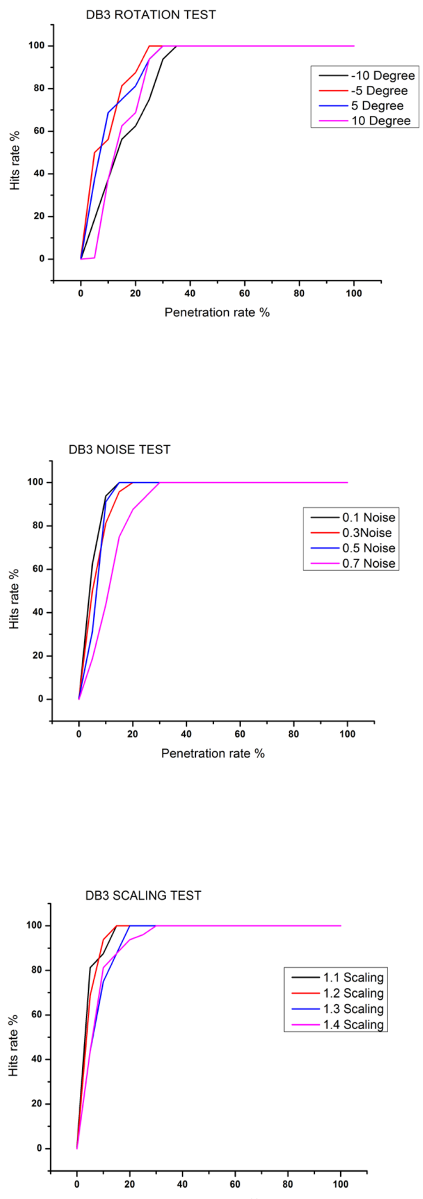


Figure 10. Performance on rotation, noise and scaling test using DB3 FVC 2002 Dataset.

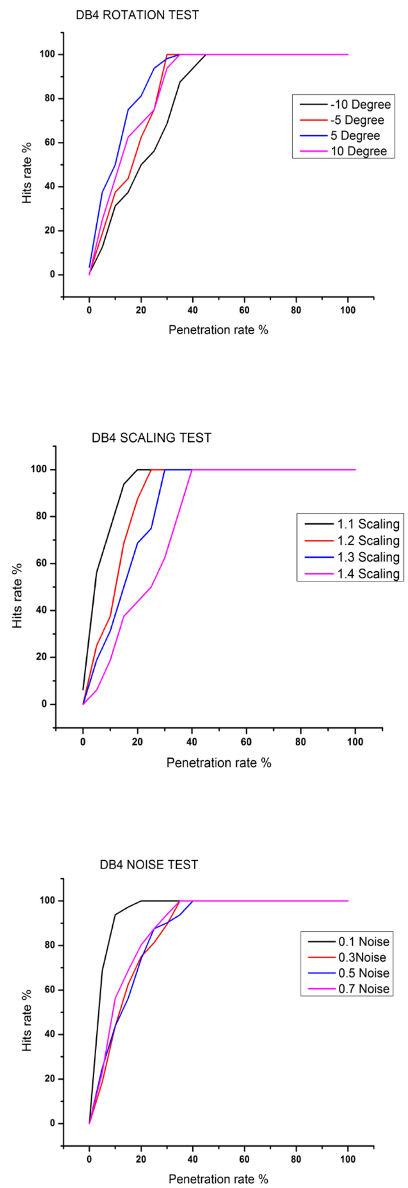


Figure 12. Performance on rotation, noise and scaling test using DB4 FVC 2002 Dataset.

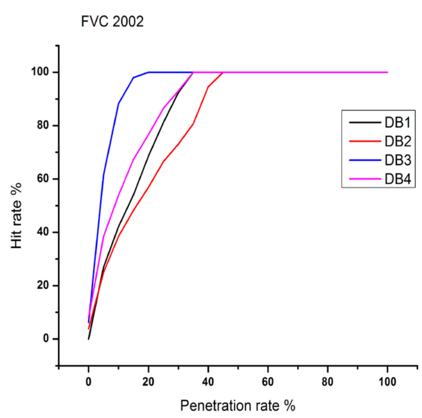


Figure 13. Performance of proposed indexing algorithm on FVC2002 Dataset.

Table 3. Comparison on indexing performance: Penetration rate at 95% and 100% hit rate

Algorithm	FVC 2002 DB1	
	95% hit rate	**100% hit rate**
Minutiae Triplets[15]	**7.2%**	38.1%
Minutiae neighborhood[5]	14%	57%
Distance feature	28%	**35%**

4.2 Comparison

The proposed algorithm is compared with Minutiae triplets[15] and minutiae neighborhood[5]. Table 3 shows the comparison between them with a penetration rate at 95% and 100% hit rate. At 95% hit rate, minutiae triplets perform better than the proposed algorithm, but at 100% hit rate, the proposed algorithm performs better than both the minutiae triplets and minutiae neighborhood algorithms.

5. Conclusion

Distance feature has been used for indexing fingerprint database. It required fewer amounts of space to store the records and fast in candidate selection among the database. It is tested from different kinds of attacks and the results show that the proposed algorithm is rotation invariance, consistency and robustness properties. Based on the experiment conducted from this work, the distance feature can be used as a fingerprint database indexing.

6. References

1. Guoqiang L, Bian Y, Christoph B. A fingerprint indexing scheme with robustness against sample translation and rotation. IEEE International Conference Biometrics Special Interest Group (BIOSIG); 2015. p. 1–8. PMid:26233651.
2. Tong L, Chao Z, Pengwei H. Fingerprint indexing based on LAS registration. IEEE Explorer ICIP; 2006. p. 301–4. PMid:16139283.
3. Jianjiang F, Anni C. Fingerprint indexing using ridge invariants. The 18th International Conference on Pattern Recognition (ICPR'06); IEEE Computer Society. 2006. p. 433–6. Crossref
4. Tong L, Guocai Z, Chao Z, Pengwei H. Fingerprint indexing based on singular point correlation. IEEE Explorer. 2005; 3:293–6. Crossref
5. Akhil V, Anoop N. Fingerprint indexing based on local arrangements of minutiae neighborhood. IEEE Explore. 2012; 71–6.
6. Gowthami AT, Mamatha HR. Fingerprint recognition using zone based linear binary patterns. 2nd International Symposium on Computer Vision and the Internet (VisionNet'15); 2015. p. 552–7. Crossref
7. Singh R, Vatsa M, Noore A. Fingerprint indexing using minutiae and pore features. Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV; Las Vegas. 2009 Jul. p. 13–6.
8. Xuefeng L, Tetsuo A, Arijit B. Distorted fingerprint indexing using minutia detail and delaunay triangle. IEEE Transactions on Information Forensics And Security. 2007 Dec; 2(4):721–33. Crossref
9. Uysal M, Gorgunoglu S. Ridge pattern representation for fingerprint indexing. ElektronikaIr Elektrotechnika. 2014; 20(7):65–8. Crossref
10. Johan DB, Asker MB, Sabih HG. Indexing fingerprint databases based on multiple features. Proc Pro RISC Workshop on Circuits, Systems and Signal Processing; 2001 Nov. p. 1–8.
11. Ogechukwu I, Aglika G, Arun R. Indexing fingerprints using minutiae quadruplets. Appeared in Proc of IEEE Computer Society Workshop on Biometrics at the CVPR Conference; Colorado Springs, USA. 2011 Jun. p. 127–3. PMid:21803300.
12. Lalhmingliana, Bhattacharyya DK, Singh KR. Fingerprint indexing and verification. International Journal of Computer Science Engineering and Information Technology Research. 2013 Jun; 3(2):167–76.
13. Alessandra A, Paulino EL, Kai C, Anil K J. Latent fingerprint indexing: Fusion of level 1 and level 2 features. Appear

in Biometrics: Theory, Applications and Systems (BTAS); 2013. p. 1–8.

14. Rajkumar R, Hemachandran K. A robust orientation-independent fingerprint matching technique using minheap. Contemporary Engineering Science. 2016; 9(26):1283–96. Crossref

15. Bhanu B, Tan XJ. Fingerprint indexing based on novel features of minutiae triplets. IEEE Trans Pattern Anal Mach Intell. 2003 May; 25(5):616–22. Crossref