



Experimental Study on Latent Fingerprint Matching Using Clustered Minutiae Patterns

Uttam U. Deshpande¹ and V. S. Malemath²

¹ Department of Electronics and Communication Engineering,
KLS Gogte Institute of Technology, Belagavi, Karnataka, India
uudeshpande@git.edu

² Department of Computer Science and Engineering,
KLE Dr. M. S. Sheshgiri College of Engineering and Technology,
Belagavi, Karnataka, India
veeru_sm@yahoo.com

Abstract. Use of Biometrics, especially fingerprints is increasing day by day due to its reliable verification and identification. This dominance has resulted in establishment of Automatic Fingerprint Identification System (AFIS). Latent AFIS is in growing demand due to absence of manual human intervention in matching. But latent AFIS is still in its early stages and has to overcome many challenges. Thus, there is a need for developing a fast, reliable and robust latent AFIS. In this paper we have proposed a robust hash-based indexing technique for large latent fingerprint matching to speed up matching. In this system minutia local neighborhood information is extracted to create minutiae features. These obtained minutiae features are invariant to scale, translation, rotation and each minutia is represented as feature vector. This information further can be used to construct hash-table and matching can be performed. In the experiments, Correct Index Power (CIP) is used to measure the retrieval accuracy.

Keywords: Biometrics · AFIS · Latent fingerprint · Hash-based indexing · Feature vector · CIP · Retrieval accuracy

1 Introduction

Automated person identification of a person through automated systems is possible due to emerging biometric technologies. Research carried out in biometric identification includes face, IRIS, palm prints and fingerprints. Among all, finger print identification systems is considered as highly preferred biometric system due to its reliability and accuracy. Main aim of fingerprint identification system is to verify the identity of a person and the challenge arises in fingerprint matching. Fingerprints have achieved great success in past several decades and are widely used in personal verification and identification [1]. The use of fingerprint identification systems avoids the need of remembering passwords and they have been extensively used in applications like attendance systems, banks, building security, law enforcement, commercial, civilian and financial applications [2]. Due to good identification accuracy, fingerprint based

identification systems have put up themselves in a better market place than any other biometric systems.

In fingerprint recognition method [3] or identification method [4], following common steps are performed to build a database of templates:

- 1 Enrollment stage: Templates are formed by extracting features from a captured fingerprint and are stored in a database.
- 2 Verification stage: The template of query fingerprint is formed in the same manner and it is compared with the stored template. Matching decision is made based on the matching criteria.

Captured fingerprint images cannot be directly stored to perform matching. Rather, feature extraction is done before proceeding for matching stage. Feature extraction is a process of identifying important properties of the fingerprint which later can be used in fingerprint matching. Fingerprint features can be made up of three levels [5]. Level-1 features provide macro level information about the singular points, ridge flow, ridge orientation or ridge frequency. These features form the global feature set. Other global features containing the singular points include core point, delta point, and ridge counts. Level-2 features are made up of minutiae points. Minutiae points are the unique fingerprint features obtained from a person's fingerprint. Ridge bifurcations and endings form the local minutiae features. These minutiae points show significant variations from one fingerprint to another. Level-3 features comprises of low level minutiae features like sweat pores, ridge shape, ridge width, dots and curves.

These features are shown in Fig. 1. Left loop, right loop, whorl, arch and tented arch are the other ridge features shown in Fig. 2. Limitation of level-3 features is that it can only be observed in images captured from high resolution cameras.



Fig. 1. Fingerprint image features.

Most of fingerprint identification systems make use level-2 features. This is because fingerprint minutiae are the most unique and reliable features. After feature extraction step, next goal will be to identify a person's fingerprint from stored database. The task of matching step will be to provide match score between two fingerprints based on their likeness. Most of the methods use minutiae based matching, but there are other methods of matching. These are based on image correlations of images, level-2 and level-3 features. Shape, edge and texture features [40] can also be used for matching

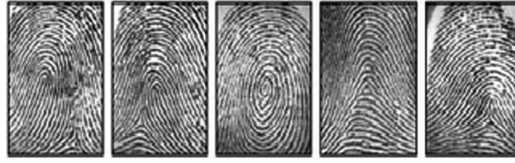


Fig. 2. Basic ridge features.

fingerprints. Pattern matching using minutiae features is being widely used to match the fingerprints from past decade. Results have shown that fingerprint storage is most efficiently used.

Also matching results have been proven in courts [6]. Recently, researchers have used local minutiae structures to align fingerprints and results at a global level are obtained from locally obtained results.

Latent fingerprint identification system is used in criminal investigation and forensic applications. Latent fingerprints are the impressions which are unintentionally left on the objects. The latent fingerprint matching techniques pose more difficult challenges compared to conventional fingerprint matching. This is mainly because minutiae in latent patterns may get missed out due to its nature or may get distorted by noise. Low image quality, poor texture, and non linear distortion in fingerprints create additional challenges in latent fingerprint identification system. Minutiae clustering based pattern matching technique for a latent fingerprint are unexplored. This paper proposes a study on minutiae clustering approach for pattern generation and recognition for latent fingerprint applications. Unique latent minutiae feature vector is defined from minutiae triangular features. A hash value is generated from different arrangements of feature vector. The obtained hash values are used to construct hash-table. Proposed hashing method shows significance improvement in the matching speed and accuracy. The paper is organized as following; Sect. 2 discusses about the related work carried out in conventional fingerprint matching. Section 3 introduces to the proposed pattern matching methods based on connected minutiae clusters. Section 4 describes the experiments carried out on latent database and Sect. 5 concludes the work.

2 Related Work on Fingerprint Matching

This section deals with work carried out by various researchers in the field of fingerprint matching.

2.1 Correlation Based Techniques

In most of the work carried out so far, fingerprint minutiae feature is most popularly used. This is because fingerprint minutiae contain unique and reliable features. In addition, the templates generated from minutiae information consumes less memory and are faster than the graphical based fingerprint matching. In correlation based technique, two fingerprint images are made to overlap on each other and the respective

overlapping pixels at different alignment settings are used to compute the similarity score. This is simple operation but this did not yield acceptable results. This is mainly due to variations that occur in brightness and contrast of the image. Many researchers used different methods to improve this technique.

2.2 Minutiae-Based Features

Minutiae based matching algorithm are mainly designed to solve problems of connected minutiae and similarity score. Most of the indexing techniques make use of minutiae features [7–10] and are derived from its neighborhood relations. These features form geometrical patterns and are invariant to rotations and translations. Minutiae Cylinder Codes (MCC) [11] has been considered as state-of-art technique for minutiae neighborhood representation and is widely used in fingerprint matching or indexing.

2.3 Fingerprint Indexing and Clustering Technique

Fast access of fingerprint templates database for identification is possible using fingerprint indexing. Partially extracted minutiae information can be one indexing technique used to build local structures. This information can be used to calculate fingerprint similarity and to generate key indexes. Ordering of candidate templates done in this way increases the probability of match. Another indexing technique uses triangle characteristics like length, and angle [10]. This type of triangulation technique helped to improve the matching efficiency [12]. Delaunay triangulation technique connects the minutiae in its neighborhood using triangles to identify minutiae orientation and position. Triangular structure obtained using this technique doesn't much vary even if there are deformations in rotations or translations. Thus, minutiae matching using Delaunay triangle technique provide good performance with respect to match time and nonlinear distortions. Similarly, another Delaunay Triangulation [13] approach for fingerprint alignment and matching was proposed. Here global matching was done by comparing using minutiae points and singular points. Fingerprint matching using local features called as Voronoi neighbor structures (VNSs) [1] were used which are robust, rotation and translation invariant.

Several researchers worked on indexing techniques based on clustering and found out significant improvement in the matching accuracy. Fingerprint indexing using dynamic clustering technique [14] extracts translation and rotation invariant features from triangles constructed from triangle spiral from fingerprints. Automatic Fingerprint Identification using clustering technique [15] uses identification module to detect identical minutiae clusters from template images. Here the evaluation of minutiae cluster is done and based on the results the feedback module trains the core vector. Translation and rotation invariant cluster based template generation for fingerprint matching [17] stores fingerprint templates instead of minutia feature vectors. Latent fingerprint clustering using deformable minutiae [16] uses multiple merged and overlapping matching minutia pair clusters.

2.4 Fingerprint Based Pattern Matching Technique

This is a process of matching the individual persons fingerprint features against the features of database templates for person verification [21]. Pattern matching method is used in latent fingerprints for matching a set of primitive latent features with enrolled fingerprint database. Pattern matching is achieved by developing templates using global and local features [22] and matching against stored database template. These features are invariant to global transformation functions. Success of any fingerprint based pattern matching technique depends on absence of fingerprint distortion, noise, arch [23] and identification of core point and minutiae points. Core points and minutiae points are most commonly used as they are invariant to ridge patterns.

The combinations of Euclidean distance and ridge features [20] can be used for minutiae based fingerprint matching. Initially, minutia pairs were obtained using Euclidian distance for representation of transformation invariant features. Here, minutiae features were obtained from ridges and they were grouped together. Finally, histogram based modeling was used for similarity score computation. This method is suitable for AFIS with limited memory space. Fingerprint pattern matching in AFIS using minutiae neighborhood and singular points [18] have produced good matching accuracy. In this method Euclidian distance is used to calculate the distance between singular point and minutiae. This relationship between singular point and minutiae is utilized to calculate pattern matching scores. Pattern recognition and matching algorithm using core point was proposed in [24]. Such technique is suitable for real time AFIS.

A Partial fingerprint recognition system based on local features [25] was proposed to overcome miniaturization problem arising from fingerprint sensors capturing small sensing areas. This is because sensors capture only a part of the fingertip due to small area limitation or improper fingertip placement of the fingertip or scars on fingerprint due to injury [26, 27]. Partial fingerprint recognition system using global features [19] was proposed. Here pattern matching depends on core point, delta point and ridge flow. Fingerprint recognition using Dynamic Time Warping (DTW) [44, 45] was proposed to compress the fingerprint feature matrix into a single vector to save the information.

A detailed survey done on available computational algorithms [43] for fingerprint recognition helps in selecting suitable algorithm for a given fingerprint application.

Latent fingerprints pose more complex challenge compared to normal fingerprints due to presence of background noise and poor quality latent image. This affects the matching speed as the time taken to extract minutiae features from poor quality fingerprint is more. Robust minutiae algorithms [28–30] were proposed to overcome false minutiae detection and matching. Image segmentation technique [31] was proposed to reduce matching speed in AFIS. Segmentation is a process of partitioning fingerprint foreground information from irrelevant background noises. For classifying fingerprint match or no match, an active learning method [41] can be used. A supervised k-NN [42] can be used for classifying the fingerprint data. Here the data which is closest to the cluster center and the farthest from the cluster fingerprint is calculated and classified to reduce classification speed. Latent fingerprint based pattern matching is an unexplored field. This paper proposes a study on latent fingerprint based pattern matching, which is discussed in Sect. 3.

2.5 Recent Trends in Fingerprint Matching

Due to advanced hardware technology, the performance of fingerprint matching used in personal identification is improving. Different fingerprint features are being used in matching to achieve high matching accuracy. These methods are discussed in the next section:

Fingerprint Matching with Hardware Acceleration: Hardware acceleration can speed up the matching process and many researchers are working in this direction. Hardware acceleration using FPGA [32], distributed computing [33] and GPU [34] based architectures are successfully being used for this purpose.

Use of Embedded Systems: Different embedded systems using smart cards [36] and sensors [35] are being used to improve the performance of matching.

Palmprint Matching: This is achieved using minutiae and ridge features. Techniques followed in latent fingerprint matching [37, 38] can be utilized in palmprint matching.

Multi-feature Based Matching: Other human features like face recognition [39] and fingerprint features can be combined to improve matching accuracy.

3 Proposed System for Latent Fingerprint Matching Using Clustered Minutiae Patterns

Latent minutiae fingerprints suffer from low resolution, background noise and non-linear distortions. To address these problems, a minutiae feature based hash index method is adopted. First we begin with minutiae feature extraction. After obtaining minutiae features, a fixed length minutiae descriptor is created to perform matching. This descriptor is based on the neighborhood geometry pattern and is invariant to affine minutiae deformations. Hence, it can be directly applied on minutiae based templates. Another advantage of using this descriptor is that there is no need to detect singular points to align the templates.

3.1 Feature Extraction

Preprocessing of latent fingerprint is done before extracting features. Preprocessing contains steps like binarization and thinning. In binarization, a grey level image is converted into a binary image. This process involves the examination of each grey pixel value and thresholding it to binary '0' or '1' based on the grey level.

Thinning is carried out before proceeding to feature extraction. In this step, each pixels neighborhood from the binarized image may be deleted based on the pixel deletion criteria. It performs a morphological operation in which foreground pixels are removed until they become one pixel wide.

Finally, minutiae extraction is carried out after the preprocessing step. Crossing Number (CN) technique is used on the obtained skeleton image from the previous step. Ridge patterns are 8-connected. Hence, minutiae points are extracted by sliding a 3×3 window on the fingerprint image and ridge pixels are found from its local

neighborhood. Based on the neighborhood connectivity, minutiae features such as ridge ending or bifurcations can be obtained.

Minutiae based indexing schemes provide better results than other indexing schemes. Hence indexing based on Minutia Cylinder Code (MCC) is most widely used and is considered to be state-of-art indexing technique. Major limitation of this method is that it is not invariant to affine transformations. A new fixed length local minutiae descriptor based indexing technique is proposed to overcome the above limitation. A latent minutiae feature vector is defined to capture the minutiae neighborhood information and is discussed next.

3.2 Latent Minutiae Feature Vector

A fixed length local latent minutiae descriptor is proposed to obtain the minutiae distinctiveness from its neighborhood. This representation is used to compare two minutiae points, which in turn helps to determine the fingerprint similarity. The procedure followed to create a latent minutiae feature vector is listed below:

- Consider ‘n’, which defines the total number of latent minutiae feature points (L) around a central latent minutia. Let $n = 7$, then L1, L2, L3, L4, L6, L7, and L8 are the 7 nearest latent minutiae neighbors obtained around L5. This is indicated in Fig. 3(a).

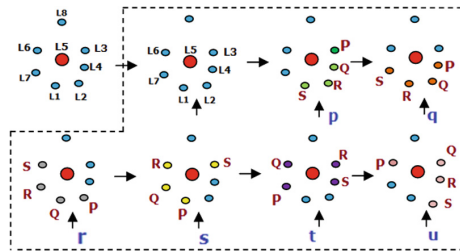


Fig. 3. (a) Nearest ‘n’ latent minutiae around L5 ($n = 7$) (b) ‘m’ invariants p, q, r, s, t, u obtained by rotating P, Q, R, S clockwise around L5.

- Consider ‘m’, which defines the possible local geometric arrangements around L5. These arrangements produce geometric invariant features which will help in the construction of feature vector. This is obtained by choosing four coplanar points P, Q, R, S on latent minutiae and rotating it in clockwise direction around L5. To decrease the probability of similar geometric arrangements, higher value of ‘m’ must be chosen. If the value of $m = 4$ (equal to number of coplanar points) is chosen to produce geometric arrangements, then it will become difficult to discriminate these similar arrangements from one another. Hence, to increase the discrimination power the value of $m = 6$ is chosen. As a result of this, six geometric arrangements namely p, q, r, s, t, u will be formed by rotating the points P, Q, R, S around the six nearest latent minutiae in clockwise cyclic order. Here, L3, L4, L2, L1, L7 and L6 are the six nearest latent minutiae obtained around L5. Since $n = 7$

and to obtain six invariant arrangements, the farthest latent minugia should be discarded. The ‘m’ invariants pqrstu indicates the feature vector shown in Fig. 3(b). This information will be used to generate feature vectors from combination of above arrangements. This is explained later.

3.3 Triangular Features

With P, Q, R, S points, different indexing features for construction of feature vector can be extracted. They are given below:

- The ratio of the triangle areas formed by latent minugia triplets P, Q, R and P, Q, S.
- The ratio of the triangle lengths (larger-side) of latent minugia triplets P, Q, R and P, R, S.
- The ratio of minimum and median triangle angles formed by latent minugia triplets P, Q, R and P, R, S.

Features mentioned above are invariant to translation or rotation. Hence, the latent minugia feature vector can be obtained by considering different combinations of above triangular features.

3.4 Fingerprint Hash-Table

Here we are creating hash table using feature vectors. The same was discussed in the previous section. By rotating P, Q, R, S points clockwise, possible permutations of these feature vectors are: pqrstu, qrstup, rstupq, stupqr, tupqrs, upqrst. Each feature vector ‘fv’ is considered for hashing. The values to be stored in the hash table depend upon the hash value and it is calculated from Eq. 1.

$$HT_{index} = \left(\sum_{i=1}^m f_u[i] \times k^i \right) \bmod HF_{size} \quad (1)$$

Here, ‘HTindex’ is the hash index used in construction of hash-table, ‘fv’ is the feature vector having the length ‘m’, ‘k’ is the level of quantization to be chosen for invariant and ‘HFsize’ is the size of the hash function table. ‘fv’ is the discretized invariant feature vector obtained from weighted combination of triangular features and the quantization value chosen should be in the range of [0, k] to generate unique hash-value in the table.

Algorithm 1 Fingerprint Hash Table Construction Algorithm.

```

Input  $\leftarrow$  Latent Fingerprint image database ldb, n, m, k
Output  $\leftarrow$  Constructed Hash-Table
for each  $lfp \in \{Latent\ Fingerprint\ image\ database\ ldb, n, m, k\}$  do
     $p \leftarrow$  Extract minutiae feature points
    for each  $p \in \{All\ latent\ minutiae\ feature\ points\ in\ lfp\}$  do
         $P_n \leftarrow$  Set of n-nearest neighbors around minutia p
        L  $\leftarrow$  List of all possible combinations of m invariants obtained from  $P_n$ 
        for each  $P_m \in \{All\ possible\ combinations\ of\ m\ invariants\ from\ L\}$  do
             $v \leftarrow$  feature vector
            CP  $\leftarrow$  list of all cyclic permutations of v
        end for
        for each  $v' \in \{Feature\ vector\ in\ the\ list\ CP\}$  do
            HTindex  $\leftarrow$  Hash-value calculate from using eq.1 register fingerprint-item,
            (latent fingerprint-ID of lfp, minutia-ID of p, Feature vector v') using
            HTindex.
        end for
    end for
end for

```

These produce unique hash values which will be used in construction of hash-table. Construction of fingerprint hash-table is explained in Algorithm 1. Figure 4 shows the complete construction of fingerprint hash table. The latent fingerprint-ID, minutia-ID and feature vector are stored in the corresponding hash-table index. All these features are stored in the form of linked lists. Collisions can occur when multiple feature vectors map on to the same index of hash-table. To avoid this chaining technique is employed to resolve this problem. Size of the hash-table is calculated as below:

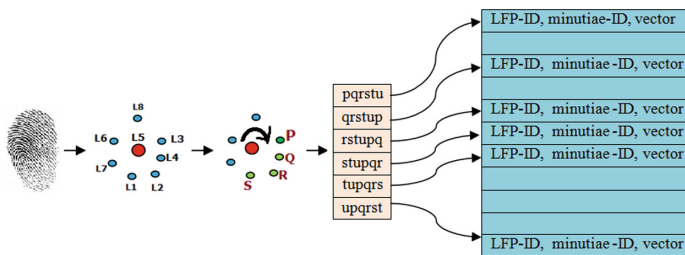


Fig. 4. Construction of fingerprint hash-table.

- $n = 7$, $m = 6$, and $k = 15$.
- Number of latent fingerprint images in the database is 80.
- Average number of feature-points from each latent fingerprint image is 6377.
- The latent fingerprint-ID, minutia-ID and feature vector are stored as 2-bytes, 2-bytes and 1-byte variables respectively in the corresponding hash-table index.

- A pointer to index this hash-table requires 8 bytes.

Size of hash-table, HFsize = $80 \times 6377 \times 7 \times (2 + 2 + 1 \times 15 + 8) = 96.42$ MB of memory.

3.5 Fingerprint Retrieval

Our next task is to perform the matching of query latent fingerprint with the fingerprints in the hash table. For query images, the fingerprint hash-table using feature vectors “v” is constructed in the similar manner as explained in the previous section. Using “v”, latent fingerprint candidate list is retrieved from the hash-table database. Finally, every minutia of query fingerprint casts a vote in its candidate list and based on maximum votes the list of top latent fingerprints is obtained. The fingerprint retrieval procedure is given in Algorithm 2.

Algorithm 2 Fingerprint Retrieval Algorithm.

```

Input  $\leftarrow$  Latent query image img, n, m, k, N
Output  $\leftarrow$  List of sorted top latent fingerprints
 $p' \leftarrow$  Extract minutiae feature points from img.
for each  $p' \in \{ \text{All latent minutiae feature points in img} \}$  do
     $P'_n \leftarrow$  Set of n-nearest neighbors around minutia  $p'$ 
    L  $\leftarrow$  List of all possible combinations of m invariants obtained from  $P'_n$ 
end for
for each  $P'_m \in \{ \text{All possible combinations of m invariants from L} \}$  do
     $v'' \leftarrow$  feature vector for  $P'_m$ 
    HTindex  $\leftarrow$  Hash-value calculate from using eq.1, lookup in Hash Table with
    HTindex and retrieve the corresponding candidate list.
    for each  $\text{fingerprint\_item} \in \{ \text{From retrieved candidate list} \}$  do
        for all item in the retrieved list do
            if  $v'' == \text{fingerprint\_item.featurevector}$  then
                Increase the vote count for latent fingerprint -ID corresponding to
                 $\text{fingerprint\_item}$ 
            end if
        end for
    end for
end for
end for
On the basis of vote counts sort and output all fingerprints in descending order

```

4 Result and Discussions

The results were obtained after conducting an experiment on latent fingerprint FVC_2002_DB4 database. This database contains 80 fingerprint impressions. These fingerprints were used to generate the latent fingerprint hash-table. For experimentation n, m and k values were set at 7, 6 and 15 respectively. The hash function table ‘Hfsize’ is set at 10,00,00,000. In the experiments, latent fingerprint retrieval accuracy is used for analysis. The accuracy is defined in terms of Correct-Index-Power (CIP). It is the

percentage of correctly indexed query latent fingerprint images (Nciq) against the total number of latent fingerprint images stored in the database (Nid) as shown in Eq. 2.

$$\text{Hence, CIP} = (\text{Nciq}/\text{Nid}) \tag{2}$$

A plot in Fig. 5 compares the performance of proposed system on the normal latent fingerprint database FVC_2002_DB4_N against FVC_2002_M (25% Missing minutiae) and FVC_2002_S (25% spurious minutiae). It can be observed that the algorithm produces average 100% retrieval for normal query fingerprints, 99% retrieval accuracy after adding 25% false (spurious minutiae) to the query fingerprints and 76% retrieval accuracy after removing 25% of minutiae (missing minutiae) from the query fingerprints.

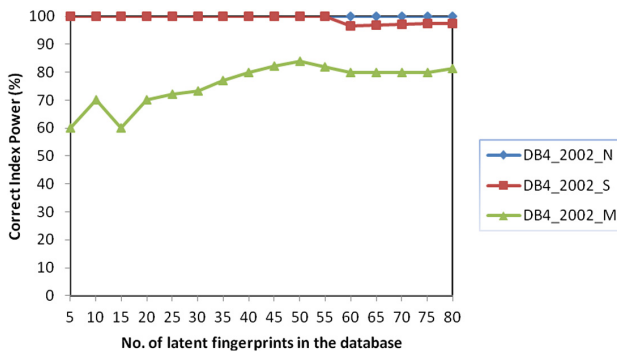


Fig. 5. Comparison of Normal (N) minutiae data vs. 25% Spurious (S) minutiae data vs. 25% Missing (M) minutiae data from FVC_2002_DB4 database.

A plot in Fig. 6 compares the performance of proposed system on the normal latent fingerprint database FVC_2002_DB4_N against FVC_2002_M (50% Missing minutiae) and FVC_2002_S (50% spurious minutiae). It can be observed that the algorithm

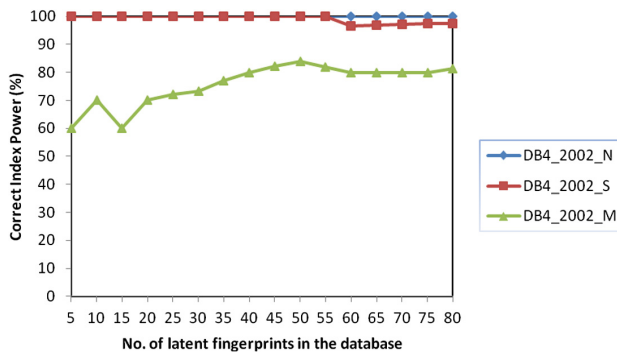


Fig. 6. Comparison of Normal (N) minutiae data vs. 50% Spurious (S) minutiae data vs. 50% Missing (M) minutiae data from FVC_2002_DB4 database.

produces average 100% retrieval for normal query fingerprints, 45% retrieval accuracy after adding 50% spurious minutiae to the query fingerprints and 22% retrieval accuracy after removing 50% of minutiae from the query fingerprints. It can be clearly observed from Figs. 5 and 6 that the proposed system produces better CIP even after adding and removing minutiae from query fingerprints.

5 Conclusion

Latent fingerprint matching using clustered minutiae patterns was proposed to tackle the fingerprint-indexing problem, where fingerprints are searched through large databases consuming lot of time. A literature survey was done on available minutiae based, index based and pattern recognition based methods. A new rotation and translation invariant method called feature vector was proposed based on neighborhood minutiae relation, which can be directly used on latent fingerprint templates. Feature vector uses triangular features like angles, length ratios, ratios of areas which are invariant to translation, rotation and scaling. A latent fingerprint hashing algorithm based on feature vector was proposed to build hash-table. For fast and accurate latent fingerprint retrieval, a fingerprint retrieval algorithm was proposed. The results were obtained by conducting experiments on FVC_2002_DB4 latent database. 25% of minutiae from query fingerprints were added and removed randomly in order to check the robustness of the proposed algorithm against minutia-appearance. Results obtained showed that the retrieval algorithm is robust against the database containing missing or spurious minutiae fingerprints (noisy and low quality images). Also the algorithm effectively dealt with latent fingerprint images containing non-linear distortions. Thus the proposed algorithms are ideal in latent fingerprint matching. However, the performance of the algorithm got degraded when more than 50% noise was introduced. This was done by adding and removing 50% of minutiae from query fingerprint images randomly. More number of false minutiae appeared in the close vicinity of original neighbors in the process of generating spurious minutiae and many minutiae points were removed while creating fingerprint database containing missing minutiae. A new algorithm can overcome this problem.

References

1. Jain, A.K., Ross, A.A., Nandakumar, K.: Introduction to Biometrics. Springer, New York (2011). <https://doi.org/10.1007/978-0-387-77326-1>
2. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of Fingerprint Recognition. Springer, New York (2009). <https://doi.org/10.1007/978-1-84882-254-2>
3. Jain, A.K., Hong, L., Bolle, R.M.: On-line fingerprint verification. In: Proceedings of IEEE Transaction Pattern Analysis Machine Intelligence, pp. 302–314 (1997)
4. Jiang, X., Liu, M., Kot, A.C.: Fingerprint retrieval for identification. IEEE Trans. Inf. Forensic Secur. **1**(4), 532–542 (2006)
5. Hasan, H., Kareem, S.A.: Fingerprint image enhancement and recognition algorithms - a survey. Neural Comput. Appl. **23**(6), 1605–1610 (2013)

6. Jain, A.K., Feng, J., Nandakumar, K.: Fingerprint matching, pp. 36–44. IEEE Computer Society (2010)
7. Germain, R., Califano, A., Colville, S.: Fingerprint matching using transformation parameter clustering. *IEEE Comput. Sci. Eng.* **4**(4), 42–49 (1997)
8. Iloanusi, O., Ross, A., Gyaourova, A.: Indexing fingerprints using minutiae quadruplets. In: Proceedings of IEEE Computer Society Workshop on Biometrics at the Computer Vision and Pattern Recognition (CVPR) Conference, pp. 127–133 (2011)
9. Boer, J., Bazen, A., Gerez, S.: Indexing fingerprint databases based on multiple features. In: Proceedings of ANN Workshop Circuits, Systems and Signal Proceedings, pp. 300–306 (2001)
10. Bhanu, B., Tan, X.: Fingerprint indexing based on novel features of minutiae triplets. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(5), 616–622 (2003)
11. Cappelli, R., Ferrara, M., Maltoni, D.: Fingerprint indexing based on minutia cylinder-code. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(5), 1051–1057 (2011)
12. Liang, X., Bishnu, A., Asano, T.: A robust fingerprint indexing scheme using minutia neighborhood structure and low-order delaunay triangles. *IEEE Trans. Inf. Forensic Secur.* **2**(4), 721–733 (2007)
13. Hawthorne, M.R.: Fingerprint analysis and understanding. CRC Press, Taylor and Francis Group, Boca Raton, London (2009)
14. Wang, C., Gavrilova, M.L.: Delaunay triangulation algorithm for fingerprint matching. In: 3rd International Symposium on Voronoi Diagrams in Science and Engineering, Banff, pp. 208–216 (2006)
15. Wencheng, Y., Hu, J., Song, W., Milos, S.: An alignment free finger print bio cryptosystem based on modified Voronoi neighbor structures. *Pattern Recogn.* **47**(3), 1309–1320 (2014)
16. Jain, A., Prasad, M.V.N.K.: A novel fingerprint indexing scheme using dynamic clustering. *J. Reliable Intell. Environ.* **2**(3), 159–171 (2016)
17. Qun, R., Jie, T., Yuliang, H., Jiangang, C.: Automatic fingerprint identification using cluster algorithm. In: ICS (2002)
18. Iwasokun, G., Charles, A.O., Officer, A.C.: Fingerprint matching using neighbourhood distinctiveness. *Int. J. Comput. Appl.* **66**, 1–8 (2013)
19. Jea, T.-Y., Venu, G.: A minutia-based partial fingerprint recognition system. *Pattern Recogn.* **38**(10), 1672–1684 (2005)
20. Iwasokun, G.B.: Fingerprint matching using minutiae-singular points network. *Int. J. Signal Process. Image Process. Pattern Recogn.* **8**(2), 375–388 (2015)
21. Iwasokun, G.B.: Development of a hybrid platform for pattern recognition and matching of thumbprints. *J. Comput. Sci. Appl.* **17**(2) (2010)
22. Zhang, Y., Jiao, Y., Li, J., Niu, X.: A fingerprint enhancement algorithm using a federated filter. In: Information Counter Measure Technique Institute. Harbin Institute of Technology (2008)
23. Lopez, A.C., Ricardo, R.L., Queeman, R.C.: Fingerprint pattern recognition. Ph.D. thesis, Electrical Engineering Department, Polytechnic University (2002)
24. Zhang, W., Wang, S., Wang, Y.: Pattern recognition and matching algorithm of fingerprint based on core point. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science, Beijing (2004)
25. Iwasokun, G.B., Akinyokun, O.C., Olabode, O.: A Block processing approach to fingerprint ridge orientation estimation. *J. Comput. Technol. Appl.* **3**, 401–407 (2012)
26. Akinyokun, O.C., Adegbeyeni E.O.: Scientific evaluation of the process of scanning and forensic analysis of thumb prints on ballot papers. In: Proceedings of Academy of Legal, Ethical and Regulatory Issue, vol. 13 (2009)

27. Jea, T.-Y., Govindaraju, V.: A minutia-based partial fingerprint recognition system. *Pattern Recogn.* **38**(10), 1672–1684 (2006)
28. Espinosa-Duro, V.: Minutiae detection algorithm for fingerprint recognition. In: *Proceedings IEEE 35th Annual 2001 International Carnahan Conference on Security Technology* (Cat. No. 01CH37186), London, England, UK, pp. 264–266 (2001)
29. Chikkerur, S., Wu, C., Govindaraju, V.: A systematic approach for feature extraction in fingerprint pattern recognition. Center for Unified Biometrics and Sensors (CUBS), University at Buffalo, New York (2004)
30. Iwasokun, G.B., Akinyokun, O.C., Alese, B.K., Olabode, O.: Fingerprint image enhancement: segmentation to thinning. *Int. J. Adv. Comput. Sci. Appl. (IJACSA)* **3**(1) (2012)
31. Ali, S.M., Al-Zewary, M.S.: A new fast automatic technique for fingerprints recognition and identification. *J. Islamic Acad. Sci.* **10**, 55–60 (1997)
32. Jiang, R.M., Crookes, D.: FPGA-based minutia matching for biometric fingerprint image database retrieval. *J. Real-Time Image Process.* **3**(3), 177–182 (2008)
33. Peralta, D., Triguero, I., Sanchez-Reillo, R., Herrera, F., Benitez, J.M.: Fast fingerprint identification for large databases. *Pattern Recogn.* **47**, 588–602 (2014)
34. Gutierrez, P.D., Lastra, M., Herrera, F., Benitez, J.M.: A high performance fingerprint matching system for large databases based on GPU. *IEEE Trans. Inform. Forensic Secur.* **9**, 62–71 (2014)
35. Bayram, S., Sencar, H.T., Memon, N.: Efficient sensor fingerprint matching through fingerprint binarization. *IEEE Trans. Inf. Forensic Secur.* **7**, 1404–1413 (2012)
36. Benhammedi, F., Bey, K.B.: Embedded fingerprint matching on smart card. *Int. J. Pattern Recogn. Artif. Intell.* **27**(02), 1350006 (2013)
37. Jain, A.K., Feng, J.: Latent palmprint matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**, 1032–1047 (2009)
38. Liu, E., Jain, A.K., Tian, J.: A coarse to fine minutiae-based latent palmprint matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 2307–2322 (2013)
39. Hong, L., Jain, A.: Integrating faces and fingerprints for personal identification. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**, 1295–1307 (1998)
40. Santosh, K.C., Antani, S.: Automated chest x-ray screening: can lung region symmetry help detect pulmonary abnormalities. *IEEE Trans. Med. Imaging* **37**(5), 1168–1177 (2018)
41. Bouguelia, M.R., Nowaczyk, S., Santosh, K.C., et al.: Agreeing to disagree: active learning with noisy labels without crowd sourcing. *Int. J. Mach. Learn. Cybern.* **9**(8), 1307–1319 (2018)
42. Szilárd, V., Santosh, K.C.: A fast k-nearest neighbor classifier using unsupervised clustering. In: Santosh, K.C., Mallikarjun, H., Vitoantonio, B., Atul, N. (eds.) *RTIP2R 2016*. CCIS, vol. 709, pp. 185–193. Springer, Singapore (2017). https://doi.org/10.1007/978-981-10-4859-3_17
43. Bhanu, B., Tan, X.: *Computational Algorithms for Fingerprint Recognition*. Springer, New York (2004). <https://doi.org/10.1007/978-1-4615-0491-7>
44. Santosh, K.C., Lamiroy, B., Wendling, L.: DTW-radon-based shape descriptor for pattern recognition. *Trans. Pattern Anal. Mach. Intell.* **27**, 1350008 (2013)
45. Santosh, K.C., Bart, L., L., Wendling: DTW for matching radon features: a pattern recognition and retrieval method. In: Jacques, Blanc-Talon, Richard, K., Wilfried, P., Dan, P., Paul, S. (eds.) *ACIVS 2011*. LNCS, vol. 6915, pp. 249–260. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23687-7_23