



Latent Fingerprint Identification System Based on a Local Combination of Minutiae Feature Points

Uttam U. Deshpande¹ · V. S. Malemath² · Shivanand M. Patil² · Sushma. V. Chaugule²

Received: 23 March 2020 / Accepted: 23 March 2021 / Published online: 10 April 2021
© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd 2021

Abstract

Latent fingerprints are the fingerprint traces obtained from various surfaces of objects found at the crime scene. Latent fingerprint quality is, therefore, poor and they suffer from non-linear distortions. This is why latent fingerprints contain partial or incomplete minutia information. Developing an automatic fingerprint identification system (AFIS) for latent fingerprints becomes a major challenge because of a small number of minutia features. Moreover, currently developed AFIS for latent fingerprints are not robust against geometric transformations. In this article, based on the existing partial minutiae characteristics, we propose a method for detecting latent fingerprints. To identify an individual with existing nearest neighbor minutia structures, we develop a scale and rotation invariant algorithm called “Ratio of Minutiae Triangles (RMT)”. The algorithm utilizes the features which are based on the local minutiae arrangements around a reference minutia. To deal with missing minutiae in latent fingerprints, we consider minutiae clusters based on nearest minutiae neighborhood relation to form hash structures. In the fingerprint retrieval process, these hash structures are used. On the FVC2004 (Maio et al., FVC2004: third fingerprint verification competition. In: International Conference on Biometric Authentication. ICBA, Hong Kong, pp 1–7, 2004) and NIST SD27 (Garris and McCabe, NIST Special Database 27: Fingerprint Minutiae from Latent and Matching Tenprint Images, Technical Report. National Institute of Standards and Technology, Gaithersburg, 2000) databases, experiments are carried out with different rotation and scale settings. For the FVC2004 and NIST SD27 datasets, we achieved Rank-1 identification accuracy of 78.75% and 86.82%, respectively. The result demonstrates substantial changes in the matching results relative to state-of-the-art algorithms, and the proposed method is robust against geometric transformations.

Keywords Automatic latent fingerprint identification system · Ratio of minutiae triangles · Minutiae clusters · Nearest minutiae neighborhood · FVC2004 · NIST SD27

Introduction

For the past, many years fingerprints have been used as one of the most reliable biometric used for identifying a person [1]. This is because fingerprint features remain unchanged throughout their lifetime. Fingerprints have been the most preferred biometric over other biometric traits [2]. They have found themselves in a wide range of applications ranging from forensics [3], secure mobile payments [4], and maintaining ID of the citizens at borders [5]. The size of these databases can exceed several million in number and the size further can increase if different fingerprint impressions of the same person being registered. Performing one-to-one fingerprint verification over such a large-scale database poses different challenges. There is also a need to build a fingerprint indexing-based automated fingerprint identification system (AFIS) to scale down the search size. In the

✉ Uttam U. Deshpande
uttamudeshpande@gmail.com

V. S. Malemath
veeru_sm@yahoo.com

Shivanand M. Patil
shpatil102@gmail.com

Sushma. V. Chaugule
vidhu_sun@yahoo.co.in

¹ Department of Electronics and Communication Engineering, KLS Gogte Institute of Technology, Belagavi, Karnataka, India

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

above-mentioned applications, AFIS has been widely used. Using rolled/plain and latent fingerprints, latent fingerprint matching using AFIS is achieved. Rolling fingerprints (Fig. 1a) are obtained by dipping the surface of fingerprints in ink and rolling them on a paper surface from one side to another. This approach is used to capture complete ridge data and the amount of minutiae collected from this type of fingerprint would be about 100. Whereas, plain fingerprints (Fig. 1b) are obtained by pressing the inked fingerprint against the paper or pressing the fingerprint over a fingerprint-scanning device. The capture area of plain fingerprints is smaller compared to the rolled fingerprint and the number of minutiae obtained is around 50 in number.

The rolled/ten-print fingerprint matching using AFIS [6] has achieved matching accuracy of around 100% with a background database of 10 K fingerprints. On the other side, latent fingerprints (Fig. 1c) are collected from the surface of objects that are unintentionally left on crime scenes and they become extremely important evidence in crime scene investigations to convict criminals. The latent fingerprint investigations using the AFIS have the highest social significance and any false convictions will have a great social impact. There have been many instances where wrong convictions have brought embarrassment to investigating agencies and many agencies had to make public apologies.

The murder case of Marion Ross dates back more than 18 years and is one of the controversial cases encountered by the Scottish government [7]. Despite Ms. McKie denied her entry into the home of a murder victim, she was wrongly accused of the murder based on fingerprints found at the crime scene. After a detailed public inquiry, it was concluded as the suspicion was caused due to “human error” and it raised several questions on the integrity of the fingerprint-matching system. Ms. McKie was compensated with £750,000 and the Scottish Police agency apologized for the wrong accusation.

In another case where a terrorist attack took place on March 11, 2004, in Madrid and it witnessed the death of 191

people. An American lawyer named “Brandon Mayfield” [8] was detained for 14 days based on the latent fingerprints found on a bag that contained detonators and explosives from the crime scene. In police custody, Brandon claimed that he did not have a passport and had not gone out of the USA for 10 years. Unable to prove the source of fingerprints by forensic agencies and after a few months of arguments, the court acquitted him. The error made by AFIS resulted in the Federal Bureau of Investigation (FBI) to seek a public apology.

Currently, latent fingerprint identification is carried out manually by forensic experts and the development of a fully automatic accurate latent fingerprint identification system is still in its early stage. The researchers have to address the challenges faced in processing low-quality latent fingerprints. Some of the challenges are explained next.

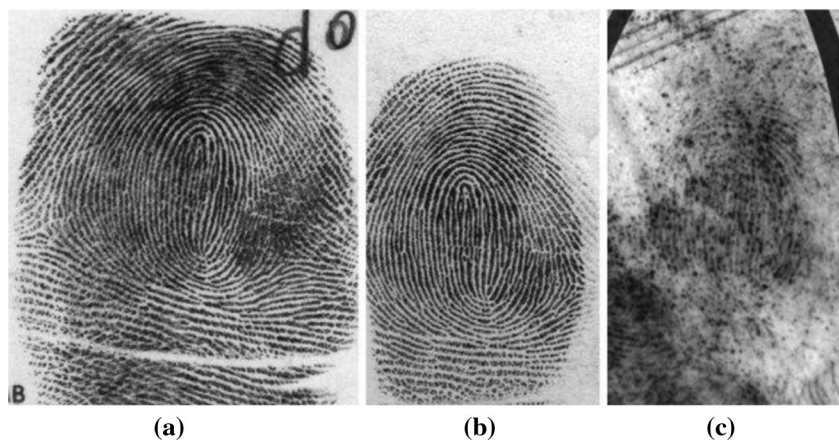
Latent Fingerprint Identification Challenges: Science vs. Fiction

Forensic investigations became popular after airing several crime investigation programs including Crime Scene Investigation (CSI) [9]. These programs showcased forensic laboratories equipped with highly trained personnel working with state-of-the-art instrumentation and solving cases in a timely fashion. Some of the illusions created by these television series are listed below:

- Prosecutors, judges, and police officers believe that CSI will have unreasonable quality and quantity of physical evidence at crime scenes.
- CSI can be easily proved in courtrooms.

The things look completely different when a CSI was carried out. One of the important challenges faced to research latent fingerprint identification is the lack of publicly available latent fingerprint databases. The CSI programs have influenced viewers to research this area, and the collection

Fig. 1 Fingerprint impressions collected using different methods. **a** Rolled, **b** plain, and **c** latent



or distribution of physical evidence in the public domain is increasing.

Other than this, the researchers have to deal with the incomplete and partial latent fingerprints. The latent fingerprints contain partial fingerprint information due to smudged ridges [10] and they result in blurred images. Overall, the quality of fingerprints is poor and they suffer from non-linear distortions due to pressure variations on the subject from where the fingerprints are obtained. Researchers [11, 12] used ridge-frequency, and ridge-orientation to represent the global ridge patterns. The major limitations of these global features are that they are not good at handling non-linear distortions and are not invariant to geometric transformations. Another problem with using global features is that it requires prior information of core points to align fingerprints and in latent fingerprints, core points may not always be present. Hence, global features are better suited for image classification.

Related Work Carried Out

Many researchers [13, 14] working in the field of AFIS used manually marked local minutia points, core point, delta point, and Region of Interest (ROI) to perform fingerprint search. On the NIST SD27 database, the manually annotated fingerprints with orientation field and accuracy maps [13] produced the highest Rank-1 identification accuracy of 79.5%. The dictionary of patches [14] helped to boost the quality of fingerprints, and the highest Rank-1 matching accuracy of 32.51% was produced on manually marked minute points of the enhanced fingerprint. Based on the top Rank-50 matching accuracy, the researchers concluded that the match performance using manually marked points produced better results compared to automatic markers. A fusion of manually marked minutia points with minutiae extracted from automatic minutiae extraction [15] produced Rank-1 identification accuracy of 40% and it was observed that the results were found to be better than methods developed using global fingerprint features. Fingerprint core point extraction and localization using a Gaussian filter [16] method was proposed to predict core points. The major limitation of the above-proposed method is that the manually marked minutiae points are not accurate [17] and the core points are not always available in latent fingerprints. These features are not tolerant of geometric transformations.

Many researchers [18–21] used geometric features that are robust against rotation and translation. These geometric features were derived from its neighboring minutiae points and the hashing technique was used to speed up the search. For performing latent fingerprint identification, an algorithm called the “Descriptor-based Hough Transform (DBHT)” [22–24] was proposed. This algorithm uses Minutiae

Cylinder Codes (MCC) [25] to perform fingerprint alignment using the Hough Transform and later local fingerprint matching is performed. This method has been proved as an effective way of representing a minutia for fingerprint matching and indexing. MCC representation requires prior knowledge about fingerprint alignment and the minutiae neighborhood representation is not robust towards rotation and scale. The system recorded 53.5% [24] of Rank-1 identification accuracy.

Clustering algorithms for latent fingerprints were developed [26]. To cope with non-linear distortions, they used minutiae descriptors and the minutiae clusters were combined in this step to locate the corresponding minutiae. The descriptors of the minutiae were rotated between 0 and $\pi/4$ to perform matching. To address the problems of non-linear distortions, different algorithms were developed. The neighbouring minutiae-based descriptor (NMD) algorithm [26] produced Rank-1 identification accuracy of 85.66%. In a fixed pixel radius, Minutiae neighborhood groups [27] were created and the fingerprint alignment based on best-matching pairs was subsequently performed. To cope with fingerprint deformations, rotations, and scale, a robust algorithm to locate the orientation and location of minutiae using Delaunay triangles [28] was proposed.

To estimate ridge-flow, extract minutiae descriptors with two minutiae templates plus texture template, convolutional neural-networks (ConvNets) are proposed [29]. Reference templates stored in the background are compared with query latent fingerprint and the highest Rank-1 recognition accuracy of 64.7% was published on the NIST SD27 database. A “Latent Fingerprint Identification” (LFI) algorithm [30] with supervised classifier was proposed. Fingerprint matching with global features, deformable minutiae descriptors independent of local minutiae cylinder code (MCC) and m-triplet descriptors was carried out. The highest Rank-1 identification accuracy of 73.26% was achieved with rank level fusion.

An end-to-end solution for fingerprint matching [31] was introduced which utilizes automated ROI cropping, pre-processing, extraction of features, and matching. For each minutia, 96-dimensional descriptors were extracted from the neighbourhood of minutiae, and the virtual minutia was used to quantify the product to reduce the length to 16. 70% rank-1 identification accuracy was produced with rank level fusion.

To summarize, the latent fingerprints contain a limited number of minutia points. The fingerprints are of poor quality and they suffer from fingerprint deformations and non-linear distortions. The latent fingerprints contain poor quality ridges and they suffer from fingerprint deformations and non-linear distortions. The approaches proposed by various researchers make use of core points for fingerprint alignment. Owing to the absence of minutia feature points, the

latent fingerprint may not always contain a core point or delta point. The existing systems are not robust against fingerprint geometric transformations. Hence, there is a need to develop a system that identifies a person with an existing small number of minutia features and without requiring fingerprint pre-alignment. In addition, the system should be robust against fingerprint deformations and scale. A nearest minutiae neighbourhood-based latent fingerprint identification algorithm called “Ratio of Minutiae Triangles (RMT)” is proposed to solve these problems. The main goal of the proposed work is to develop an automated algorithm for fingerprint identification that is more robust for fingerprint deformation, rotation, scale, and does not rely on any alignment of fingerprints. We test our proposed work on FVC 2004_DB1 [35], and NIST SD27 [36] latent fingerprint database. The performance of the algorithms is discussed in the later sections.

The paper is organized as follows. The next section introduces the ratio of minutiae triangles (RMT) algorithm which was developed on nearest minutiae pairs and it makes uses of Latent fingerprint descriptors. The next following section analyses the performance of the proposed algorithms on databases with different settings and compares the performance with the state-of-art algorithms. The last section discusses the results and future scope.

Ratio of Minutiae Triangles (RMT)

We propose a minutia descriptor-based matching system called “Ratio of Minutiae Triangles (RMT)”. RMT uses local minutiae triangular features that are invariant to minutiae deformations, rotations, and scale [18]. The proposed RMT algorithm does not depend upon fingerprint pre-alignment information required for matching. Thus, eliminating the need for fingerprint alignment using a singular point and core point. After the minutiae feature extraction phase, the

feature vectors for the gallery fingerprints and query fingerprints are developed. For matching, the feature vectors of query fingerprints are compared to feature vectors of gallery fingerprints. Feature extraction and RMT algorithms are explained next.

Dealing with Distortions and Geometric Transformations

The triangular features obtained from triangle angles, lengths, and sides are proven to be invariant to translation, rotation, and scale [18]. The RMT algorithms use the minutiae triangle features to produce latent minutiae feature vectors. Latent fingerprints suffer from distortions, and it results in minutia points being shifted away compared to the minutia points in its corresponding ground truth (true mate) fingerprints.

Dealing with Distortions

The minutiae distance measure is invariant under translation, rotation, and fairly invariant to the scale. The ratio of minutiae distance defines angle, and the ratio of distance is invariant under the above-mentioned transformations. The latent fingerprints are most of the times affected by non-linear distortions, the minutia points extracted from query fingerprint look slightly different than its true mate stored in the background gallery database. From Fig. 2a, it can be observed that ‘O’ represents the vertex of the minutiae triangle at reference (0, 0). A, B minutia points are shifted away from their original positions when the fingerprint suffers from distortions. This results in minutia points x_1, x_2, y_2 getting distributed within maximum minutiae distance (L) of 150 pixels such that $0 < x_1 < L, 0 < y_2 < L$ and $x_2 < L$ (Table 1).

Several experiments were conducted to test the robustness of the system against distortions [18]. The researchers

Fig. 2 Triangle variables

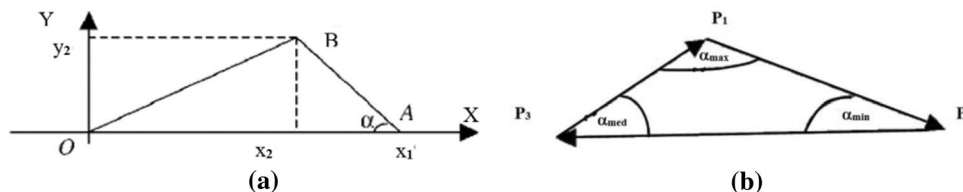


Table 1 Tolerable angle ($\alpha_{min}, \alpha_{med}, \alpha_{max}$) variations under different threshold values

Angle's type	Angle change thresholds					
	$\pm 1^\circ$	$\pm 2^\circ$	$\pm 3^\circ$	$\pm 4^\circ$	$\pm 5^\circ$	$\pm 6^\circ$
α_{min}	51.6	93.2	98.5	99.6	99.9	100.0
α_{med}	56.6	87.3	94.5	97.3	98.7	99.4
α_{max}	1.0	67.7	87.3	94.2	97.2	98.7

observed that the min (α_{\min}), median (α_{med}) minutiae triangle angles can tolerate distortions better than the maximum minutiae triangle angle (α_{\max}) thresholds. The minutiae deviations (α) between 2° and 4° can tolerate most uncertain distortions and can help in reducing the search time.

Let minutia points $P_1, P_2,$ and P_3 represent the minutiae triangular variables, and α_i represent the three angles of the triangle where $i = 1, 2, 3$. The triangle angles are given by

$$(\alpha_{\max}) = \max\{\alpha_i\}, (\alpha_{\min}) = \min\{\alpha_i\} \text{ and } (\alpha_{\text{med}}) = 180^\circ - \alpha_{\max} - \alpha_{\min}$$

From Fig. 2b it can be observed that minutia point ‘P₁’ represents the vertex with an angle α_{\max} , ‘P₂’ represents the minutia vertex point with an angle α_{\min} and ‘P₃’ is the vertex minutiae point with α_{med} angle. Hence, a minutiae triangle formed using minimum (α_{\min}) and median angle (α_{med}) provide robustness towards variations in triangle angle and is given by

$$0^\circ < \alpha_{\min} \leq 60^\circ \text{ and } \alpha_{\min} \leq \alpha_{\text{med}} < 90^\circ.$$

These triangle angles are used to build minutiae invariant feature vectors in our proposed algorithms.

Dealing with Geometric Transformations

We use the following geometric features to identify minutia points that are influenced under different rotation and scale.

- **Relative translation:** Let $Z_i = x_i + j y_i$ be the complex number with $j = \sqrt{-1}$ corresponding to the minutia location point $P_i(x_i, y_i)$, where $i = 1, 2, 3$.

Then, we define, $Z_{21} = Z_2 - Z_1, Z_{32} = Z_3 - Z_2,$ and $Z_{13} = Z_1 - Z_3$.

Further, the centroid of the triangle ‘Z_C’ is defined as follows:

$$Z_C = (Z_1 + Z_2 + Z_3)/3 \text{ and } |Z - Z'| < \delta_r,$$

where Z and Z' are the centroids of the triangle (Z_C) found in two different impressions and $|Z - Z'|$ is the translation between the centroids of these two triangles.

- **Relative rotation:** Let $\theta_{13} = \angle Z_{13}, \theta_{21} = \angle Z_{21}, \theta_{32} = \angle Z_{32},$ and $\angle Z$ is the phase angle.

$$\text{Then, } |\theta - \theta'| < \delta_r,$$

$$\text{Then, } |\theta - \theta'| < \delta_r,$$

where θ and θ' are the minutiae with phase angles $\theta_{13}, \theta_{21},$ and $\theta_{32},$ found in two different impressions.

These features offer robustness towards false correspondences between true minutia points and their corresponding minutia points found in query fingerprint.

Ratio of Triangles

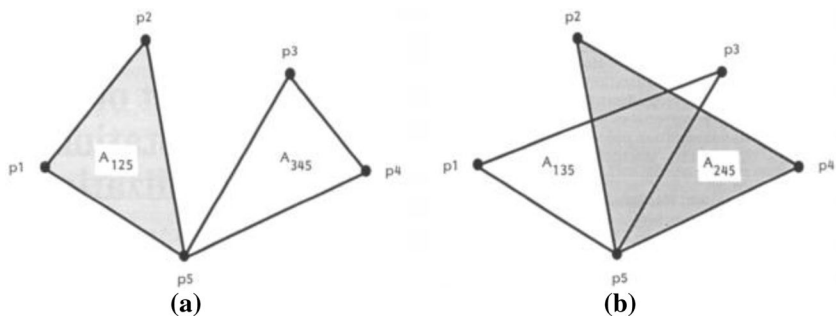
Another set of experiments was conducted by Dunn et al. [24] to reconstruct dental radiographic images captured with different image arrangements. Using projective invariants, the algorithm locates the pixel in the rotated image. Initially, in the reference image, p1, p2, p3, and p4 feature points are chosen and their corresponding P1, P2, P3, and P4 points are identified in the rotated image. Then a fifth point, p5, is scanned through the reference image. The values for two ratios of triangles can be determined using these five-point positions in the reference image as follows:

$$x = \frac{\Delta(1, 2, 5) \cdot \Delta(3, 4, 5)}{\Delta(1, 3, 5) \cdot \Delta(2, 4, 5)}, \tag{1}$$

$$y = \frac{\Delta(1, 4, 5) \cdot \Delta(2, 3, 4)}{\Delta(2, 4, 5) \cdot \Delta(1, 3, 4)}, \tag{2}$$

where the total of eight triangular areas is defined as shown in Fig. 3. The images were rotated in terms of 0, 2, 4, 8, and 16° angles against the reference image after performing initial alignment, and the algorithm produced fewer standard deviations for the difference image. The algorithm transformation performance is determined by the initial selection of feature points. To overcome this limitation, we propose an alignment-free fingerprint-matching algorithm after the minutiae feature extraction step which is explained in “Latent Fingerprint Registration”.

Fig. 3 Representations of the ratio of triangles from eight triangles. **a** Ratio of triangle representation (x) obtained from areas of triangles $\Delta(1, 2, 5), \Delta(3, 4, 5), \Delta(1, 3, 5), \Delta(2, 4, 5)$ **b** Ratio of triangle representation (y) obtained from areas of triangles $\Delta(1, 4, 5), \Delta(2, 3, 4), \Delta(2, 4, 5), \Delta(1, 3, 4)$



Latent Minutiae Feature Extraction

Latent fingerprints contain incomplete partial fingerprint information. Accurate minutiae detection plays a very important role in obtaining good matching results and intern helps to correctly identify a person. We used NIST’s FpMV minutiae viewer [37] open-source minutiae detection tools to obtain reliable minutiae features to achieve better matching accuracy. The software is developed above the MINDTCT application and the extraction results are found to be close to the manually marked minutiae. The block diagrams seen in Figs. 4 and 5 explain the steps used to extract minutiae from the FVC2004 and NIST SD27 databases using FpMV, respectively. Since the quality of FVC2004 query fingerprints contains good-quality ridges, the minutiae features are extracted using the FpMV tool without performing any additional pre-processing operation. Minutiae features are extracted in the form of x, y coordinates and are later read by the RMT algorithm for further matching operation.

Compared to FVC2004 fingerprints, the accuracy of latent NIST SD27 fingerprints is very low. Hence, the fingerprints need to be pre-processed to remove background noise and bring out important foreground ridge information. We use the deep neural network model called “Pre-ProcessNet” [33] to perform automatic segmentation (ROI cropping), and

enhancement operations. Pre-ProcessNet generates orientation maps from the learned dictionary of orientation patches. Similar to orientation maps, the segmentation map is generated to suppress background noise. The orientation information with segmentation information is used to enhance the fingerprint by passing through the bank of Gabor filters as shown in Fig. 5. The fingerprint skeleton is extracted from the enhanced image and binarization operation is carried out before extracting minutiae features. Finally, the minutiae features are extracted using the FpMV tool, and x, y minutiae coordinate information is passed onto the RMT algorithm to perform fingerprint-matching operation.

Figure 6 indicates the complete implementation details of the proposed pre-processing, feature extraction, and matching steps. The RMT algorithm is explained next.

Ratio of Minutiae Triangles (RMT) Algorithm

We propose an RMT algorithm, which uses a distinctive fixed-length latent minutia descriptor defined from nearest minutiae neighborhood invariants. The process begins by first choosing ‘ m ’ minutiae points among ‘ n ’ nearest neighbors, and later calculating the minutiae invariants. The process is explained in detail next.

Fig. 4 Minutiae feature extraction steps for FVC2004 database

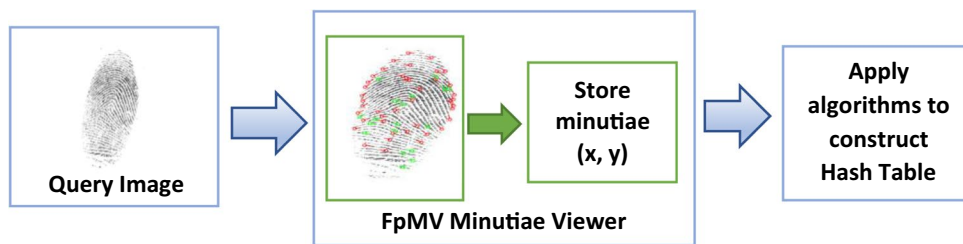


Fig. 5 Minutiae feature extraction steps for NIST SD27 database

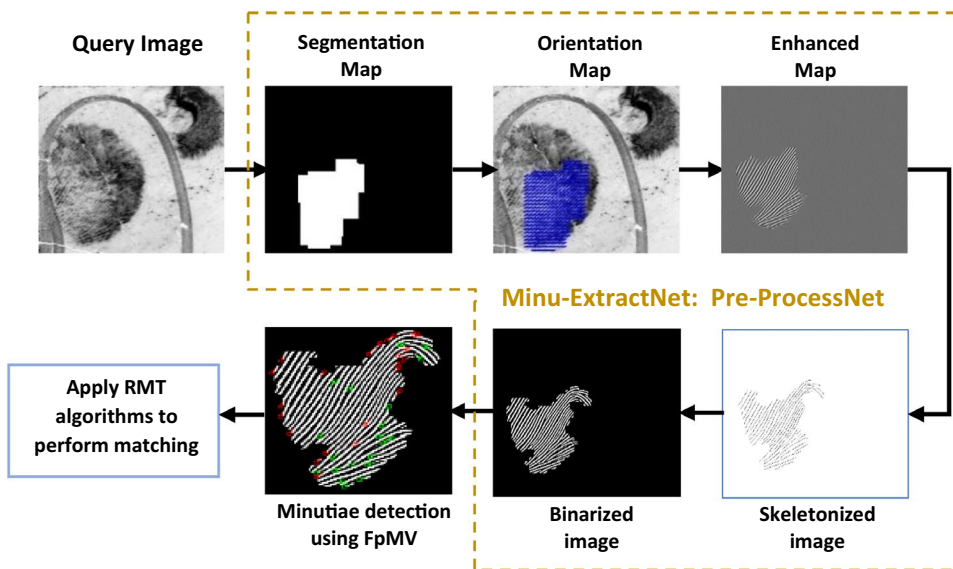
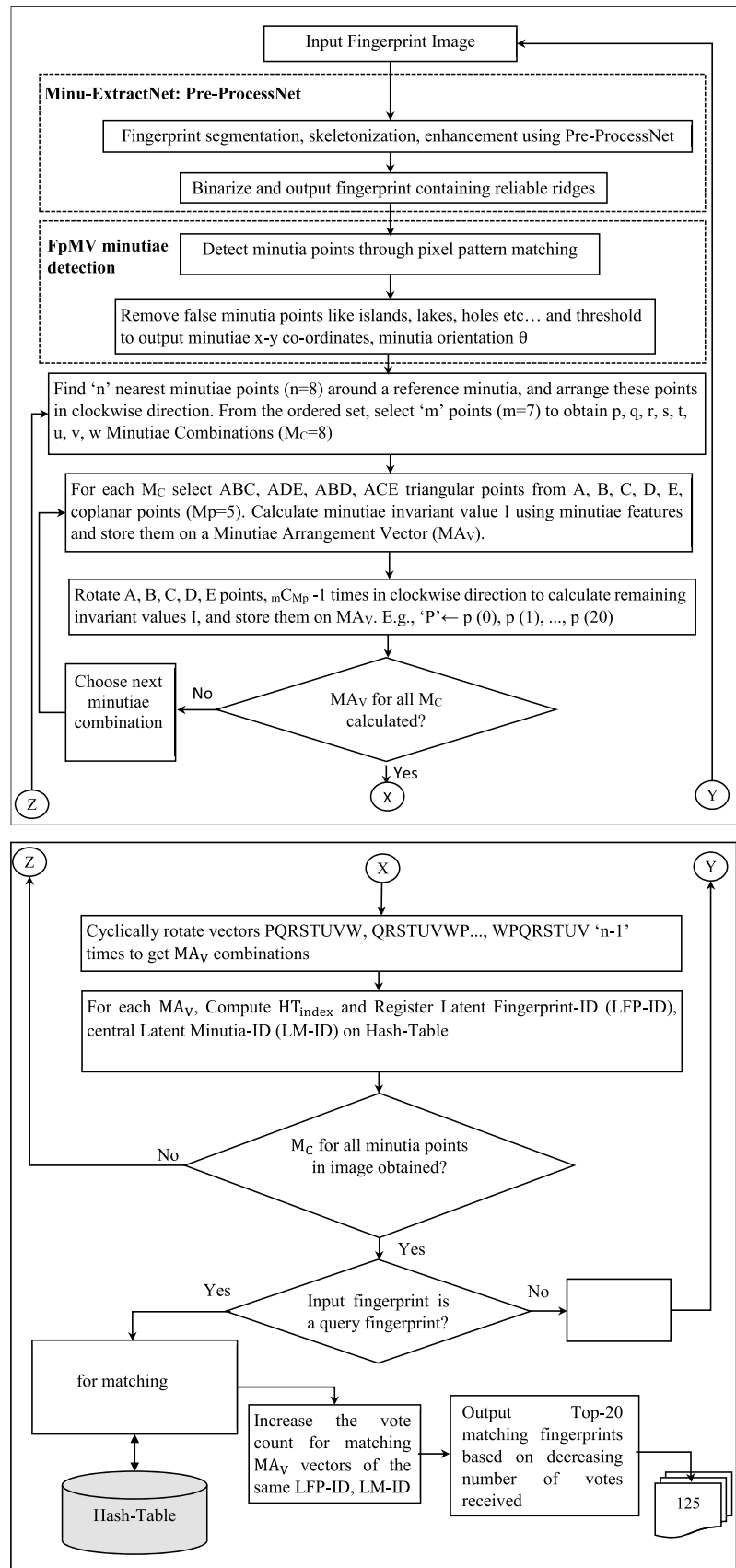


Fig. 6 A complete flow diagram indicating proposed fingerprint pre-processing, extraction, and matching steps



Latent Fingerprint Registration

We start the fingerprint registration process on the gallery fingerprints stored in the background database. From the minutiae points extracted using the previous FpMV application, we select a reference minutia point. To generate minutiae invariants from the set of minutiae neighborhood points, we select ‘n’ minutia points ($n=8$) around a reference minutia ‘P’ and arrange them in clockwise sequence P1, P2, P3, P4, P6, P7, and P8 as shown in Fig. 7. We select ‘m’ minutiae points ($m=7$) from ‘n’ points such that $m < n$ to obtain a set of minutiae combinations (M_C). With $n=8, m=7$ we obtain $\binom{n}{m} = nCm = 8$ minutiae combinations namely p, q, r, s, t, u, v, and w, as seen in Fig. 7. For each M_C , select A, B, C, D, and E coplanar minutia points ($M_p=5$) and calculate minutiae invariant ‘I’ using the ratio of minutiae triangular products using the following equation and store them on a minutiae arrangement vector (MA_V):

$$I = \frac{\Delta(A, B, C) \cdot \Delta(A, D, E)}{\Delta(A, B, D) \cdot \Delta(A, C, E)}, \tag{3}$$

where (A, B, C), (A, D, E), (A, B, D) and (A, C, E) are the area of triangles formed from points A, B, C, D, E. To calculate the remaining invariant values, rotate points A, B, C,

D, and E in the clockwise direction and store them on MA_V as shown in Fig. 8.

The total number of MA_V for a M_C is equal to ${}_mC_{M_p} = 21$. E.g., P ← p (0), p (1), p (2) ..., p (20) invariants are stored on MA_V ‘P’. The steps are repeated to calculate minutiae invariants for p, q, r, s, t, u, v, and w minutiae combinations (M_C) and are stored as P, Q, R, S, T, U, V, and W ‘MA_V’, respectively. These MA_V are arranged as PQRSTUWV, QRSTUWVP, RSTUWVWPQ ... WPQRSTUWV vectors on the hash-table as shown in Fig. 9.

To perform hashing, each MA_V is used to calculate the hash value and is stored on the hash-table using the following equation:

$$HTindex = \left(\sum_{x=0}^{mC5-1} MA_V[x] \cdot k^x \right) \bmod HTsize, \tag{4}$$

where MA_V[x]=MA_V containing $mC5$ invariants obtained from Eq. 3.

M is the number of minutiae points selected from n points ($m=7$). k is the level of quantization ($k=15$). HT_{size} = Hash-table size (kept as 1 TB).

Continuous values contained by MA_V is converted into discrete values by quantizing ($k=15$) between [0, 15] range. On the unique HTindex value, a Latent Fingerprint-ID

Fig. 7 ‘n’ nearest minutiae points ($n=8$) around a reference minutia ‘P’. Clockwise ordering of ‘m’ minutiae points ($m=7$) over ‘n’ points to obtain p, q, r, s, t, u, v, and w minutiae combinations (M_C)

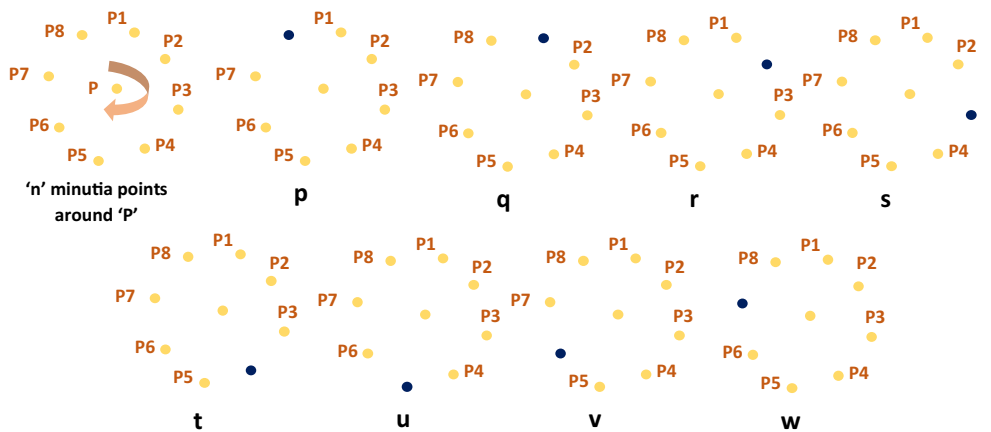


Fig. 8 Minutiae invariants. **a** Arrangement of ‘n’ nearest minutiae points ($n=8$) in a clockwise direction. **b** Possible combinations ‘m’ minutiae points ($m=7$) around ‘n’ points. **c** A possible minutiae invariant with 5 points chosen around ‘m’

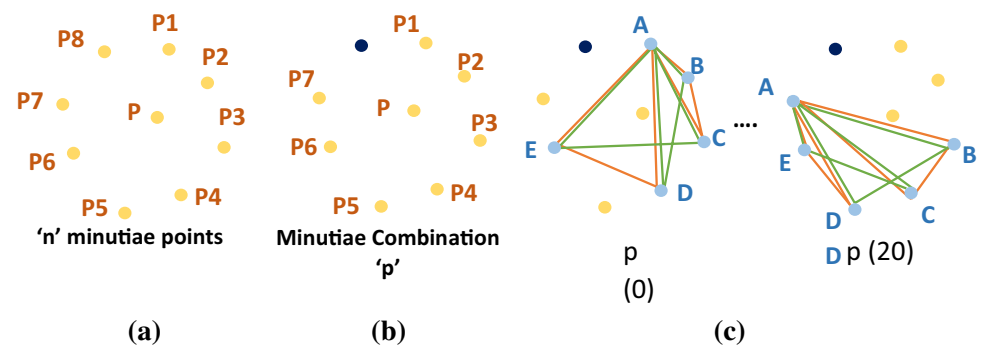
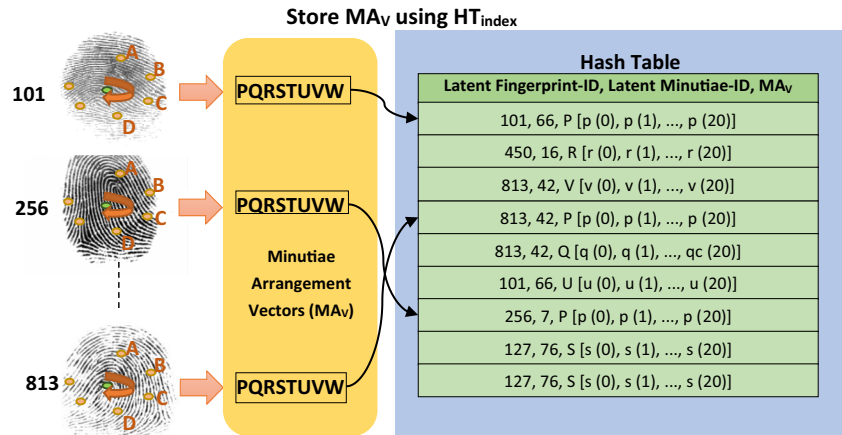


Fig. 9 Process of hash-table construction during fingerprint registration



(LFP-ID), a Latent Minutia-ID (LM-ID) and MA_V are stored into the corresponding cell of the hash-table. These steps are repeated for the remaining minutia points present in the fingerprint, and a hash-table is constructed. The RMT fingerprint registration algorithm is explained next.

values stored on hash-table and the candidate list (LFP-ID, LM-ID, and MA_V) is retrieved. Vote counts for the number of matching feature vectors of LM-ID belonging to the corresponding LFP-ID are increased. The voting process will be ineffective for the repeated matching of the same LM-ID

Algorithm 1: RMT Fingerprint Registration Algorithm.

```

Input ← Latent fingerprint image (fp) in Database (ldb), n, m, k.
Output ← Registered fingerprints in Hash-Table.
for each fp ∈ {Latent fingerprint image belonging to database ldb}
  p ← Read minutiae feature points (x, y) from fp
  for each p ∈ {All minutiae feature points of image fp}
    Pn ← Clockwise arranged ‘n’ nearest neighbour minutiae points around p
    for each Pm ∈ {‘m’ minutia points selected from Pn} do
      MC ← Calculate minutiae combination
      for each MC ∈ {All possible minutiae combinations (MC) of minutiae point Pm} do
        MAV ← Calculate invariant feature vectors I using eq.3.
      end for
      MAV ← Cyclically arrange MAV
      HTindex ← Hash-Table index calculated from using eq.4.
      Register ← {Latent fingerprint-ID (LFP-ID), Latent Minutia-ID (LM-ID), r (0), r (1)
        ..., r (mCs-1) Feature vectors belonging to p using HTindex}.
    end for
  end for
end for
  
```

Latent Fingerprint Retrieval

Similar to the hash-index (HTindex) values calculated from minutiae invariant feature vectors calculated for gallery fingerprints, the HTindex for query fingerprints are calculated. The query HTindex values are compared with the HTindex

and LFP-ID. The final matching result is obtained by sorting fingerprint ID’s (LFP-ID) in descending order of votes received. The fingerprint with the highest votes represents Rank-1 matching accuracy. The experiment steps and the results are explained in the next section.

Algorithm 2: RMT Fingerprint Retrieval Algorithm.

Input \leftarrow Extracted minutiae points of query latent fingerprint image (qfp), n , m , k .
Output \leftarrow List of top retrieved latent fingerprints.
 $p' \leftarrow$ Read minutiae feature points (x , y) from qfp
for each $p' \in \{\text{All minutiae feature points of image } qfp\}$
 $P'_n \leftarrow$ Clockwise arranged 'n' nearest neighbor minutiae points around p'
for each $P'_m \in \{\text{'m' minutia points selected from } P'_n\}$ **do**
 $M_C \leftarrow$ Calculate minutiae combination
for each $M_C \in \{\text{All possible minutiae combinations } (M_C) \text{ of minutiae point } P'_m\}$ **do**
 $MA_V \leftarrow$ Calculate invariant feature vectors I using eq.3.
end for
 $MA_V \leftarrow$ Cyclically arrange MA_V
QHTindex \leftarrow Query Hash-Table index calculated from using eq.4.
QRegister $\leftarrow \{\text{Latent fingerprint-ID (LFP-ID), Latent minutia-ID (LM-ID), qr (0), qr (1) \dots, qr}_{(mC_{5..1})} \text{ Feature vectors belonging to } p' \text{ using HTindex}\}.$
end for
end for
Candidate List \leftarrow Look up the QHTindex and obtain a matching list from HTindex
for each $qr(i) \in \{\text{feature-vectors of query latent fingerprint image } p'\}$ **do**
for all $qr(i) == r(i)$ of registered latent fingerprint image p **do**
Increase the vote of corresponding latent fingerprint-ID (if it's the first incident of voting for LF-ID with minutia p and it's the first incident of voting for LM-ID for LFP-ID)
end for
end for
Sort the LF-ID in descending order based on votes and output the fingerprint list.

RMT Fingerprint Registration and Retrieval Computational Complexity

Let us consider 'M' as the average number of minutia points present in a fingerprint. Let 'N' represent the number of gallery fingerprints used for registration, and ' M_p ' represent the number of points used to determine the minutiae invariant features.

- Initially 'm' points over 'n' are arbitrarily chosen and all possible minutiae combinations of 'm' points over 'n' $\binom{n}{m}$ are calculated. This step increases the computational cost by 'n' times.
- Next, ' M_p ' points are chosen from 'm' minutia points belonging to a particular minutiae combination, and $\binom{m}{M_p}$ minutiae invariants are obtained.
- This step is repeated for the 'n' minutiae combinations. Total $\binom{m}{M_p} n$ minutiae invariants for all 'n' minutiae combinations are obtained.
- This step is repeated by selecting other minutia points 'M' in a fingerprint. Total $\binom{m}{M_p} nM$ minutiae invariants for all minutia points in a fingerprint are obtained.

- The computational cost for registering all 'N' fingerprints on a hash-table is $O\left(\binom{m}{M_p} nMN\right)$. This is because $\binom{m}{M_p} nM$, minutiae invariants are calculated for every fingerprint belonging to a fingerprint database containing 'N' fingerprints.
- Similarly, the computational cost of retrieving a matching fingerprint is $O\left(\binom{m}{M_p} nM\right)$. This is because $\binom{m}{M_p} n$ minutiae invariants for all 'M' minutiae points in a query fingerprint are retrieved from hash-table.

Choosing 'm' Minutiae Points from 'n' Points and Establishing Rotation Invariance

In the RMT algorithm, we match a query fingerprint MA_V against the MA_V in the hash-table. The selection of the same starting minutiae combination (M_C) belonging to gallery fingerprints is not assured when producing minutiae invariants (I) for query fingerprints. Hence, to make the algorithm alignment-free and robust towards any global transformations, we generate minutiae invariants for all minutiae combinations (M_C) in query, gallery fingerprints.

Fig. 10 Minutiae extracted from 101_1 fingerprint (FVC2004_DB1). **a** Original fingerprint. **b** Fingerprint extracted using SFM [32]. **c** Fingerprint extracted using ExtractNet [33], **d** ground truth fingerprint

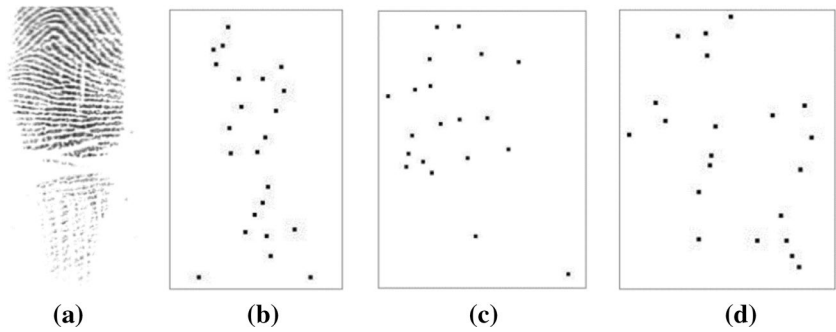


Fig. 11 Minutiae extraction of query fingerprint 101_1 (FVC 2004_DB1). **a** Original fingerprint. **b** Fingerprint extracted using FpMV. **c** Fingerprint compressed and rotated using MINDTCT [38] application

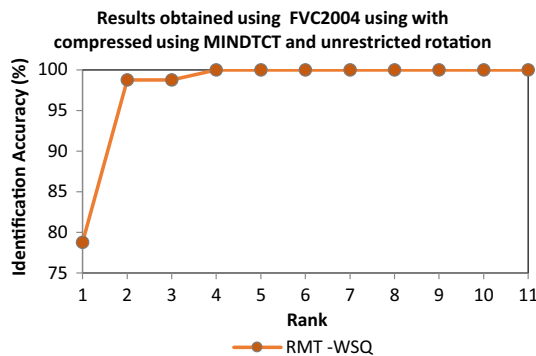
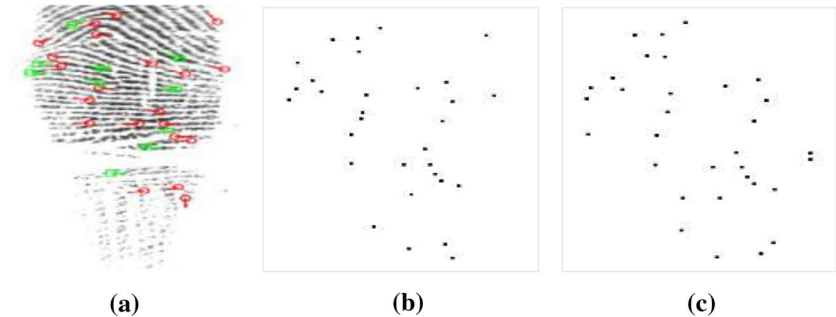


Fig. 12 Rank-1 identification accuracy of RMT algorithm tested on FVC 2004 database query fingerprints with WSQ compression and unrestricted rotation

To start with the RMT algorithm, we identified ‘n’ minutia neighborhood points around a reference minutia and selected ‘m’ minutia points to obtain ‘n C m’ minutiae combinations (M_C). For each M_C , we calculate minutiae invariants, store them on minutiae arrangement vectors (MA_V). To perform these operations, the value of n and m are carefully selected after satisfying the following conditions.

1. If the value of $n = m$, then the number of $M_C = nCm = 1$. Hence, the value of $m < n$ to generate a sufficient number of M_C as well as minutiae invariants (I).
2. For every ‘ M_C ’, we select A, B, C, D, and E co-planar points ($M_p = 5$), and calculate minutiae invariants (I).

The value of $n \geq M_p$ to obtain at least one minutia invariant for a reference central minutia.

3. Large value of ‘n’ will produce a greater number of M_C and it results in good discriminative power. However, this adds the complexity of calculating a large number of invariants (I) for a reference minutia and it also increases the memory storage requirement of the hash-table.

Since the value of $n > m \geq M_p$, we select $n = 8$, $m = 7$, $k = 15$ to satisfy the above conditions.

Results and Discussion

We test the performance of our proposed RMT algorithms on FVC2004 [35] and NIST SD27 [36] databases separately. We store the ground truth fingerprints in the gallery database and apply the RMT fingerprint registration algorithm (Algorithm 1) to generate hash-indexes (HTindex). Later, we store the minutiae arrangement vectors (MA_V) on the hash-table location indexed by HTindex value. Similarly, for query input fingerprints we apply the RMT fingerprint retrieval algorithm (Algorithm 2) to generate hash-indexes (HTindex) and retrieve matching MA_V . Fingerprints with matching MA_V receive the votes for corresponding fingerprints and the fingerprints are sorted based on the descending order of votes received. The fingerprint with the highest vote represents the top Rank-1 result. For both the FVC and NIST databases, we set values of n , m , and k (Algorithms 1 and 2) as 8, 7, and 15, respectively.

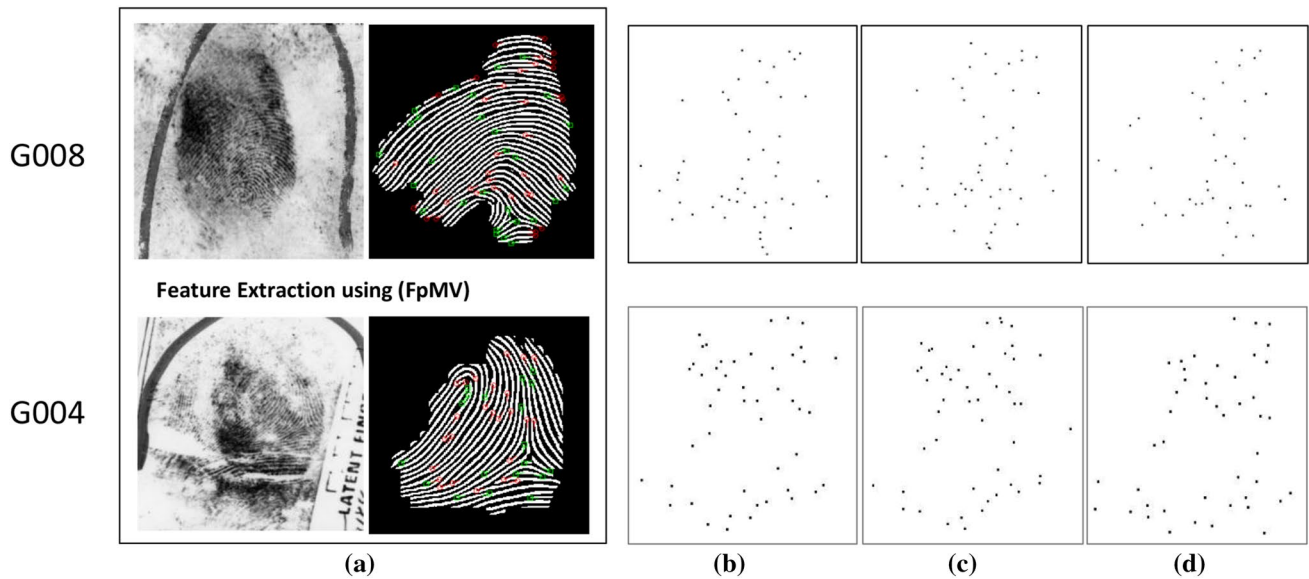


Fig. 13 Minutia points extracted from good-quality G008 (top) and G004 NIST SD27 latent fingerprints using different methods. **a** Pre-processed, enhanced, and extracted query fingerprints using FpMV application [37]. **b** Gallery fingerprint database constructed using

COTS1 [34] matcher. **c** Gallery fingerprint database constructed using COTS2 [34] matcher. **d** Manually annotated NIST SD27 ground truth fingerprints corresponding to **a**

Fig. 14 NIST SD27 query fingerprint (G004) rotation and scaling operation. **a** Original fingerprint. **b** Fingerprint extracted using FpMV. **c** Fingerprint randomly rotated and scaled by 10%

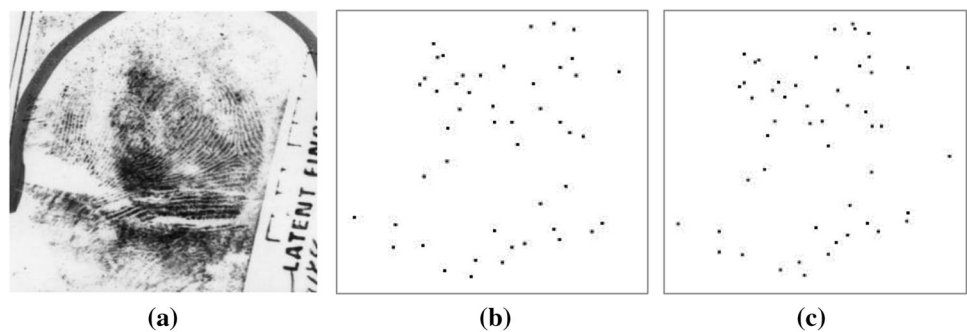


Table 2 The Rank-1 identification accuracy reported by the proposed RMT algorithm and state-of-the-art algorithms for NIST SD27 [25] public dataset using minutiae features

References	Algorithm	Performance results
Jain et al. [27]	Neighboring minutiae-based descriptor-2 (NMD-2)	Rank-1 identification accuracy of 74% with extended features
Paulino et al. [23]	Descriptor-based hough transform (DBHT)	Rank-1 identification accuracy of 53.5%
Miguel Angel Medina-Pérez et al. [26]	Clustering algorithms—cylinder codes (CC), m-triplets, neighbouring minutiae-based descriptor (NMD)	Rank-1 identification accuracy of CC 85.66%, m-triplets 83.3%, NMD 82.9%
Cao et al. [31]	Latent matcher (LM)	NIST SD 27 Rank-1 identification accuracy of 70%
Proposed	Ratio of minutiae triangles (RMT) algorithm	Rank 1 identification accuracy of 86.82%

FVC 2004 Setup

We store the ground truth fingerprints in the gallery database as seen in Fig. 10d. To increase the size of the background database, we extract the minutiae features from different

extraction methods and store them in the gallery database. We accomplish this task by extracting fingerprint features from a MATLAB application called “Simple Fingerprint Matching (SFM)” [32] and a deep neural network model called “ExtractNet” [33] as shown in Fig. 10b, c. The total

Fig. 15 Different quality NIST SD27 latent fingerprints. **a** Good, **b** bad, and **c** ugly

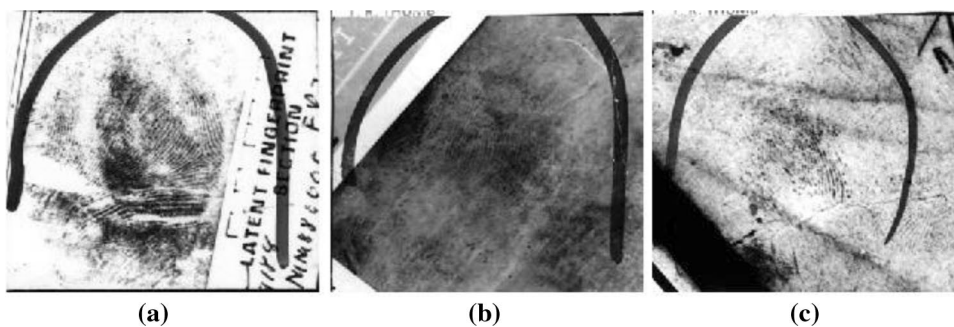
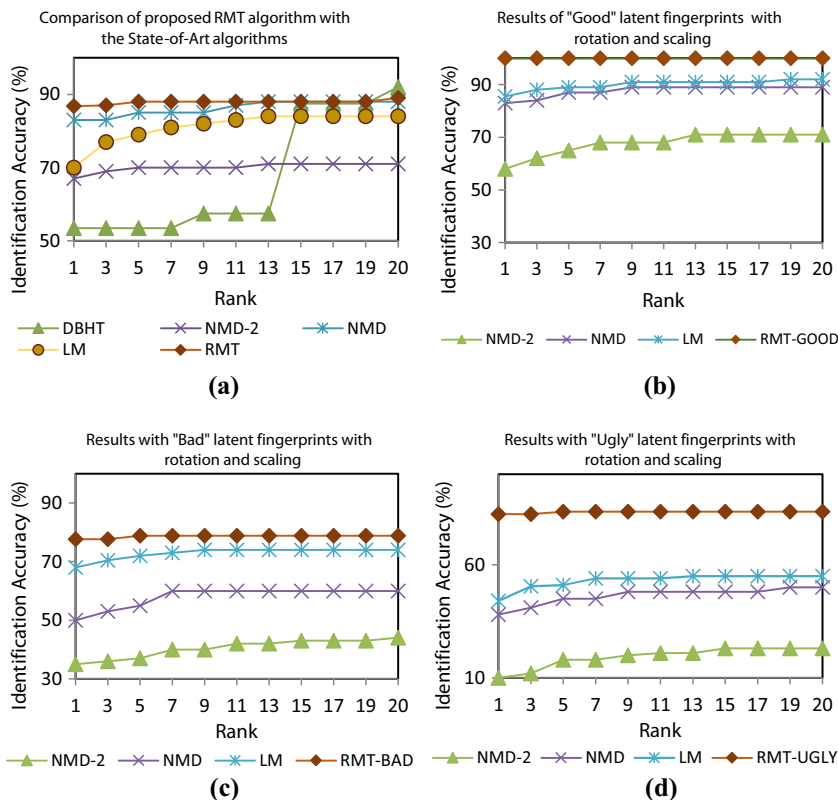


Fig. 16 CMC curves displaying latent fingerprint-matching results of proposed RMT algorithm vs different state-of-art algorithms on a NIST SD27 database. **a** All fingerprints, **b** good quality, **c** bad quality, and **d** ugly quality fingerprints



size of the gallery database (including ground truth) formed is equal to 2560 images.

To perform matching, we extract minutiae features from query fingerprints using the NIST FpMV application. We achieved the highest Rank-1 identification accuracy of 100% on FVC2004 query fingerprints. To test the robustness of our proposed RMT algorithm, we compress (RMT-WSQ) and rotate the query fingerprint images of the FVC2004 database using another NBIS MINDTCT application [38] as shown in Fig. 11a–c.

We obtained a Rank-1 identification accuracy of 78.75% and Rank-4 identification accuracy of 100%. Figure 12 shows the CMC curve of top Rank-20 FVC2004 fingerprint retrievals.

NIST SD27 Setup

We store the manually annotated NIST SD27 ground truth fingerprints in the gallery database as shown in Fig. 13d. To increase the size of the gallery database, we extract minutiae features using the COTS application [34]. Figure 13b, c indicates the minutiae features extracted for G008 (top) and G004 (bottom) ground truth fingerprints using COTS1 and COTS2 matchers. A total of 3096 fingerprints are created and stored in the background gallery database. Minutiae features for query fingerprints are extracted using the FpMV minutiae extractor. Figure 13a illustrates the minutiae extraction operation on good-quality G008 and G004 query fingerprints using the FpMV tool.

Table 3 Confusion matrix which classifies NIST SD27 fingerprints belonging to two different classes as 4 possible results

	Predicted		
	Class	Positive	Negative
Observed	Positive	224	20
	Negative	34	238

Table 4 Different measures derived from the confusion matrix

Measure	Percentage
Error rate	10.47
Accuracy	89.53
Sensitivity (recall or true-positive rate)	87.50
Specificity (true-negative rate)	87.50
Precision positive predictive value	86.82
False-positive rate	12.50

As seen in Fig. 13a, the query fingerprint contains many spurious, and missing minutia points compared to its corresponding manually annotated ground truth minutiae image (Fig. 13d). To further test the robustness of the proposed algorithms, we randomly rotate the query fingerprint images of the NIST SD27 database between 0 and 360 degrees and scale by 10% (see Fig. 14a–c).

The performance of state-of-art algorithms developed for NIST SD27 fingerprint matching is listed in Table 2. The highest Rank-1 matching accuracy of 85.6% was achieved by clustering-based cylinder codes [26]. This algorithm uses minutiae descriptors which are rotated to a maximum value of $\pi/4$ for alignment of the fingerprint. Our proposed RMT does not require any prior information about fingerprint pre-alignment. It can be observed from Fig. 16a that our proposed RMT algorithm achieves the maximum 86.82% Rank-1 identification accuracy. Compared to the state-of-the-art Cylinder-Codes [26], the RMT algorithm produced improvement in Rank-1 accuracy by 1.16%. Further, RMT algorithm showed improvement in Rank-1 identification accuracy of 12.82%, 20.42%, and 16.82%, respectively compared to the other state-of-art NMD-2 [27], DBHT [23], and LM [31] algorithms.

Further, we conduct experiments on a different category of NIST SD27 latent fingerprints. Based on the qualities of the latent fingerprints in NIST SD27, the latent examiners classified fingerprints as “good”, “bad”, and “ugly” fingerprints. The classification is subjective and examples of different quality NIST SD27 latent fingerprints are highlighted in Fig. 15.

Good, bad, and ugly quality NIST SD27 fingerprints are tested individually and the results were reported. We obtained a Rank-1 identification rate of 100% over 85.5%

(LM), 83% (NMD), and 58% (NMD-2) matchers for good-quality fingerprints as shown in Fig. 16b. For bad category images, we obtained a Rank-1 identification rate of 77.65% over 68% (LM), 50% (NMD), and 35% (NMD-2) matchers as shown in Fig. 16c. Finally, for the ugly category of images, we obtained a Rank-1 identification rate of 82.35% over 44% (LM), 38% (NMD), and 10% (NMD-2) matchers as seen in Fig. 16d.

As discussed earlier, we restricted our experiments to NIST SD27 gallery images. The experiments can be extended on a large background gallery database created by adding NIST SD4, SD14, and synthetic fingerprints to the existing NIST SD27 background database.

Confusion Matrix

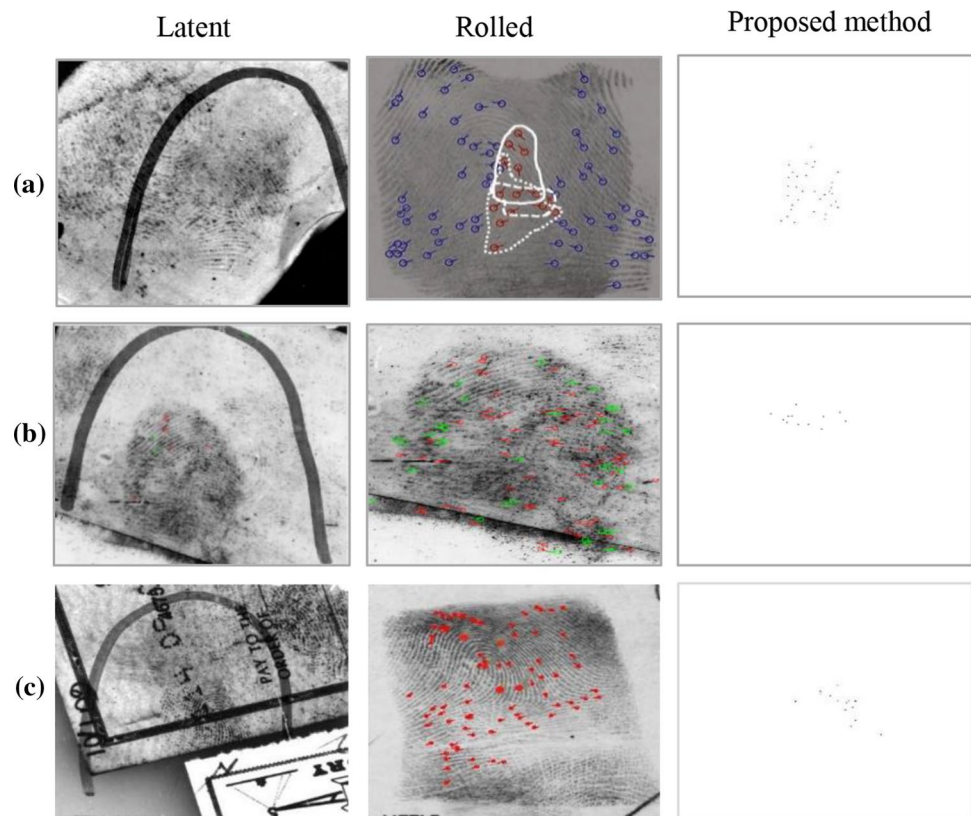
The confusion matrix is used to measure the performance of fingerprint sets belonging to two different classes. The right diagonal elements classify the fingerprints as true-positive and true-negative instances as correct, whereas left diagonal elements classify fingerprints as false-positive and false-negative instances as incorrect.

We select 258 latent fingerprints from the NIST SD27 database and other 258 fingerprint images from the FVC2002 database to measure the performance of this 2-class problem. Hence, a total of 516 query fingerprints from 2 different datasets are formed.

The confusion matrix shown in Table 3 is used to measure the performance of a two-class problem. Based on these two classifications, it generates four possible outcomes. A right match is treated as the “positive class” in the case of the fingerprint-matching problem, and the wrong match is treated as the “negative class”. The right diagonal elements distinguish instances of True Positive (TP) and True Negative (TN), while the left diagonal elements incorrectly classify instances of False Positive (FP) and False Negative (FN). TP is used to calculate the right positive prediction of fingerprints and FP is used to evaluate the wrong prediction of positive fingerprints. Whereas, the accurate negative fingerprint prediction is given by TN and the incorrect negative prediction by FN.

A gallery fingerprint dataset is formed with two distinct classes to obtain a confusion matrix for our proposed RMT algorithm. A Gallery fingerprint dataset with 512 instances is created for NIST SD27 query fingerprints by selecting 256 fingerprints from NIST SD27 and 256 fingerprints from FVC2002. The fingerprints from these two cases are selected randomly during processing. The current and expected classification is demonstrated in Table 2. The TP, FP, TN, and FN values were obtained as 224, 34, 238, and 20 fingerprints, respectively. Various steps calculated from the confusion matrix are shown in Table 4.

Fig. 17 Identified and un-identified NIST SD27 latent fingerprints by RMT. **a** Good-quality G005 fingerprint un-identified by the MCC SDK v1.4 [39], M3gl [40] and identified by RMT. **b** Ugly-quality U213 (overlapping) fingerprint un-identified by the state-of-art algorithm [26] and identified by RMT. **c** Bad-quality B119 fingerprint un-identified by our proposed RMT algorithm as well as the state-of-art algorithms [26, 27]



Latent Fingerprints Unidentified by RMT

Our proposed RMT algorithm was able to identify good-quality (G005) fingerprint which was un-identified by the state-of-art MCC SDK v1.4 [39], and M3gl [40] applications. Similarly, ugly quality (U213) fingerprint was identified by our RMT algorithm which was un-identified by the state-of-art algorithm [26]. RMT algorithm was able to generate a sufficient number of discriminative minutiae arrangement vectors (MA_V), and vote for the corresponding ground truth fingerprints while retrieving. However, our proposed RMT algorithm and state-of-the-art algorithms [26, 27] were unable to identify the correct ground truth fingerprint due to a large number of spurious minutia points contained in low-quality (B119) fingerprint. For our algorithm to work successfully, the minutiae extraction framework should detect at least eight minutia points ($n=8$) to generate invariant feature vectors (MA_V). The identified and un-identified fingerprints by our RMT algorithm is illustrated in Fig. 17.

Conclusion and Future Scope

Unlike normal fingerprints, latent fingerprints present distinct challenges. This is primarily because incomplete and low-quality ridge data are found in the latent fingerprints. The core points required for fingerprint alignment may not

always be present because of its partial fingerprint capture area and incomplete minutia information. Due to these reasons, latent fingerprint identification is considered to be a more challenging research field. We have proposed a geometric transformation invariant automatic latent fingerprint identification system that can detect a person without fingerprint pre-alignment. The proposed system can identify a person with missing minutia or partial minutia information. To achieve this, we proposed “Ratio of Minutiae Triangles” (RMT) fingerprint registration and retrieval algorithms. The algorithm makes use of existing nearest minutia neighbors to generate unique invariant feature vectors. We used the ratio of triangles on minutiae neighborhoods and obtained discriminative feature vectors. These feature vectors were stored in the hash-table and later compared with feature vectors of query fingerprints to perform matching. The proposed method does not utilize any pre-fingerprint alignment information and works well on partial fingerprints. To test the robustness of algorithms, we compressed, rotated FVC2004 query fingerprints and matched them with the gallery fingerprint database. To test the performance of the criminal NIST SD27 database, we randomly rotated-scaled query fingerprints and performed matching with the gallery database. RMT algorithm produced the highest Rank-1 identification of 78.75% for WSQ compressed FVC2004 dataset and 86.82% for NIST SD27 dataset, respectively. The reported experimental results showed that the RMT

algorithm produced improved results on all NIST SD27 fingerprints as well as good, bad, and ugly quality fingerprints (separately) compared to the state-of-art algorithms. The performance of the proposed RMT algorithm proved that it is robust against rotation and scale.

However, the proposed method performs minutia feature extraction and matching separately. This affects the performance of matching in terms of matching time. The system was tested with the gallery database of 3 k images and the match time was moderate. For large fingerprint databases, the match time will increase with the existing separate extraction and matching steps. An improved end-to-end automatic latent fingerprint identification system with integrated automatic feature extraction and matching steps can be developed to overcome the limitation of the proposed system. Since the bad, and ugly quality fingerprints contain very few minutia points, an improved enhancement algorithm can help to extract a sufficient number of genuine minutia points to improve matching performance.

References

- Galton F. Finger prints. *J Anthropol Inst Gt Br Irel.* 2020;22:1893. <https://doi.org/10.2307/2842054>.
- Maltoni D, Maio D, Jain AK, Prabhakar S. Handbook of fingerprint recognition. 2nd ed. New York: Springer; 2009.
- FBI's Criminal Justice Information Services (CJIS)-Next Generation Identification (NGI). 2020. <https://www.fbi.gov/services/cjis/fingerprints-and-other-biometrics/ngi>.
- Guo B, Jain R. iOS Security. 2014. https://www.cse.wustl.edu/~jain/cse571-14/ftp/ios_security/index.html.
- How Biometrics Assure Identity - U.S. Department of Homeland Security. 2020. <https://www.dhs.gov/obim-biometric-identification-services>.
- Wilson C, Grother P, Micheals R, Otto S, Watson C, Hicklin R, Korves H, Ulery B, Zoepfl M. Fingerprint vendor technology evaluation 2003: summary of results and analysis report. Gaithersburg, MD; NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology; 2004. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=151592. Accessed 7 Apr 2021.
- Print case - shames Scots justice. BBC News. 20 September 2004. http://news.bbc.co.uk/2/hi/uk_news/scotland/3671938.stm.
- Cole SA. More than zero: accounting for error in latent fingerprint identification. *J Crim Law Criminol.* 2004;95(3):985–1078.
- Houck MM. CSI: reality. *Sci Am.* 2006;295(1):84–9. <https://doi.org/10.1038/scientificamerican0706-84> (PMID: 16830684).
- Blotta E, Moler E. Fingerprint image enhancement by differential hysteresis processing. *Forensic Sci Int.* 2004;141(2):109–13.
- Liu T, Zhu G, Zhang C, Hao P. Fingerprint indexing based on singular point correlation. In: IEEE International Conference on Image Processing 2005, Genova, Italy, 2005, pp. II–293. <https://doi.org/10.1109/ICIP.2005.1530386>.
- Jiang X, Liu M, Kot A. Fingerprint retrieval for identification. *IEEE TIFS.* 2006;1:532–42.
- Jain AK, Feng J, Nagar A, Nandakumar K. On matching latent fingerprints. In 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Jun; 2008. pp. 1–8. <https://doi.org/10.1109/CVPRW.2008.4563117>.
- Feng J, Zhou J, Jain AK. Orientation field estimation for latent fingerprint enhancement. *IEEE Trans Pattern Anal Mach Intell.* 2013;35:925–40.
- Paulino AA, Jain AK, Feng J. Latent fingerprint matching: fusion of manually marked and derived minutiae. In: 2010 23rd SIB-GRAPI Conference on Graphics, Patterns and Images, Gramado, Brazil, 2010, pp. 63–70. <https://doi.org/10.1109/SIBGRAPI.2010.17>.
- Su C, Srihari SN. Latent fingerprint core point prediction based on gaussian processes. In 20th International Conference on Pattern Recognition, {ICPR} 2010, Istanbul, Turkey, 23–26 August 2010; 2010. pp. 1634–1637. <https://doi.org/10.1109/ICPR.2010.404>.
- Ulery BT, Hicklin RA, Buscaglia J, Roberts MA. Accuracy and reliability of forensic latent fingerprint decisions. *Proc Nat Acad Sci USA.* 2011;108(19):7733–8.
- Bhanu B, Tan X. Fingerprint indexing based on novel features of minutiae triplet. *IEEE Trans Pattern Anal Mach Intell.* 2003;25:616–22.
- de Boer J, Bazen AM, Gerez SH. Indexing fingerprint databases based on multiple features. In: ProRISC the 12th Annual Workshop on Circuits, Systems and Signal Processing Workshop. STW; 2001. pp. 300–6.
- Germain R, Califano A, Colville S. Fingerprint matching using transformation parameter clustering. *IEEE Comput Sci Eng.* 1997;4(4):42–9.
- Liang X, Asano T, Bishnu A. Distorted fingerprint indexing using minutia detail and delaunay triangle. In: 2006 3rd International Symposium on Voronoi Diagrams in Science and Engineering, Banff, AB, Canada, 2006, pp. 217–23. <https://doi.org/10.1109/ISVD.2006.42>.
- Liu E, Arora SS, Cao K, Jain AK. A feedback paradigm for latent fingerprint matching. In: 2013 International Conference on Biometrics (ICB), Madrid, Spain, 2013, pp. 1–8. <https://doi.org/10.1109/ICB.2013.6613013>.
- Paulino AA, Feng J, Jain AK. Latent fingerprint matching using descriptor-based hough transform. *Trans Info For Sec.* 2013;8(1):31–45. <https://doi.org/10.1109/TIFS.2012.2223678>.
- Fisher E, van der Stelt PF, Ostuni J, Dunn SM. The effect of independent film and object rotation on projective geometric standardization of dental radiographs. *Dentomaxillofac Radiol.* 1995;24(1):5–12. <https://doi.org/10.1259/dmfr.24.1.8593908> (PMID: 8593908).
- Cappelli R, Ferrara M, Maltoni D. Fingerprint indexing based on minutia cylinder-code. *IEEE Trans Pattern Anal Mach Intell.* 2011;33(5):1051–7. <https://doi.org/10.1109/TPAMI.2010.228>.
- Medina-Pérez MA, Moreno AM, Ballester MAF, Garcia-Borroto M, Loyola-González O, Altamirano-Robles L. Latent Fingerprint Identification Using Deformable Minutiae Clustering. *Neurocomputing.* 2016;175(PB):851–65. <https://doi.org/10.1016/j.neucom.2015.05.130>.
- Jain AK, Feng J. Latent fingerprint matching. *IEEE Trans Pattern Anal Mach Intell.* 2011;33(1):88–100. <https://doi.org/10.1109/TPAMI.2010.59> (PMID: 21088321).
- Hawthorne MR. Fingerprints analysis and understanding. Boca Raton, London, New York: CRC Press, Taylor and Francis Group; 2009. <https://doi.org/10.1201/9781420068658>.
- Cao K, Jain AK. Automated latent fingerprint recognition. *IEEE Trans Pattern Anal Mach Intell.* 2019;41(4):788–800.
- Danilo VR, Medina-Pérez MA, Monroy R, “Stacking fingerprint matching algorithms for latent fingerprint identification”, Iberoamerican Congress on Pattern Recognition. Cham: Springer; 2019.
- Kai C, et al. End-to-end latent fingerprint search. *IEEE Trans Inf Forensics Secur.* 2020;15:880–94.
- Abraham J, Kwan P, Gao J. Fingerprint matching using a hybrid shape and orientation descriptor. In: Yang J, Nanni L, editors.

- State of the art in Biometrics. IntechOpen; 2011. <https://doi.org/10.5772/19105>. Available from: <https://www.intechopen.com/books/state-of-the-art-in-biometrics/fingerprint-matching-using-a-hybrid-shape-and-orientation-descriptor>.
33. Deshpande UU, Malemath VS. MINU-EXTRACTNET: automatic latent fingerprint feature extraction system using deep convolutional neural network. In: Santosh KC, Gawali B, editors. Recent Trends in Image Processing and Pattern Recognition. RTIP2R 2020. Communications in Computer and Information Science, vol 1380. Singapore: Springer; 2021. https://doi.org/10.1007/978-981-16-0507-9_5.
 34. Liu E, Jain AK, Tian J. A coarse to fine minutiae-based latent palmprint matching. *IEEE Trans Pattern Anal Mach Intell.* 2013;35(10):2307–22.
 35. Maio D, Maltoni D, Cappelli R, Wayman JL, Jain AK. FVC2004: third fingerprint verification competition. In: Zhang D, Jain AK, editors. Biometric Authentication. ICBA 2004. Lecture Notes in Computer Science, vol 3072. Berlin, Heidelberg: Springer; 2004. https://doi.org/10.1007/978-3-540-25948-0_1.
 36. Garris MD, McCabe RM. NIST Special Database 27: Fingerprint Minutiae from Latent and Matching Tenprint Images, Technical Report. Gaithersburg: National Institute of Standards and Technology; 2000.
 37. NIST Fingerprint Minutiae Viewer (FpMV). Available online at: <https://www.nist.gov/services-resources/software/fingerprint-minutiae-viewer-fpmv>.
 38. Watson CI, Garris MD, Tabassi E, Wilson CL, McCabe RM, Janet S, et al. User's guide to NIST biometric image software (NBIS). In: NIST Interagency/Internal Report 7392. 2007. <https://doi.org/10.6028/NIST.IR.7392>.
 39. Biometric System Laboratory, University of Bologna. <http://biolab.csr.unibo.it/researchPages/download/MCCSdk%20v2.0.zip>.
 40. Medina-Pérez MA, García-Borroto M, Gutierrez-Rodríguez AE, Altamirano-Robles L. Improving fingerprint verification using minutiae triplets. *Sensors.* 2012;12(3):3418–37. https://www.codeproject.com/KB/Library/MatchingFramework/FingerprintRecognition_v2.2.zip.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.