Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2023) 31: 412 – 430

doi:10.55730/1300-0632.3992

A multistep fusion matcher approach for large scale latent fingerprint/palmprint recognition

İsmail KILINÇ*, Yusuf ARTAN, Emre BAŞESKİ
Image and Video Processing Group, HAVELSAN Incorporation, Ankara, Turkey

Received: 24.08.2022 • **Accepted/Published Online:** 03.02.2023 • **Final Version:** 23.03.2023

Abstract: Latent fingerprints are ubiquitously used as forensic evidence by law enforcement agencies in solving crimes. However, due to deformations and artifacts within latent fingerprint images, performance of the automated latent recognition systems are far from desired levels. A basic matcher specifically designed for clean fingerprints using a minutiae-based matching algorithm can have high speed and accuracy in a sensor-to-sensor matching task, but low accuracy in matching latent prints, due to scale, rotation and quality differences between latent and sensor images. In this study, we propose a unique multistep fusion matcher (FM) on top of a base matcher that would utilize scale, rotation, and quality attributes of minutiae with speed, memory, and accuracy trade options in the latent recognition process. FM match characteristics are analyzed by using a private dataset consisting of 5560 latent and 1M slap/rolled fingerprint images. In addition, 292 domain expert selected latents are used to compare the nationwide performance of the proposed method. FM's with multiresolution fusion (MRF) option have achieved competitive accuracy rates when searching 292 latent against 1 million background and projecting predictions for 69 million background. On the NIST SD302 public dataset, FM6 (FM option prioritizing accuracy for latent-to-sensor search) with MRF correctly recognizes 911 latent in rank-1, while the COTS system referenced in the NIST SD302 documentation recognizes only 790 from a gallery composed of 5950 latent and 100K rolled background database. FM6 MRF rank-1 count for 10K latent of NIST SD302 is 1415, whereas NIST's referenced matcher rank-1 count is 880 for the same dataset. In addition, NIST SD302 rank-1 latents are used to construct 722 latent pairs to evaluate latent-to-latent matching performance. FM8 (FM option prioritizing accuracy for latent-to-latent search) with MRF has 46.1% rank-1 identification rate for latent-to-latent search against 10K latent background. Moreover, on a private 1457 latent palmprint versus 2296 sensor palmprint background, a palm matcher designed by dividing latent and palm images into 512x512 pixel segments produces 85.45% rank-1 accuracy by using FM6.

Key words: Latent recognition, multistep fusion, cylinder-code, NIST SD302, fingerprint matcher, palmprint matcher

1. Introduction

For the past several decades, fingerprint recognition systems have been the most widely adopted biometrics in the world thanks to the ease of data acquisition process and the cost effectiveness of the overall system [1–3]. Applications of fingerprint recognition systems have been expanding fast in recent years with some of the popular application domains being law enforcement & forensic agencies, border control, social security, and national ID programs. Fingerprint images can be broadly classified into 3 categories; rolled, slap/plain, and latent [1, 3–5]. While rolled fingerprints are typically larger in size and contain greater number of minutiae, slap fingerprints

*Correspondence: ikilinc103@gmail.com

are often less deformed and have clear ridges. Since rolled and slap images are typically acquired through digital fingerprint scanners under the supervision of a law enforcement officer, they are generally considered to have high image quality. Figures 1a–1c demonstrate sample slap, rolled, and latent images from a publicly available NIST SD302 dataset [6]. However, latent fingerprint scans are collected from the crime scene through a variety of means simply from photographing the scene to more complex dusting and chemical processing [3]. Therefore, latent images generally have poor image quality and contain unclear ridge structure, strong deformations and artifacts that would render the overall fingerprint recognition task more difficult. Figures 1d and 1e illustrate a sample image pair (rolled-latent) used in this study. In recent years, several public latent datasets, e.g., NIST SD302 [6], NIST SD27 [7] and IIIT-D Latent Fingerprint Database [8] were made publicly available for research purposes [9]. While achieving a high accuracy performance is a desired property of fingerprint recognition systems, operating at high speed with efficient resource usage is also an important property that needs to be considered when designing a matcher algorithm to be used in a large or mega-scale AFIS system that may include millions or even billions of fingerprints (e.g., India's Aadhaar [1] and the FBI's [2] Databases) stored in a database and thousands of fingerprint identification search requests per day received from clients.

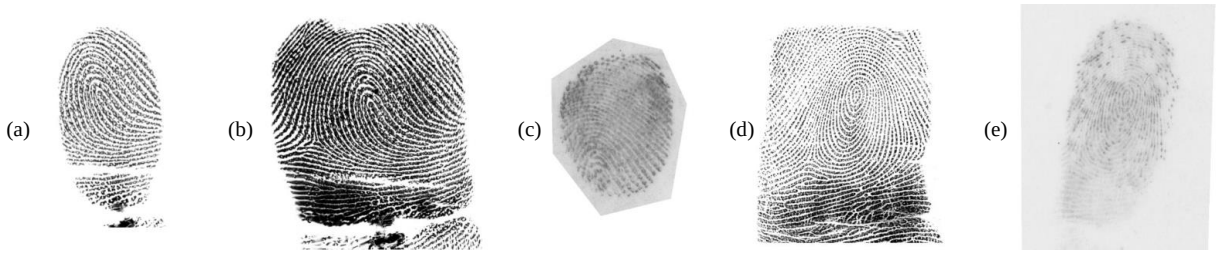

Figure 1. Sample fingerprints (a) slap/plain, (b) rolled, (c) latent (d)-(e) pair of latent and rolled.

Early studies on fingerprint recognition task treated fingerprint matching task as a 2D minutiae point cloud matching problem aimed at resolving a global alignment problem leading to optimal minutia pairings [10, 11]. However, these global matching methods were computationally expensive and lacked robustness against distortions and missing minutiae. To remedy these problems, local minutia matching methods were developed utilizing only spatial coordinates and angular information associated with each minutiae in their analysis [1, 12–14]. In these approaches [12], local descriptors extracted within a fixed neighborhood of minutiae are used to compare similarity of images corresponding to rolled and slap fingerprint impressions. Note that these methods only utilized spatial coordinates/positions, angle and type information associated with each minutia in their analysis. Minutia cylinder coding (MCC) [12] has been considered one of the most popular local matching methods that is computationally efficient and works well compared to earlier methods [10, 11]. Besides MCC algorithm, minutia-triplets [13, 15] and neighboring minutia-based descriptor (NMD) [14] are other popular local matching methods. These algorithms that utilized both local and global features have been widely accepted and used in sensor fingerprint image recognition tasks in the past decade. However, their performance has been limited on latent image recognition task due to distortion within latent images. In order to handle the inherent complexity within latent fingerprints, different strategies have been proposed by researchers [16–19] to extend local matching-based methods. In a more recent study, Perez et al. [20] developed a deformation tolerant local

[1]Unique Identification Authority of India. Website https://uidai.gov.in/aadhaar dashboard/ [accessed 24 August 2022]
[2]Next Generation Identification. Website https://www.fbi.gov/services/cjis/fingerprints-and-other-biometrics/ngi [accessed 24 August 2022]

descriptor-based matching algorithm (DMC) for latent fingerprint recognition task that can handle nonlinear distortion present in latent images. Recent studies [21, 22] reported GPU implementations of these popular approaches, namely MCC and DMC algorithms [12, 20], that would improve the match speed considerably compared to their CPU-based implementations.

In the past few years, we have seen a surge in the automatic latent fingerprint recognition research using deep neural networks [23–28]. Many novel methods have been proposed towards problems within latent recognition task such as latent image minutia extraction, latent image quality assessment, orientation field estimation and minutia descriptor generation tasks using various deep-neural-network-based architectures [26–28]. For instance, Tang et al. [27] presented a deep-learning-based minutia extraction method, FingerNet, for latent image minutia extraction that boosted latent-to-sensor match accuracy significantly compared to earlier methods. In another study, Cao et al. [23] developed an end-to-end latent image recognition method that would utilize deep-learning-based auto-encoder architecture towards minutia extraction in latent and sensor images. In a recent study, Ezeobiejesi et al. [26] proposed a latent image quality assessment method, similar to NFIQ metric [29] used for sensor images, using deep learning techniques. Recently, Öztürk et al. [28] and Engelsma et al. [30] proposed two different deep-learning-based local patch embedding models to represent local ridge flow and spatial/angular minutiae distribution around a fixed neighborhood of a minutiae. Even though these methods have shown success on several private benchmark datasets, their performance improvement is heavily dependent on the usage of GPUs and it is not easy (without compromising speed performance) to directly integrate these algorithms into existing legacy systems that lacks GPUs.

Biometric fusion is another subject that has received a considerable attention of researchers in recent years [2, 5, 31–33]. Paulino et al. [34] presented a method to fuse manually marked and automatically extracted minutiae templates to improve latent fingerprint recognition performance. Several studies [5, 33, 35] have investigated the fusion of plain and rolled fingerprints in latent finger recognition task. However, these fusion methods had limited performance improvements in terms of accuracy because of distortion and artifacts within latent images. While these studies attempted to perform fusion at score level, feature level, and image level, Feng et al. [5] noted that score level fusion provided the greatest improvement in identification accuracy [5]. However, in general, there is only one latent fingerprint, which has scale and rotation distortions since it is left unintentionally and collected from challenging environments with different tools. Other fusion techniques [32] such as multiple matcher algorithm use, multiple feature extractor, or multiple latent or rolled/slap data impression may also improve performance, but their use is out of scope. Tulyakov et al. [31] recently reported that using multiple steps in the fusion process further improves effectiveness and the overall recognition accuracy.

In this study, we propose a multistep fusion matcher (FM) algorithm that jointly optimizes speed, accuracy, and memory in a latent fingerprint recognition task. Table 1 summarizes properties and usage of FM configurations. In the next section, we explain the fusion matcher design, multiresolution usage and its associated details and present the details of segmentation approach used for palmprint matching task. Section 3 presents our detailed experimental results. Section 4 includes summary of experiments and discussions of results. Finally, we present our conclusions and future research directions in Section 5.

2. Methods

Before we explain the details of the proposed multistep fusion matcher, let us briefly present the base matcher, namely MCC algorithm [12], to create cylinder descriptors used in the matching process. MCC algorithm

Table 1. Matcher performance and recommended use for the proposed FM configurations.

Performance	Base	FM23	FM3	FM6	FM8
Speed	fastest	fast	fast	modest	slow
Accuracy	low	modest	modest	high	high
Source memory	lowest	modest	low	modest	high
Target memory	lowest	lowest	low	modest	high

Recommended use	Base	FM23	FM3	FM6	FM8
Latent-to-large sensor database	×	✓	✓	✓	×
Sensor-to-large latent database	×	×	✓	✓	×
Latent-to-large latent database	×	×	×	✓	✓
Latent palmprint use	×	✓	✓	✓	×
Sensor-to-sensor	✓	×	×	×	×

generates a local fixed-length orientation invariant descriptor within a fixed radius around each minutia of the minutiae template. This representation encodes spatial and angular relationships between a minutia and its fixed radius neighborhood in the form of a cylinder whose base and height are related to the spatial and directional information of the enclosed minutiae, respectively. Fundamental parameters used in the original MCC study [12] are number of cells width N_S (8 or 16), number of cells height N_D (6), and maximum rotation allowed between two templates in degree δ_θ (90). Base matcher implementation in this work uses N_S as 8, N_D as 4, and δ_θ as 100. Note that N_S and N_D parameters have a direct impact on matcher speed and memory consumption; these values are chosen to be smaller than the reference study [12] to achieve a fast and lightweight matcher implementation. Base matcher needs 80-byte per minutiae in which 64-byte ($2\times\ N_S\ \times\ N_S\ \times\ N_D$) is used to store a single cylinder and its associated validity bits, and 16-byte is used to store position, angle, and quality information of the minutia. Additional 80-byte is used in each cylinder set as a header part. Base matcher speed for 50x50 minutiae (typical minutia counts for slap-rolled to slap-rolled compare) compare is measured as 86,831 match per seconds for genuine match and 96,901 match per second for impostor match. Base matcher speed for 20x78 minutiae (typical minutia counts for latent to slap-rolled match) is 120,529 for genuine match and 134,967 for impostor match. Base matcher is able to create 1777 cylinder codes in a second for 70 minutiae, which affects template creation or enrollment time. Measurements have been obtained by using a single thread on the hardware reported in Section 3. The base matcher is intolerant for large directional differences close to or greater than δ_θ since this parameter specifies whether to compare two cylinders. The base matcher is also scale intolerant, and minutia quality is not taken into account in the original MCC study. Therefore, we propose a method that is robust for scale, minutia quality, and orientation differences between latent and sensor images.

2.1. Fusion matcher (FM)

While the baseline matcher works well for fingerprints acquired through sensors, its performance degrades for latents with strong deformations/distortions and low image quality. FM handles the inherent distortion problem by generating multiple cylinder templates by performing an affine transformation and by quality thresholding on extracted minutiae. As illustrated in Figure 2, the proposed FM algorithm has two phases: template creation and search.

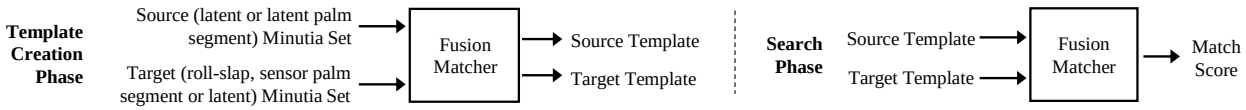

Figure 2. FM template creation phase and search phase.

Template creation phase parameters and lookups are listed in Table 2. The minimum number of minutia required for template creation is specified as 7 for all FM configurations. SS/SR/TS corresponds to the range of scale/rotation values chosen from SS/SR/TS lookups. These values are used to apply affine transformation (scale/rotate) on minutia positions (x, y). For instance, when SS value is 3 or 5, then scale values of [1, 1.2, 0.8] or [1, 1.2, 0.8, 1.12, 0.88] are utilized in template creation phase. A minutiae may be discarded or used according to SQ/TQ thresholds. Minutia extractor generated minutia quality values are typically between 50 and 100 and SQ/TQ lookup values are distributed in this range. MR parameter is used to create cylinder templates for each radius values listed in MR lookup. Radius effects minutia neighborhood selection during cylinder creation. SGM limits number of cylinder generated in template creation phase for source. If mode is 0, then all rotate values (in SR lookup) for scale 1.0, and all scale values (in SS lookup) for rotate 0 are used to create cylinders. If mode is 1, then all scale values for first 3 rotate [0, 90, –90] is used. SFIM specifies the maximum number of source cylinders per latent that will be generated by FM algorithm. The number of cylinders may be less than the minimum minutia for high source quality (SQ) values and for this reason actual item numbers may be less than SFIM value. TFIM specifies the maximum number of target cylinders per finger used for fusion and is set to 1 for FM23 to create a minimum memory solution.

$$\begin{aligned} ASME &= SFIM \cdot\ SC \cdot\ (SMA+1) \cdot\ 80 \\ ATME &= TFIM \cdot\ TC \cdot\ (TMA+1) \cdot\ TFA \cdot\ 80 \end{aligned} \tag{1}$$

Table 2. Base, FM23-3-6-8 template creation phase parameters and lookups.

Parameter	Base	FM23	FM3	FM6	FM8
Source scale (SS) / rotate (SR) / quality (SQ)	1	7 / 8 / 10	5 / 8 / 4	5 / 14 / 8	7 / 14 / 10
Target scale (TS) / quality (TQ)	1	1	5 / 2	5 / 2	5 / 4
Multi radius (MR) / source generation mode (SGM)	-	0 / 1	0 / 0	1 / 0	6 / 1
Source / Target fusion item max (SFIM / TFIM)	1	290 / 1	76 / 10	392 / 20	2702 / 140
SS lookup / TS lookup	1.0 1.2 0.8 1.12 0.88 1.04 0.96 / 1.0 0.88 1.08 0.94 1.04				
SQ lookup / TQ lookup	60 85 50 70 55 65 75 80 90 95 / 60 85 50 75				
SR lookup	0 -90 90 180 -30 30 -60 60 -15 15 -45 45 -120 120				
MR lookup	60 76 68 52 84 80 64				

Application Source/Target Memory Estimate (ASME/ATME) in bytes is used to estimate the memory requirement for an application. SC/TC is source/target count. SMA/TMA is source/target minutia average and 80 is the single cylinder memory allocation value in bytes. One value is added to SMA/TMA since there is a header per fusion item. TFA is target finger average per person and it is 20 for complete slap and rolled dataset. TFIM has a significant effect on the overall recognition process since it specifies the amount of memory used for a large target set. If target size is in the tens of millions, then memory requirements would be an issue. For instance, 100 million (TC) rolled and slap set (suppose each person has 20 (TFA) fingerprints = 10 rolled + 10 slap fingerprints) with an average of 79 minutia (TMA) requires $12.8 \times$ TFIM TerraByte memory or disc space. Source cylinder size may be important for slap/rolled to large latent database search, but latent data base sizes are expected to be relatively small when compared to slap and rolled database.

Search phase of the FM algorithm consists of 4 steps as shown in Figure 3 and parameters in Table 3. Each step of this serial architecture aims to keep recall rate high while eliminating mismatches by utilizing a

set of parameters and their threshold levels to control entrance of match operation to next steps. It is assumed that if two fingerprints produce a match score above a certain threshold value for a small number of cylinder combinations, which means the likelihood of belonging to the same person for these two templates is high, it would be better to perform the search in more depth by using all possible cylinder combinations produced in template creation phase. The default match quality threshold value is 84 and it is used to discard minutiae with quality values lower than this value during the matching process. Unused low quality minutiae in match process may not have cylinders, but contribute to cylinder neighborhood of the high quality minutiae. Step 1 to Step 3 uses default match quality, but extra step searches for best match quality to yield the highest match score value. Match max source limits the maximum number of the source minutiae used in the match process. Good quality minutia count for latent is typically between 20 and 30 (e.g., 22 marked minutia average in NIST SD302 dataset) and too many minutiae may lead to false detections. Since the number of minutia used affects the base matcher speed, small number of minutiae are used on initial step (Step 1) and gradually increased in following steps. Match max target limits the maximum number of target minutiae used in the match process. Average count of good quality minutiae for rolled fingerprint is typically around 80 and too many minutiae may result in false detection. This value is higher than expected average minutia count in all steps. Extra step searches for the best maximum target minutia number that yields the highest score. Match Axis usage means all source scale values are used if target scale is 1.0 and all target scale values will be used if only source scale is 1.0 which controls base matcher call counts and mainly used in partial step (Step 2) of FM8 and FM23 and all steps on FM3 as well as on FM6. If source scale and target scale for cylinders are lower than 1.0 or larger than 1.0 at the same time, then these cylinders are not compared to avoid unexpected superious score generation and redundant comparison. Step thresholds α_1 to α_6 used to balance matcher call counts between steps as shown in Figure 3. $\alpha_1/\alpha_2/\alpha_3$ are utilized from Step 1 to 3 sequentially, and α_4/α_5 values are used in Step 4 to control and block unwanted base matcher calls for potential unmatch candidates. Multiradius threshold (α_6) value controls the use of multiradius cylinder templates.

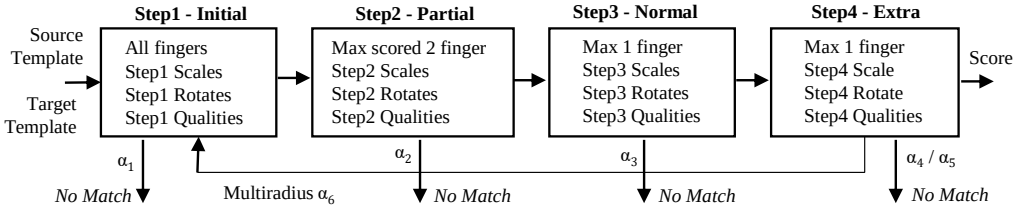

Figure 3. FM search phase initial, partial, normal, and extra steps.

In the latent-to-slap/rolled search process, a latent source is expected to match only with two fingers, i.e. slap or rolled impressions of the same finger within a rolled/slap dataset. For this reason, when searching for a mate of a latent image, at Step 1 (initial step) we eliminate matches with scores less than step-threshold α_1, then the remaining rolled/slap finger matches are reduced to 2 fingers with top matching scores for each person. In Step 2 (partial step), two fingers are reduced to one for Step 3 by choosing the finger with the highest match score. Fusion parameters (such as angle of rotations, scale values etc.) and matcher call counts starts with small numbers and increase from Step 1 to 4. If an FM configuration with the multiradius option is selected, then all steps are executed again if the multiradius threshold level is exceeded.

Figure 4 shows a similarity view for a latent and corresponding sensor image in which the figure on the left shows base matcher similarity and the figure on the right shows FM6 similarity correspondences of matching

Table 3. Base, FM23, FM3, FM6, FM8 search phase parameters. Steps 1–4 are specified in the form of ·/·/·/·.

Parameter	Step	Base	FM23	FM3	FM6	FM8
Source scale	1/2/3/4	1	3/5/7/7	3/3/3/5	3/3/5/5	3/5/5/7
Source rotate	1/2/3/4	1	3/6/8/8	3/3/3/8	3/3/6/14	3/6/6/14
Source quality	1/2/3/4	1	1/2/4/10	1/1/2/4	1/3/4/8	1/3/4/10
Target scale	1/2/3/4	1	1	1/3/3/5	1/3/4/5	3/4/5/5
Target quality	1/2/3/4	1	1	1/1/2/2	1/1/2/2	1/2/4/4
Multiradius	1–4	-	0	0	1	6
Match max source	1/2/3/4	35	16/20/25/35	16/20/25/35	18/20/25/45	18/20/25/45
Match max target	1–3/4	120	120/250	120/250	120/250	120/250
Match axis	1/2/3/4	-	1/1/0/0	1/1/1/1	1/1/1/1	1/1/0/0

minutia pairs. Red minutiae are below in cylinder quality thresholds and not used in the match computations. Yellow minutiae are used in cylinder generation but not used for matching. Green minutiae have cylinders and contribute to matching score. Pairing lines are colored from green to red and numbered from 1 to 12 according to contribution values. Note that 5 minutiae for source and 11 minutiae for target are discarded for cylinder generation (becomes red) and some minutiae are discarded for match (becomes yellow) in FM6. Best source cylinder that produces maximum score has scale 1.0, rotation –60, cylinder quality 70, and match quality 95 for FM6 (right image in Figure 4). Best target cylinder has scale 0.88, cylinder quality 85 and maximum minutia number 81 for this case. Since there is a match, the matching process follows all steps in search phase including the extra step. Target match quality is zero since the extra step gradually decreases the maximum minutia number and sets it to 81 to obtain the highest score without using any match quality threshold. Similarly, the source match quality value increased from the default value of 84 to the optimal value of 95 for FM6. Score is improved by those selections in FM6 from 0.201 to 0.29075 and improvement shall also be verified by focusing on the vanishing pairing lines scattered in the first image.

Quality-related parameters (SQ/TQ lookups) in Table 2 depend on minutiae extractor characteristics and other FM parameters (SS/TS/SR lookups) are applicable for all minutiae extractors, marked manual minutiae and scale and rotation intolerant custom base matchers. In this study, since we utilized FingerNet [27] algorithm for minutia extraction using its default parameters, quality-related lookups are selected according to FingerNet derived minutiae quality distributions. Threshold values depend on score distribution of base matcher, and α_1 to α_6 have been selected to satisfy that step2 to step4 total matcher call counts do not exceed step1 total matcher call counts. SS/TS/SR lookups used in this study are constructed by trial and error. A private dataset among 1000 latent (sampled from private latent dataset randomly) against 10K slap and rolled background is used for heuristics and threshold/quality parameters selections.

2.2. Multiresolution fusion (MRF)

Both minutia extractor and matcher take advantage of fusion by using multiple resolution images. Minutia extractor generates some minutiae better in alternative resolutions than default 500 PPI, and matcher generates better score by using different minutia combinations. Latent images may have inherently scale differences due to varying latent collection methods. By using alternative resolutions, minutiae positions and qualities have been generated slightly differently especially for low-quality minutiae. By using those alternative minutiae, it would be possible to generate better scores by the help of fusion process. However, MRF usage slows matching

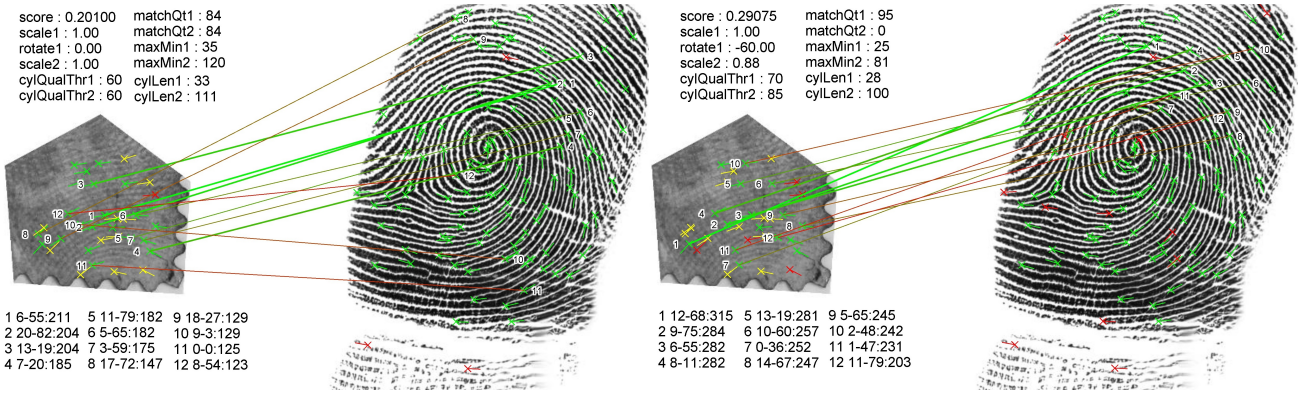

Figure 4. Base (left) and FM6 (right) similarity view for a latent and sensor (images are from NIST SD302 data set).

speed approximately three times, increases source memory use by four times, and needs four minutia extraction execution per image. The use of MRF is not applicable for minutia only input search such as user minutia marks search against sensor database.

Multiresolution fusion shall be applied by using the following procedure: (1) Resample 450, 500, 550, 600 PPI images from higher resolution latent images if available or use 500 PPI latent image for resampling. Create four images by this way. (2) Rotate three of resampled images by 90 (450 PPI), 270 (550 PPI), 180 (600 PPI) degrees. (3) Generate minutiae list for each resampled and rotated image using FingerNet algorithm. (4) Rescale minutia x and y coordinates by 500/PPI. (5) Use each latent resolution template to generate a match score using matcher and sort four scores in descending order (S1 > S2 > S3 > S4). Use the best target finger index found in the first two score calculation to calculate the last two scores to speed up overall matching speed. (6) If nonzero score count is one or two (S1, S2), then use S1, if nonzero score count is three (S1,S2,S3), then use score as $0.7 \times S1 + 0.3 \times S2$, if all four scores are nonzero, then use $0.5 \times S1 + 0.5 \times S2$ for score level fusion.

2.3. Palmprint matcher

Use of palmprint data has an important place in solving criminal cases and approximately 30% of the traces collected from the criminal field are palm prints [36]. Similar to the fingerprint recognition task, minutia-based matching methods are mostly used for latent palmprint identification [37]. Typical latent and sensor image set for palmprint matching for single person is shown in Figure 5. Latent image size is 945×973, two sensor scanners' palmprint image size is 900×2500 and the other four sensor palm image size is 2500×2500 in unprocessed images. Since minutia extractors are not designed for these large-sized images, we divide the image into overlapping segments with maximum 512×512 pixels, which suits well for minutia extraction. Segment boundaries are displayed with red squares filled with 10% red color opacity to visualize overlapping areas. The yellow segment in quadruplicated latent is matched with the green segment in the sensor right lower palm. A minutia similarity view is also shown in Figure 5b. Each segment has maximum 256×512 pixel overlap with the right-hand and the lower neighborhood segment if there is a continuity in ridges. Phalanx and parts that appear separate from others have been segmented without overlap if they fit to 512×512 pixel. Some segments containing partial ridges do not produce minimum required minutia (7), hence not used in the match process. Latent segments are generated for multiple resolutions (450, 500, 550, 600 PPI) with multiple rotations (90, 0, –90, 180) and sensor segments are generated for only 500 PPI for palm-matching performance evaluation.

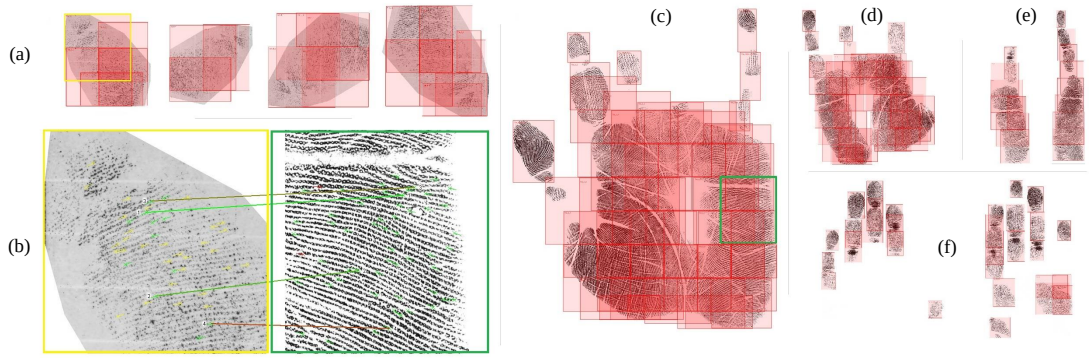

Figure 5. Palmprint matching. (a) Latent segments rescaled to 500, 450, 550, 600 PPI and rotated for 0, 90, –90, 180 degree, (b) similarity view for matching segments, (c) matching sensor right lower palm, (d) sensor left lower palm, (e) left/right writer's palm, (f) left/right upper palm (Images are from NIST SD302 data set).

Minutiae quality with balance count (MQ_{BC}) for a minutia list with sorted qualities $(mq_1, mq_2, ..., mq_N)$ in descending order shall be found by using

$$MQ_{BC} = \sum_{i=1}^{min(N,BC)} mq_i/BC, \tag{2}$$

which takes into account minutiae count and minutiae qualities in a balanced way by using a balance count (BC) value. Latent segmented minutia files are sorted according to MQ_{50} and only top 12 have been used for matching. Six palm regions are processed like six slap or rolled finger with separate finger indices and segments in each palm region are used to expand target fusion items. Overall score is calculated by score level fusion similar with methods explained in Section 2.2. Each segment is matched with full sensor segments like latent against slap-rolled matching and scores are sorted in descending order (S1 to S12). If nonzero score count is two, then S1 is used. If nonzero score count is three, then $0.7 \times$S1$+0.3 \times$S2 is used. Otherwise $0.5 \times$S1$+0.5 \times$S2 is used to calculate overall matching score. It has been observed that using more than 12 segments for input has not improved accuracy and slowed down the matching speed.

3. Experiments

In this section, similar to the earlier studies [38, 39], rank values, cumulative match characteristics (CMC), equal error rate (EER) and FMR1000, match time, memory consumption, and enrollment time are used for evaluation. Table 4 lists the details of the datasets used in our experiments. All minutiae files are generated via FingerNet algorithm [27] except for CLPE (Certified Latent Print Examiner) marks. All private data are based on operationally used data and not synthetic. If minutia count for a file is less than 7, then it is not used.

NIST SD302 [6] is a new public dataset that replaces NIST SD27. It contains 50 latent per 200 person with slap/rolled data. Each latent (DS1) and slap/rolled (DS2) data have images and CLPE (certified latent print examiner) marks for 119 person where 2149 of 5950 (36.1%) latent is individualized although individualization term is not objective and depends on examiner [40]. NIST SD302 contains data collection with optical, touch-free, and solid state sensors. FBI operator assisted slap (R and S) and roll dataset (V) collection specified as baseline.

Table 4. Overview of the datasets used in experiments.

Description	Privacy	Valid file	Average minutiae	Average minutiae quality
DS1: NIST 119×50 latent auto/marked	Public	2872/2522	30.12/21.18	83.37/-
DS2: NIST 119 baseline slap/rolled	Public	1189/1190	57.71/94.45	89.41/88.26
DS3: 5560 latent/slap/rolled	Private	5560/57620/62110	51.2/69.52/106.76	84.22/88.25/86.09
DS4: 292 expert latent/slap/rolled	Private	292/5071/5066	54.79/66.43/105.91	84.27/90.97/89.53
DS5: 1M background slap/rolled	Private	9531290/9960972	64.34/97.28	88.86/87.17
DS6: NIST latent-to-latent pair (722)	Public	3610	47.23	85.56
DS7: 10K latent background	Private	10000	51.36	83.61
DS8: Palm latent (1457)/sensor (1865)	Private	43760/287338	35.75/59.02	80.92/85.78
DS9: NIST palm sensor (431)	Public	49181	47.45	83.42

Private 5560 latent dataset (DS3) consists of images corresponding to distinct individuals and 10 fingerprints of rolled and slap images corresponding to unique people. Expert selected latent dataset (DS4) is a private dataset consisting of 292 latent images (separate from DS3) with slap/rolled images. Expert dataset is of importance since it was prepared by latent-fingerprint-experts and contains samples that are deemed hard to match. An acceptance criteria (in terms of rank-20 identification rate) for 69M background is supplied by experts for this data set. Background dataset (DS5) is 1000K set of rolled/slap images which is mainly used to characterize FM configurations and identification rate changes as background data size enlarged.

Latent-latent dataset (DS6) is a public dataset that comprises 722 latent fingerprint images with corresponding 722 latent images (second impression) and a set of private 10K (DS7) latent background images. Since MRF usage quadruplicates source latent count, total number of files are 3610 (722×5). Latent pairs are constructed by using rank-1 latents which are found by NIST SD302 autogenerated latent search against baseline sensor roll data with 100K rolled background (subset of DS5). If a latent has a rank 1 and matched person number and sensor finger index was the same with another rank 1 latent, then these two latents are considered to be a pair.

1457 latent and 1865×6 sensor image (DS8) as private data with 431×6 sensor image from NIST SD302 database (DS9) is used as palmprint data set. Palmprint sensor images contain 6 impressions that capture different regions of the palm region. Average valid segment count for latent is 30.03 (total of 4 resolution) and for sensor it is 25.68 (154.07 for 6 image per person in single resolution) per image in private dataset. NIST SD302 palm average sensor segment count is 19.02. Although 158 of 431 NIST palm sensor set have not produced minutia files with minimum 7 minutiae for writer's palm images, they are not discarded and used to extend background data size.

3.1. NIST SD302 dataset latent to sensor results

NIST provides its own matcher and its hit rates for 10K (200×50) latent in NIST SD302 with 100K background [41] as shown in Figure 6a. NIST matcher performs better than base matcher and all FM's outperforms NIST matcher in different rates in all data sets. NIST's matcher rank-1 value specified as 880 and FM6 MRF rank-1 value is 1415 for baseline data (V). Average (V, A, B, C, D, E, F) rank-1 is 709 for NIST's matcher and 1180 for FM6 MRF. FM6 MRF performs better 66.43% in rank-1 than NIST's matcher in average identification rates. C has better rank-1 than baseline which complies with NIST's evaluation [42].

Another NIST SD302 document [6] references a COTS rank-1 as 790 for 119×50 latent vs. 100K roll (V)

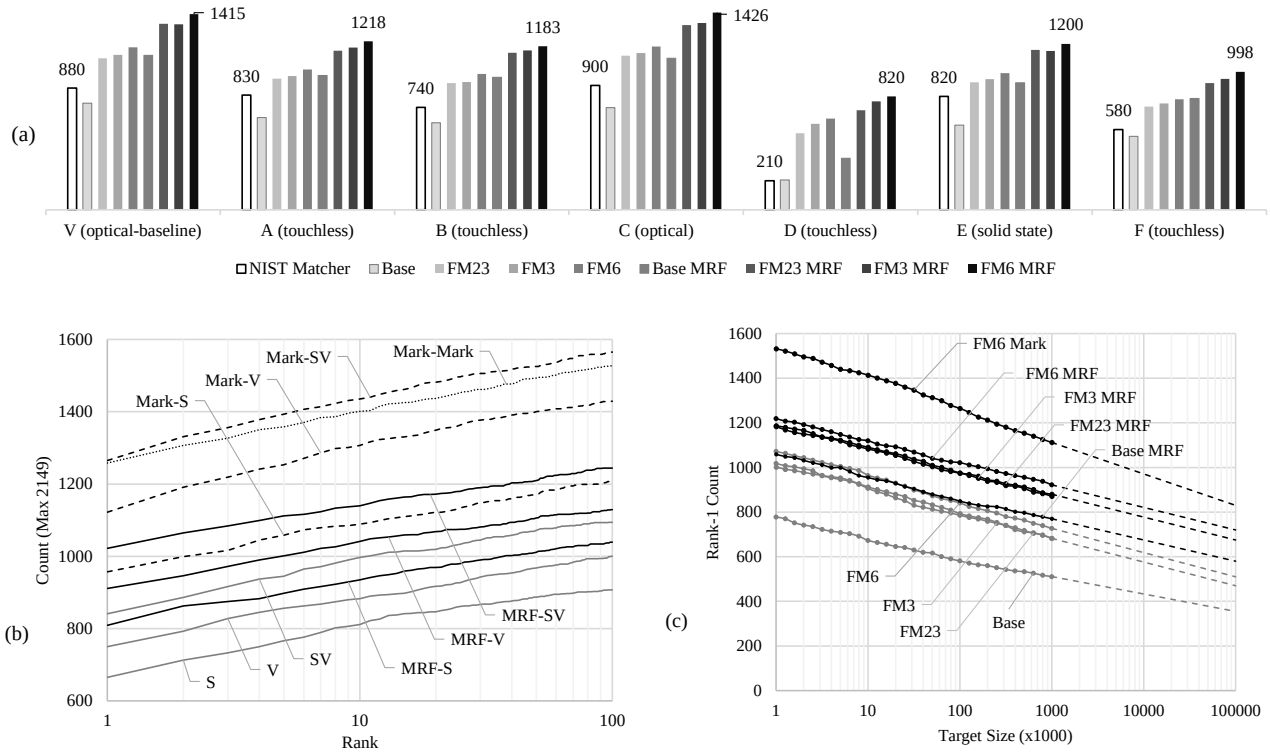

Figure 6. (a) NIST Matcher vs. FM rank-1 for 10K latent with various sensor data against 100K rolled background, (b) FM6 CMC curves for auto and CLPE marked minutia for NIST baseline dataset for 100K background, (c) FM's NIST SD302 rank-1 observations and 100M predictions for 119×50 latent and baseline slap-rolled data (SV)

background. FM6 MRF rank-1 to rank-100 CMC curves for auto and marked latent against baseline dataset (S, V, SV) are shown in Figure 6b. FM6 MRF rank-1 count for auto latent (minutiae extracted using FingerNet) is 1022 and FM6 for marked latent is 1265 for slap and rolled use. Measurements show that a perfect minutia extractor that generates the same minutiae with CLPE's shall enhance latent identification performance by 23.78% for 100K slap rolled background. Figure 6c shows rank-1 observation and prediction lines for 119×50 latent with slap and rolled (SV) sensor data against several background sizes for marked and autogenerated minutiae. Observations show that identification counts gets reduced as background size increases.

Searching marked or automatically generated minutiae on a slap and rolled dataset is the default use case which maps to SV for baseline dataset. Single usage of rolled images gives better accuracy results than slap images, but using slap and roll together improves performance considerably in all configurations. User marks also performs better than auto minutia input, but MRF usage shrinks the gap between auto and marked search with the cost of speed reduction and increased source memory use. According to the results of 119×50 latent against 100K background data presented in Table 5, FM23 MRF (863) / FM3 MRF (859) / FM6 MRF (911) with auto latent to auto rolled (V) target performs 9.24%/8.73%/15.32% better than COTS (790). Target memory values and load time values are considerably higher than that of source as expected since total sensor files (10×100K for S and V, 20×100K for SV) used are significantly higher than total source files (2872 auto, 2522 mark) used. FM6 ASME (application source memory estimate) for auto minutia input is 2.8 GB (SC=2872, SFIM=392, FMA=30.12) and measured value is 2.47 GB which are close to each other. FM6 ASME for mark is

1.75 GB (SC=2522, SFIM=392, FMA=21.18) and measured value is 1.97 which are also close. MRF multiplies source load time by four as expected. FM6 ATME (application target memory estimate) for V is 148 GB (TC≈100K, TFIM=20, TFA≈10, TMA=91.8) and measured value is 152 GB which complies with ATME.

Table 5. FM NIST 302 119×50 (5950) latent vs. 100K background performance.

Source	Target	Rank1				Match time (×1000 s)			
		Base	FM23	FM3	FM6	Base	FM23	FM3	FM6
Auto	S	475	646	643	665	0.188	1.6	1.4	4.0
	V	486	698	720	750	0.261	2.4	1.9	5.4
	SV	581	786	793	841	0.538	2.9	2.3	7.7
Auto MRF	S	681	773	781	809	0.741	6.2	3.6	13
	V	719	**863**	**859**	**911**	1.0	8.1	4.9	16
	SV	849	975	975	1022	1.7	9.5	6.6	20
Mark	S	730	893	892	957	0.186	1.7	1.5	4.3
	V	843	1072	1078	1122	0.316	2.4	2.0	4.5
	SV	966	1214	1218	1265	0.486	2.8	2.4	6.7
Mark	Mark	957	1261	1224	1258	0.311	2.3	1.8	4.7

Dataset	Load time (s) / Memory use (GB)			
	Base	FM23	FM3	FM6
Src Auto	0.26/0.42	3.14/1.74	1.12/0.74	4.52/2.47
Src MRF	0.32/0.4	12.37/5.76	3.69/2.09	31.49/8.68
Src Mark	0.21/0.39	3.22/1.51	0.93/0.61	4.7/1.97
Target S	19.0/5.37	19.1/5.47	134/46.9	593/94.5
Target V	13.5/8.35	16.7/8.55	242/74.0	914/152
Target SV	24.8/12.9	30.1/13.0	541/114	1972/226

3.2. Private latent to slap-rolled dataset results

Table 6 and Figure 7a contains measurements for 5560 latent source versus 100K target data. Rank-1 is used generally in the literature to measure performance, but experts in criminal may use up-to rank-20. Memory-use- and load-time-related observations are similar to NIST SD302 measurements and not presented in this section. All FMs have considerable accuracy improvement according to base matcher. While FM6 MRF has best accuracy, FM23/3 has lower accuracy but is relatively fast.

292 expert selected latent versus 69M target rank-20 performance acceptance criteria is specified by experts as 78.7%. Figure 7b shows FM's observations and predictions for 292 latent. Predicted value for 69M is between 70% and 75% for FM23/3/6. FM23/3/6 MRF mean prediction value is 84% and considerably higher than experts acceptance criteria.

Table 6. FM performance for 5560 latent vs 100K background.

Matcher	Rank-1 (%)	Rank-20 (%)	EER (%)	FMR1000 (%)	Match per second
Base / MRF	59.26 / 70.25	72.48 / 80.23	10.047 / 8.274	30.395 / 24.964	485001 / 113902
FM23 / MRF	74.1 / 84.15	85.14 / 91.56	5.907 / 4.525	15.629 / 10.521	64250 / 20268
FM3 / MRF	76.31 / 85.36	86.37 / 92.19	6.483 / 4.652	14.802 / 10.611	73323 / 33282
FM6 / MRF	79.19 / 87.64	88.08 / 93.06	5.051 / 3.924	13.201 / 9.154	29211 / 10518

3.3. Latent-latent and palmprint dataset results

Latent-to-latent match results for 722 × 10K latent dataset are shown in Table 7. CMC curves are also constructed for base and FMs as shown in Fig. 8 (a) to visualize matcher characteristics. Base rank-1 is 27.42% and FM8 MRF rank-1 is 46.12%. MRF usage improves performance dramatically as shown in CMC curves.

Latent palmprint to sensor palmprint and latent fingerprint to slap-rolled identification results are shown in Table 8 and CMC curve is shown in Figure 8b. Sensor palm index distribution paired with FM3 rank-20 latents have been observed as 62.43% for lower palms, 26.78% for upper palms, and 10.77% for writer's palms. FM23/3/6 have close characteristics on palm matching, and their performance is between base and FM6 of the

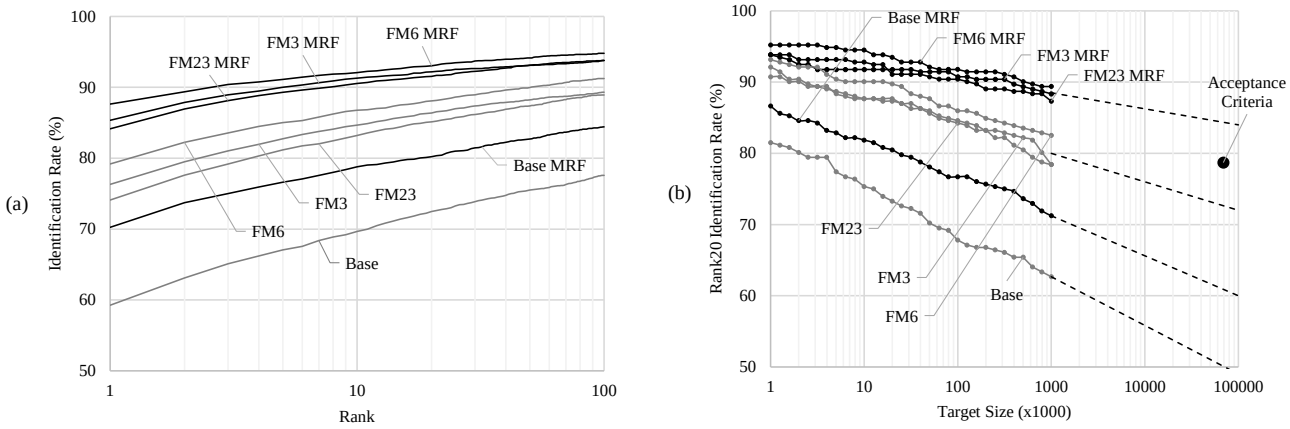

Figure 7. FM's (a) CMC curves for 5560 latent against 100K slap-rolled background, (b) 1 millon slap-rolled background observations and 100M predictions for 292 expert selected latent.

Table 7. FM 722 × 10K latent-to-latent performance.

Matcher	Rank1(%)	Rank20(%)	EER(%)	FMR1000(%)	Src mem. (GB)	Target mem.(GB)	Match per second
Base/MRF	27.4 / 39.5	35.9 / 49.7	34.3 / 27.2	68.6 / 27.2	0.25 / 0.39	0.28	3583915/1061414
FM6/MRF	39.3 / 44.3	48.1 / 52.4	30.0 / 26.7	56.6 / 26.7	1.14 / 3.89	1.43	192066 / 46462
FM8/MRF	39.9 / 46.1	50.6 / 54.6	28.0 / 26.3	55.0 / 26.3	6.28 / 23.4	10.45	13825 / 3934

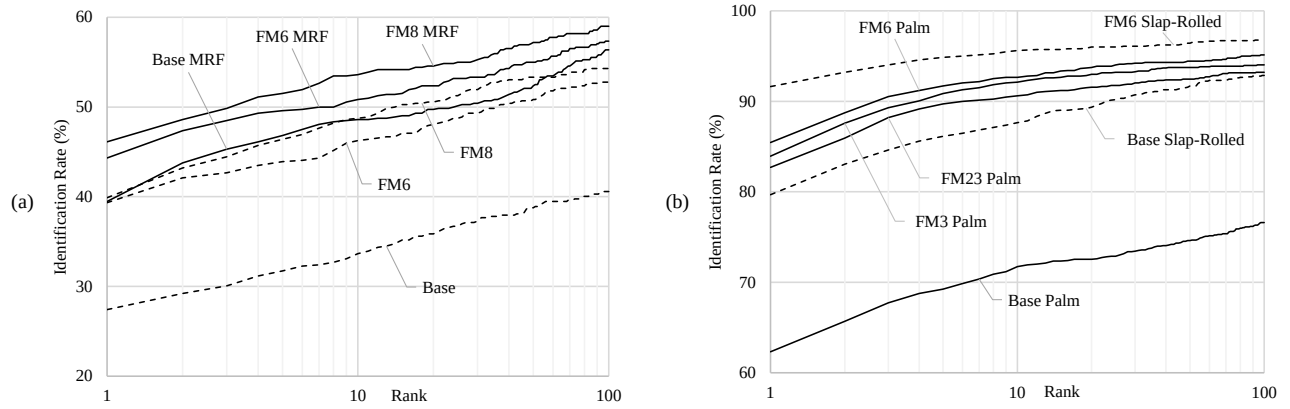

Figure 8. (a) CMC curves of 722 × 10K latent-latent dataset (b) CMC curves of palm and fingerprint for 1457 × 2296 data

matcher's fingerprint matching performance. However, latent palmprint matcher consumes higher memory and is slower than latent fingerprint matcher. In the literature, best EER performance of 14.84% is reported in [43] using a 380 × 102 LPIDB database. In this study, EER observed for FM6 in 1457 × 2296 dataset is 4.317%. Rank-1 identification rate of 79.4% was observed in a comparison of 446 latent vs 12489 full palmprints in [44] while achieving average matching time 141 ms for genuine and 50 ms for impostor attempts. Average latent palmprint matching time in Ruiz et al.'s survey [37] is specified as minimum 200 ms and maximum 4.5 s. Rank-1 for FM23/3/6 is 82.7/83.94/85.45% for 1457 latent against 2296 × 6 sensor dataset. FM23/3/6 average matching time per palm latent against 6 palm sensor data of the same person is 1.64/0.68/3.3 ms in measurements.

Table 8. FM 1457 × 2296 palm latent vs palm sensor / finger latent vs finger slap roll measurements.

Matcher	Rank-1 (%)	Rank-20 (%)	EER (%)	FMR1000 (%)	Memory (GB)	Match per second
Base	62.32 / 79.75	72.55 / 89.43	17.583 / 7.635	40.494 / 23.472	2.01 / 0.8	9096
FM23	82.7 / 89.02	91.56 / 94.78	6.147 / 4.703	32.669 / 12.971	15.95 / 2.13	610
FM3	83.94 / 89.84	93.0 / 94.23	5.294 / 5.784	31.434 / 11.53	17.14 / 3.36	1464
FM6	85.45 / 91.63	93.75 / 96.02	4.317 / 4.031	31.297 / 9.951	46.47 / 7.6	303

4. Discussion

Multistep fusion introduced in this work boosts base matchers' rank-1 identification rate from 59.26% to 87.64% for 5560 latent against 100K rolled-slap background in a private dataset. When compared to the performance of some recent algorithms reported in [28] and [24] on 5560 latent-roll (one-to-one) dataset, FM6/FM6-MRF yields 91.92/94.92% rank-1 accuracy, while MinNet [28] and Cao et al. [24] achieves 92.39% and 85.88% rank-1 accuracy, respectively. These results clearly show the superiority of the proposed FM6-MRF method when compared to existing machine learning (i.e. MCC) and deep-learning-based matching methods. Similarly, FM6 MRF improves average base rank-1 value from 595.14 to 1180 in NIST SD302 10K latent dataset against 7 different sensor dataset with 100K background. FM is developed using a private latent-to-slap/rolled dataset and applied for latent-to-latent identification, latent palmprint identification and exposed to NIST SD302 dataset without any modifications. Algorithm seems not to have any overfitting problems and performs well for different datasets and problem domains.

NIST SD302 dataset includes maximum 4397 usable latent in 10K latent image set with automatically generated minutiae. Measurements show that FM6 with MRF (1415) outperforms NIST's matcher (880) by 60.8% in autogenerated latent with baseline sensor rolled data use against 100K rolled background. In a recent study [45], 1081 rank-1 accuracy was observed for 10K NIST SD302 latent against the 63K rolled background, which is considerably lower when compared with 1415 FM6 MRF rank-1 with 100K rolled background.

The FM6 MRF identification count (911) for 5950 latent image (subset of 10K) with 2872 auto minutia file using baseline rolled sensor data against 100K rolled background is also 15.32% better than NIST referenced COTS (790). For the same dataset, if slap and rolled sensor data is used together, the FM6 MRF identification count becomes 1022. Using CLPE marks with FM6 instead of autogenerated minutia set with FM6 MRF improves rank-1 rate from 1022 to 1265. FM's identification rates would be improved if it will combined with a minutia extractor that would perform close to CLPE performance. Using CLPE marks in NIST SD302 dataset to train neural net-based minutia extractors would enhance minutiae extraction performance.

Observations for 1M background with NIST SD302 dataset shows that identification rates follows a predictable decrease over the logarithmic scale of the background axis. Trendlines for 100M background show that the difference between marked and autogenerated minutia identification rates gets reduced if multiresolution fusion applied as the background data size is enlarged. Prediction for 292 expert selected latent against 69M is used to make FM comparison with experts acceptance criteria. FM23/3/6 MRF 69M background rank-20 prediction based on observations for 1000K background is 84% while experts acceptance criteria is 78.7%. Although predictions according to 1 million observation is useful, it would be better to enlarge background data set to 10 million to get better predictions.

Latent-to-latent identification rates have been observed lower than latent-to-sensor identification rates, as expected since latents mostly contain partial regions of full fingerprint and two latent impression potentially

may not overlap. However, latent background database used for search would be much less than nationwide slap-rolled database, and slower version, but more accurate solution FM8 has been applicable. FM6/8 MRF rank-1 is 44.32/46.12% for 722 individualized latent against 10K latent background.

FM shall be used as a palmprint matcher if palmprint images are segmented as explained in Section 2.3 before the matching process. Performance figures of the proposed palmprint matcher is remarkably good when compared to the results in the literature. FM3 have average 0.68 ms time for latent palmprint to full (6 image per person) palm compare and 83.94 rank-1 for 1457 latent against 2296 palmprint sensor using AMD EPYC 7H12 64-Core and 128 thread. Palm matching speed and accuracy would be improved by improving segmentation methods used and needs further study. Palm region detection of latent and left/right-hand-based search on sensor data would be helpful to improve performance as well. A more refined palmprint matcher algorithm tailored for palmprints would certainly enhance speed and accuracy by using more minutiae and segment counts in palmprints than fingerprints.

FM23/3/6 and MRF's are slower by 7.55/6.61/16.6 and 23.93/14.57/46.11 times than the base matcher executions on a private dataset of 5560 latent and 100K slap and rolled background. FM23 and FM3 perform similar accuracy behavior, while FM23 fuses 290/1 and FM3 fuses 76/10 source/target combinations. FM23 best fits for fast search on latent to very large slap/rolled background, which maps a new criminal latent search against existing nationwide database. FM3 best fits for fast slap/rolled search against large latent databases, which is usable when new slap/rolled data added to criminal office and searched against all latents of existing unsolved cases. FM6 suits well for all use cases if time is not critical or hardware resources are sufficient. Performance of slap/rolled against latent search is not presented in this study, but identification performances and speed measures will be the same with latent-to-slap/rolled measures, and memory usage shall be estimated by formulas presented in document, which would be correlated with source and target fusion item counts.

Private fingerprint and palmprint latents are based on solved cases by using manually marked minutiae, using more than one latent, or using other clues in a criminal investigation. For this reason, identification rates for private dataset have been observed higher when compared to NIST SD302 dataset. However, NIST SD302 dataset results show probable performance figures that AFIS operators will encounter on a criminal investigation.

5. Conclusion

The motivation behind this work is to improve the base matcher to be used in the latent recognition task. Since the base matcher is fast enough and lightweight, it is suitable to fuse scale, rotation, and quality combinations with multiresolution input. The result is several FM configurations applicable for large scale latent applications covering latent-to-slap/rolled, slap/rolled-to-latent, latent-to-latent, and palmprint latent-to-palmprint search requirements over large background databases. Base matcher overview, design details of fusion matcher, and performance characteristics of configurations are clearly described with metrics measured on several public/private datasets, including 5560 latent with one million background slap-rolled database. Trendlines of identification rates for 100 million background are also constructed by using observations on 1 million background.

NIST SD27 was the widely referenced public latent database in the literature and has been served nearly 20 years. It has been replaced by new NIST SD302, and replacement dataset has been publicly available since December 2021. Fusion matcher presents highest rank-1 rate by outperforming NIST referenced matcher by 60.8% and NIST referenced COTS by 15.32% on a standard NIST SD302 latent and sensor dataset over the

same sized and typed background database used by references.

FM algorithm has been implemented in standard C++ as a library and executable which is part of a biometric SDK (software development kit). FM implementation does not depend on any third party library and provides applicable and practical framework with adequate speed and accuracy for large scale AFIS/APFIS applications. Although notable improvement is achieved by the fusion, there is still a room for further improvement in speed and accuracy in the latent identification domain. FM performance measures with further public and private datasets, fusion with multiple minutia extractors, fusion with other matchers, optimization in speed and accuracy are ongoing work, and outputs will be published in the future.

Acknowledgement

This work is done as part of AFIS project which is supported by HAVELSAN Inc. We would like to thank all the supporters, customers, data suppliers, AFIS staff and those who contributed to the study and wish success to those who will manage, analyze, design, develop, test, and deploy end user applications.

I.K. fully designed and implemented multistep fusion matcher and palm matcher as a software library and executable including all verification applications, implemented and optimized base matcher in C++ by using first version written in Python and added bitwise MCC implementation from scratch, analyzed, designed, and performed all public and private data fusion matcher measurements, and drafted majority of the paper.

Y.A. implemented first version of base matcher in Python and, with his team members, contributed to the development of optimized base matcher, drafted majority of the introduction, designed the flow of the paper, managed/worked on private 5560 latent dataset preparation, made the document more understandable and academic, and made majority of the literature survey.

E.B. managed/worked for data collection, managed/worked on preprocessing part, and feature extraction which inputs data to fusion matcher, improved base matcher speed around 10%, and revised the paper.

References

[1] Maltoni D, Maio D, Jain AK, Prabhakar S. Handbook of Fingerprint Recognition. Second Ed, Springer Publishing Company, Incorporated, 2009.

[2] Singh M, Singh R, Ross A. A Comprehensive Overview of Biometric Fusion. Information Fusion 2019; 52:187-205. https://doi.org/10.1016/j.inffus.2018.12.003

[3] Lee HC, Ramotowski R, Gaensslen RE. Advances in Fingerprint Technology. Second Ed, CRC Press, New York, 2001. https://doi.org/10.1201/9781420041347

[4] Kayaoglu M, Topcu B, Uludag U. Standard Fingerprint Databases: Manual Minutiae Labeling and Matcher Performance Analyses. arXiv:1305.1443, 2013. https://doi.org/10.48550/arXiv.1305.1443

[5] Feng J, Yoon S, Jain AK. Latent Fingerprint Matching: Fusion of Rolled and Plain Fingerprints. In: Advances in Biometrics, Third International Conference, ICB 2009; Alghero, Italy; 2009. pp 695–704.

[6] Fiumara G, Schwarz M, Heising J, Peterson J, Flanagan P et al. NIST Special Database 302: Supplemental Release of Latent Annotations, Technical Note. National Institute of Standards and Technology, Gaithersburg, MD, 2021. https://doi.org/10.6028/NIST.TN.2190

[7] Garris MD. Latent Fingerprint Training with NIST Special Database 27 and Universal Latent Workstation. NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, 2001.

[8] Sankaran A, Vatsa M, Singh R. Hierarchical Fusion for Matching Simultaneous Latent Fingerprint. In: 2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS); Arlington, VA, USA; 2012, pp. 377-382.

[9] Indovina M, Dvornychenko V, Hicklin R, Kiebuzinski G. ELFT-EFS Evaluation of Latent Fingerprint Technologies: Extended Feature Sets [Evaluation no.2], NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, 2021. https://doi.org/10.6028/NIST.IR.7859

[10] Ratha NK, Karu K, Chen S, Jain AK. A Real-Time Matching System for Large Fingerprint Databases. IEEE Transactions on Pattern Analysis and Machine Intelligence 1996; 18 (8): 799-813. https://doi.org/10.1109/34.531800

[11] Chang SH, Cheng FH, Hsu WH, Wu GZ. Fast Algorithm for Point Pattern Matching: Invariant to Translations, Rotations and Scale Changes. Pattern Recognition 1997; 30 (2): 311-320. https://doi.org/10.1016/S0031-3203(96)00076-3

[12] Cappelli R, Ferrara M, Maltoni D. Minutia Cylinder-Code: A New Representation and Matching Technique for Fingerprint Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 2010; 32 (12): 2128-2141. https://doi.org/10.1109/TPAMI.2010.52

[13] Medina-Perez MA, Garcia-Borroto M, Gutierrez-Rodriguez AE, Altamirano-Robles L. Robust Fingerprint Verification Using M-Triplets. In: 2011 International Conference on Hand-Based Biometrics; Hong Kong, China; pp. 1-5.

[14] Jain AK, Feng J. Latent Fingerprint Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 2011; 33 (1): 88–100. https://10.1109/TPAMI.2010.59

[15] Medina-Perez MA, Garcia-Borroto M, Gutierrez-Rodriguez AE, Altamirano-Robles L. Improving Fingerprint Verification Using M-Triplets. Sensors 2012; 12 (3): 3418-3437. https://doi.org/10.3390/s120303418

[16] Jain AK, Feng J, Nandakumar K. On Matching Latent Fingerprints. In: IEEE 2008 Computer Society Conference on Computer Vision and Pattern Recognition Workshops; Anchorage, AK, USA; 2008. pp. 1-8. https://doi.org/10.1109/CVPRW.2008.4563117

[17] Paulino AA, Feng J, Jain AK. Latent fingerprint matching using descriptor-based Hough Transform. IEEE Transactions on Information Forensics and Security 2013; 8 (1): 31-45. https://doi.org/10.1109/TIFS.2012.2223678

[18] Yoon S, Feng J, Jain AK. Latent Fingerprint Enhancement via Robust Orientation Field Estimation. In: 2011 International Joint Conference on Biometrics (IJCB); Washington, DC, USA, 2011, pp. 1-8. https://doi.org/10.1109/IJCB.2011.6117482

[19] Baig WU, Munir U, Ellahi W, Ejaz A, Sardar K. Minutia Texture Cylinder Codes for Fingerprint Matching. In: 2019 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan; 2019. pp. 77-775. https://doi.org/10.1109/FIT47737.2019.00024

[20] Medina-Perez MA, Moreno AM, Ballester MAF, Garcia-Borroto M, Loyola-Gonzalez O et al. Latent Fingerprint Identification using Deformable Minutiae Clustering. Neurocomputing 2016; 175 (B): 851–865. https://doi.org/10.1016/j.neucom.2015.05.130

[21] Gutierrez PD, Lastra M, Herrera F, Benitez JM. A High Performance Fingerprint Matching System for Large Databases Based on GPU. IEEE Transactions on Information Forensics and Security 2014; 9 (1): 62–71. https://doi.org/10.1109/TIFS.2013.2291220

[22] Cappelli R, Ferrara M, Maltoni D. Large-scale fingerprint identification on GPU. Information Sciences 2015; 306: 1–20. https://doi.org/10.1016/j.ins.2015.02.016

[23] Cao K, Jain AK. Automated Latent Fingerprint Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 2019; 41 (4): 788–800. https://doi.org/10.1109/TPAMI.2018.2818162

[24] Cao K, Nguyen DL, Tymoszek C, Jain AK. End-to-end latent fingerprint search. IEEE Transactions on Information Forensics and Security 2020; 15:880–894. https://doi.org/10.1109/TIFS.2019.2930487

[25] Deshpande UU, Malemath VS, Patil SM, Chaugule SV. End-to-End Automated Latent Fingerprint Identification With Improved DCNN-FFT Enhancement. Frontiers in Robotics and AI 2020; 7:594412. https://doi.org/10.3389/frobt.2020.594412

[26] Ezeobiejesi J, Bhanu B. Latent Fingerprint Image Quality Assessment Using Deep Learning. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); Salt Lake City, UT, USA; 2018. pp. 621-629. https://doi.org/10.1109/CVPRW.2018.00092

[27] Tang Y, Gao F, Feng J, Liu Y. FingerNet: An Unified Deep Network for Fingerprint Minutiae Extraction. In: 2017 IEEE International Joint Conference on Biometrics (IJCB); Denver, CO, USA; 2017. pp. 108-116. https://doi.org/10.1109/BTAS.2017.8272688

[28] Ozturk H, Selbes B, Artan Y. MinNet: Minutia Embedding Network for Automatic Latent Recognition. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); New Orleans, LA, USA; 2022. pp. 1626-1634. https://doi.org/10.1109/10.1109/CVPRW56347.2022.00169

[29] Ko K. User's Guide to NIST Biometric Image Software (NBIS). National Institute of Standards and Technology, Gaithersburg, MD, 2007. https://doi.org/10.6028/NIST.IR.7392

[30] Engelsma JJ, Cao K , Jain AK. Learning a Fixed-Length Fingerprint Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 2021; 43 (6): 1981-1997. https://doi.org/10.1109/TPAMI.2019.2961349

[31] Tulyakov S, Sankaran N, Mohan D, Setlur S, Govindaraju V. Multistage Fusion of Face Matchers. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); Nashville, TN, USA; pp. 1444-1452, 2021. https://doi.org/10.1109/CVPRW53098.2021.00160

[32] Dvornychenko VN. Evaluation of Fusion Methods for Latent Fingerprint Matchers. 2012 5th IAPR International Conference on Biometrics (ICB); New Delhi, India; 2012. pp. 182-188. https://doi.org/10.1109/ICB.2012.6199806

[33] Marcialis GL, Roli F. Fingerprint Verification by Decision-level Fusion of Optical and Capacitive Sensors. Pattern Recognition Letters 2004; 25 (11): 1315–1322. https://doi.org/10.1016/j.patrec.2004.05.011

[34] Paulino AA, Jain AK, Feng J. Latent Fingerprint Matching: Fusion of Manually Marked and Derived Minutiae. In: 2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images; Gramado, Brazil; 2010. pp. 63–70. https://doi.org/10.1109/SIBGRAPI.2010.17

[35] Ross A, Jain AK, Reisman J. A hybrid Fingerprint Matcher. Pattern Recognition: In: 2002 International Conference on Pattern Recognition; Quebec City, QC, Canada; 2002. pp. 795-798. https://doi.org/10.1109/ICPR.2002.1048138

[36] Dewan SK. Elementary Watson: Scan a palm, find a clue; The New York Times, 21 November 2003.

[37] Rodriguez-Ruiz J, Medina-Perez MA, Monroy R. A Survey on Minutiae-based Palmprint Feature Representations, and a Full Analysis of Palmprint Feature Representation Role in Latent Identification Performance. Expert Systems with Applications 2019; 131: 30-44. https://doi.org/10.1016/j.eswa.2019.04.018

[38] Cappelli R, Maio D, Maltoni D, Wayman JL, Jain AK. Performance Evaluation of Fingerprint Verification Systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 2006; 28 (1): 3-18. https://doi.org/10.1109/TPAMI.2006.20

[39] Maio D, Maltoni D, Cappelli R, Wayman JL , Jain AK. FVC2004: Third Fingerprint Verification Competition. In: Biometric Authentication, First International Conference, ICBA 2004; Hong Kong, China; 2004. pp.1-7.

[40] Ulery BT, Hicklin RA, Roberts MA, Buscaglia J. Measuring What Latent Fingerprint Examiners Consider Sufficient Information for Individualization Determinations. The PLOS ONE Staff 2014; 9 (11): e110179. https://doi.org/10.1371/journal.pone.0110179

[41] Fiumara G, Ko K, Tabassi E, Flanagan PA, Grantham JD et al. Nail to Nail Fingerprint Challenge Enrollment Set Variability. NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, 2019. https://doi.org/10.6028/NIST.IR.8257

[42] Fiumara G, Tabassi E, Flanagan P, Grantham J, Ko K et al. Nail to Nail Fingerprint Challenge: Prize Analysis, NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, 2018. https://doi.org/10.6028/NIST.IR.8210

[43] Morales A, Medina-Perez MA, Ferrer MA, Garcia-Borroto M, Robles LA. LPIDB v1.0-Latent Palmprint Identification Database. In: IEEE International Joint Conference on Biometrics, Clearwater, FL, USA; 2014. pp. 1-6. https://doi.org/10.1109/BTAS.2014.6996268.

[44] Liu E, Jain AK, Tian J. A Coarse to Fine Minutiae-based Latent Palmprint Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 2013; 35 (10): 2307-2322. https://doi.org/10.1109/TPAMI.2013.39

[45] Wyzykowski ABV, Jain AK. Synthetic Latent Fingerprint Generator. In: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV); Waikoloa, HI, USA; 2023. pp. 971-980. https://doi.org/10.1109/WACV56688.2023.00103