ISSN:0975-9646

Protecting Fingerprint Privacy Using Combined Minutiae Template Generation Algorithm

Prabhakara Rao T[1], Gintanjali Sahu[2]

[1] Assistant professor, Aitam college of Engineering, India,
[2] M.Tech student, Aitam college of Engineering, India

Abstract:The primary purpose of using a biometric system is to provide non-reputable authentication. Authentication implies that (i) only legitimate or authorized users are able to access the physical or logical resources protected by the biometric system and (ii) impostors are prevented from accessing the protected resources. While a biometric system can be compromised in a number of ways, one of the potentially damaging attacks is the leakage of biometric template information. The leakage of this template information to unauthorized individuals constitutes a serious security and privacy threat. Therefore in this paper we propose a model of creating a combined minutiae template. By using the combined minutiae template, the complete minutiae feature of a single fingerprint will not be compromised when the database is stolen. The experimental results show that our system can achieve a very low error rate. Compared with the state-of-the-art technique, our work has the advantage in creating a better new virtual identity when the two different fingerprints are randomly chosen.

1. INTRODUCTION

WITH the widespread applications of fingerprint techniques in authentication systems, protecting the privacy of the fingerprint becomes an important issue. Traditional encryption is not sufficient for fingerprint privacy protection because decryption is required before the fingerprint matching, which exposes the fingerprint to the attacker [1]. Therefore, in recent years, significant efforts have been put into developing specific protection techniques for fingerprint.

Most of the existing techniques make use of the key for the fingerprint privacy protection, which creates the inconvenience. They may also be vulnerable when both the key and the protected fingerprint are stolen. Teoh et al. [2] propose a biohashing approach by computing the inner products between the user's fingerprint features and a pseudorandom number (i.e., the key). The accuracy of this approach mainly depends on the key, which is assumed to be never stolen or shared [3]. Ratha et al. [4] propose to generate cancelable fingerprint templates by applying noninvertible transforms on the minutiae. The noninvertible transform is guided by a key, which will usually lead to a reduction in matching accuracy. The work in [2] and [4] are shown to be vulnerable to intrusion and linkage attacks when both the key and the transformed template are stolen [5]. Nandakumar et al. [6] propose to implement fuzzy fault on the minutiae, which is vulnerable to the key-inversion attack [7]. Our work in [8] imperceptibly hide the user identity on the thinned fingerprint using a key. The user identity may also be compromised when both the key and the protected thinned fingerprint are stolen.

In this paper, we propose a novel system for protecting fingerprint privacy by combining two different fingerprints into a new identity. During the enrollment, the system captures two fingerprints from two different fingers. We propose a combined minutiae template generation algorithm to create a combined minutiae template from the two fingerprints. In such a template, the minutiae positions are extracted from one fingerprint, while the minutiae directions depend on the orientation of the other fingerprint and some coding strategies. The template will be stored in a database for the authentication which requires two query fingerprints. A two-stage fingerprint matching process is further proposed for matching the two query fingerprints against a combined minutiae template. By using the combined minutiae template, the complete minutiae feature of a single fingerprint will not be compromised when the database is stolen. In addition,the combined minutiae template share a similar topology to the original minutiae templates, it can be converted into a real-look alike combined fingerprint by using an existing fingerprint reconstruction approach [9]. The combined fingerprint issues a new virtual identity for two different fingerprints, which can be matched using minutiae based fingerprint matching algorithms.

The rest of the paper is organized as, Section 2 discuss proposed fingerprint privacy system. Section 3 explains how to generate combined fingerprint from two different fingerprints. Section 4 presents experimental results. Finally, Section 5 concludes the paper.

2. FINGERPRINT PRIVACY SYSTEM

In the enrollment phase, the system captures two fingerprints from two different fingers, say fingerprints A and B from fingers A and B respectively. We extract the minutiae positions from fingerprint A and the orientation from fingerprint B using some existing techniques [10], [11]. Then, by using our proposed coding strategies, a combined minutiae template is generated based on the minutiae positions, the orientation and the reference points detected from both fingerprints. Finally, the combined minutiae template is stored in a database. In the authentication phase, two query fingerprints are required from the same two fingers, say fingerprints A' and B' and from fingers A and B. As what we have done in the enrolment, we extract the minutiae positions from fingerprint A' and the orientation from fingerprint B. Reference points are detected from both query fingerprints. These extracted information will be matched against the corresponding template stored in the database by using a two-stage fingerprint matching. The authentication will be successful if the matching score is over a predefined threshold.

2.1 Reference Points Detection

The reference points detection process is motivated by Nilsson et al. [12], who first propose to use complex filters for singular point detection. Given a fingerprint, the main steps of the reference points detection are summarized as follows:

1. Compute the orientation O from the fingerprint using the orientation estimation algorithm proposed in [11]. Obtain the orientation in Z complex domain, where

$$Z = \cos(2O) + \mathrm{j}\sin(2O) \qquad (1)$$

2. Calculate a certainty map of reference points

$$C_{ref} = Z * \overline{T_{ref}} \qquad (2)$$

where " *" is the convolution operator and $\overline{T_{ref}}$ is the conjugate of

$$T_{ref} = (x+iy)\frac{1}{2\pi\sigma^2}\exp(-\frac{x^2+y^2}{2\sigma^2}) \qquad (3)$$

3. Calculate an improved certainty map

$$C'_{ref} = \begin{cases} C_{ref}.\sin(Arg(C_{ref}))ifArg(C_{ref}) > 0 \\ 0otherwise \end{cases} \qquad (4)$$

where Arg(Z) returns the principal value of the argument of (Z)

4. Locate a reference point satisfying the two criterions: (i) the amplitude of C'_{ref} the point (hereinafter termed as the certainty value for simplicity) is a local maximum, and (ii) the local maximum should be over a fixed threshold T. Suppose we locate a reference point at ($r_{x,}r_y$), the corresponding angle can be estimated as Arg($C'_{ref}(r_x, r_y)$).
5. Repeat step 4) until all reference points are located
6. If no reference point is found for the fingerprint in steps 4) and 5) (e.g., an arch fingerprint), locate a reference point with the maximum certainty value in the whole fingerprint image.

2.2 Combined Minutiae Template Generation

Given a set of N minutiae positions P_A={P_a=(x_{ia},y_{ia}), 1≤i≤N}of fingerprint, the orientation of fingerprint A and the reference points of fingerprints and , a combined minutiae template is generated by minutiae position alignment and minutiae direction assignment.

1. *Minutiae Position Alignment:* Among all the reference points of a fingerprint for enrolment, we define a reference point with the maximum certainty value as the primary reference point. Therefore, we have two primary reference points R_a and R_b for fingerprints A and B. Let's assume R_a is located at r_a=(r_{xa}, r_{ya}) with the angle β_a, and R_b is located at r_b=(r_{xb}, r_{yb}) with the angle β_b. The alignment is performed by translating and rotating each minutiae point $P_{ia\ to}$ P_{ic}=(x_{ic}, y_{ic}) by

$$(P_{ic})^T = H.(P_{ia} - r_a)^T + (r_b)^T \qquad (5)$$

where ( )^T is the transpose operator and H is the rotation matrix

$$H = \begin{bmatrix} \cos(\beta_b - \beta_a), & \sin(\beta_b - \beta_a) \\ -\sin(\beta_b - \beta_a) & \cos(\beta_b - \beta_a) \end{bmatrix} \qquad (6)$$

As such R_a, and R_b are overlapped both in the position and the angle after the minutiae position alignment.

Minutiae Direction Assignment: Each aligned minutiae position P_{ic} is assigned with a direction θ_{ic} as follows:

$$\theta_{ic} = O_B(x_{ic}, y_{ic}) + \rho_i\pi \qquad (7)$$

where ρ_i is an integer that is either 0 or 1. The range of O_B(x_{ic}, y_{ic}) is from 0 to π. Therefore, the range of θ_{ic} will be from 0 to 2π, which is the same as that of the minutiae directions from an original fingerprint.

Sometimes, P_{ic} may be located outside the area of fingerprint B, where O_B(x_{ic}, y_{ic}) is not well defined. In such a case, we need to predict O_B(x_{ic}, y_{ic}) before the direction assignment. Some existing works for modeling the fingerprint orientation can be adopted to do the prediction. For example, the work in [15] can estimate the missing orientation structure even for a partial fingerprint. Here, we simply predict the value of O_B(x_{ic}, y_{ic}) (if it is not well defined) as the value of nearest well defined orientation in O_B.

Once all the N aligned minutiae positions are assigned with directions, a combined minutiae template M_C={m_{ic}=(P_{ic},θ_{ic}), 1≤i≤N} is created for enrolment. In some cases, a global minutiae position translation may be necessary for M_C such that all the minutiae points are located inside the fingerprint image.

Two-Stage Fingerprint Matching:

Given the minutiae positions P_A of fingerprint A', the orientation O_B of fingerprint B' and the reference points of the two query fingerprints. In order to match M_C the stored in the database, we propose a two-stage fingerprint matching process including query minutiae determination and matching score calculation.

Query Minutiae Determination*:* The query minutiae determination is a very important step during the fingerprint matching. In order to simplify the description of our algorithm, we first introduce the local features extracted for a minutiae point in M_C. The local feature extraction is similar to the work proposed in [14]. Given a minutiae point m_{ic} and another minutiae point m_{jc} in M_C, we define

1) L_{ij} as the distance between m_{ic} and m_{jc}:

$$L_{ij} = \sqrt{(x_{ic} - x_{jc})^2 + (y_{ic} - y_{jc})^2} \qquad (11)$$

2) γ_{ij} as the difference between the directions (after modulo π) of m_{ic} and m_{jc}

$$\gamma_{ij} = \theta_{ic} \bmod \pi - \theta_{jc} \bmod \pi \qquad (12)$$

3) σ_{ij} as a radial angle

$$\sigma_{ij} = \Re(\theta_{ic} \bmod \pi, a\tan 2(y_{jc} - y_{ic}, x_{jc} - x_{ic}) \qquad (13)$$

where atan2(y,x) is a two-argument arctangent function in the range (-π,π) and

$$\Re(\mu_1, \mu_2) = \begin{cases} \mu_1 - \mu_2 if -\pi < \mu_1 < \mu_2 \le \pi \\ \mu_1 - \mu_2 + 2\pi if\, \mu_1 - \mu_2 \le -\pi \\ \mu_2 - \mu_1 + 2\pi if\, \mu_1 - \mu_2 > \pi \end{cases} \quad (14)$$

4) Suppose Fu are the local features extracted for the uth minutiae point in Mc(T) , while Fv are the local features extracted for the vth minutiae point in Mc. Calculate the difference between Fu and Fv by.

$$D_r(u,v) = w_1 \sum_{j=1}^{3} | F_u(j) - F_v(j) | + w_2 \sum_{j=4}^{9} | F_u(j) - F_v(j) | \quad (15)$$

where $F_i(j)$ refers to the jth component of F_i, w_1 and w_2 are the weights for different features. We follow the same weight settings as in [13], where w_1 and w_2 are empirically

set as w_1 =1 and $w_2 = 0.3 * {}^{180}\!/_{\pi}$ Then, we define the difference between Mc(T) and Mc as

$$d_r = \min_{u,v} D_r(u,v) \quad (16)$$

5) Repeat steps 1) to 4) until all the possible pairs of reference points are selected and processed

Matching Score Calculation: For the combined minutiae templates that are generated using *Coding Strategy* 1, we do a modulo π for all the minutiae directions in M_Q and M_C, so as to remove the randomness. After the modulo operation, we use an existing minutiae matching algorithm [10] to calculate a matching score between M_Q and M_C for the authentication decision. For other combined minutiae templates, we directly calculate a matching score between M_Q and M_C using an existing minutiae matching algorithm.

3. Combined Fingerprint Generation

In a combined minutiae template, the minutiae positions and directions (after modulo) are extracted from two different fingerprints separately. These minutiae positions and directions share a similar topology to those from an original fingerprint. Therefore, the combined minutiae template has a similar topology to an original minutiae template. Some existing works [9], [13], [14] have shown that it is possible to reconstruct a full fingerprint image from a minutiae template. By adopting one of these fingerprint reconstruction approaches, we are able to convert our combined minutiae template into a combined fingerprint image. Given any two different fingerprints as input, we first generate a combined minutiae template using our combined minutiae template generation algorithm. Then, a combined fingerprint is reconstructed from the combined minutiae template using one of the existing fingerprint reconstruction approaches.

It should be noted that the combined minutiae template generated by adopting Coding Strategy 1 is not appropriate for generating a combined fingerprint. The reason is that we set ρ_i as 0 or 1 randomly during the minutiae direction assignment, i.e., we add π randomly for each minutiae direction in such a coding strategy. we need to perform a modulo operation for the minutiae directions during the fingerprint matching, so as to remove such randomness. Therefore, we will not be able to match the corresponding combined fingerprint by using a general fingerprint matching algorithm. While the purpose of generating a combined fingerprint is to issue a new virtual identity for two different fingerprints, which should be matched using general fingerprint matching algorithms.

Among the existing fingerprint reconstruction approaches, our previous work [9] achieves excellent performance. We here adopt this approach for generating a combined fingerprint from a combined minutiae template. However, the work in [9] does not incorporate a noising and rendering step to make the reconstructed fingerprint image real-look alike. To create a real-look alike fingerprint image from a set of minutiae points, we further apply a noising and rendering step after adopting the work in [9].

4. Experimental Results

The experiment is conducted on the first two impressions of the FVC2002 DB2_A database, which contains 200 fingerprints from 100 fingers (with 2 impressions per finger). The VeriFinger 6.3 [16] is used for the minutiae positions extraction and the minutiae matching. The algorithm proposed in [11] is used for the orientation extraction.

In order to evaluate the performance of our system, we randomly pair the 100 fingers in the FVC2002 DB2_A database to produce a group of 50 non-overlapped finger pairs, where each finger pair contains two different fingers. The random pairing process is repeated 10 times to have 10 groups of 50 non-overlapped finger pairs. For the two fingerprints captured from two different fingers, we can generate two combined minutiae templates in total, where one fingerprint serves as fingerprint B, the other serves as fingerprint or vice versa. The system designer can choose to enrol one or both of the two templates in the database, which depends on the applications. Thus, we consider the following two cases in building the system database for each group of finger pairs: The first impressions of each finger pair are used to produce only one combined minutiae template for enrolment. Therefore, there are 50 templates stored in the database. To compute the False Rejection Rate (FRR), the second impressions of a finger pair are matched against the corresponding enrolled template, producing 50 genuine tests. To compute the False Acceptance Rate (FAR), the first impressions of a finger pair are matched against the other 49 enrolled templates,50*49=2450 producing imposter tests The first impressions of each finger pair are used to produce two combined minutiae templates for enrolment. Thus, there are 100 templates stored in the database. Similarly, 100 genuine tests are performed to compute FRR and 100*99= 9900 imposter tests are performed to compute FAR.

Fig. 1 plots the average FRR (at different FAR) computed from the 10 groups of finger pairs for the two cases. We can see that our system performs similarly for the two cases. However, the error rates vary among different coding strategies, where the *Coding Strategy* 1 achieves the lowest error rate with FRR=0.4% for both cases. While the results of using *Coding Strategy* 3 are the worst, with over 1% FRR for both cases.

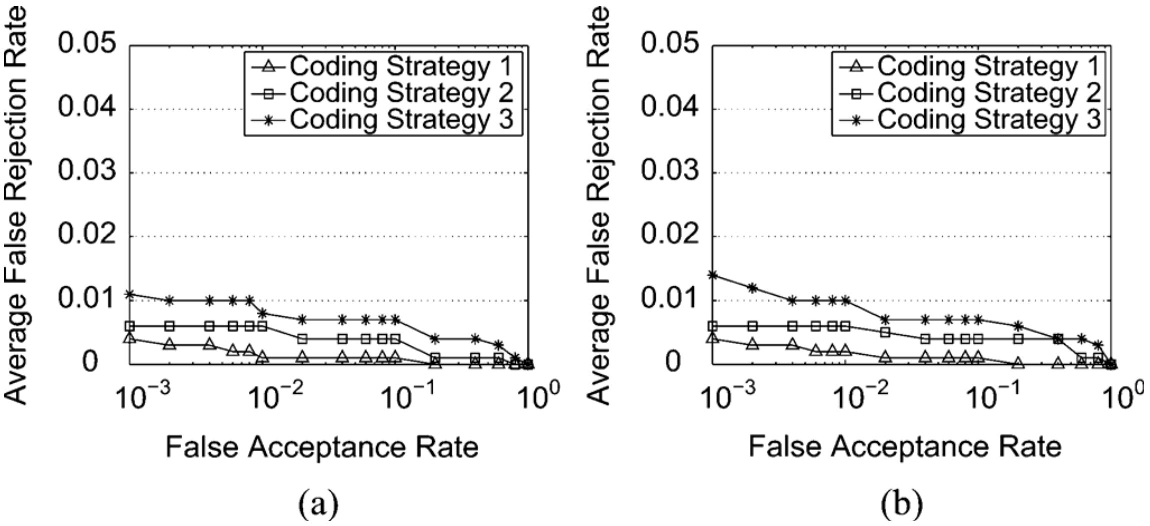

Fig1: performance of our proposed mechanism for a) case 1 b) case 2

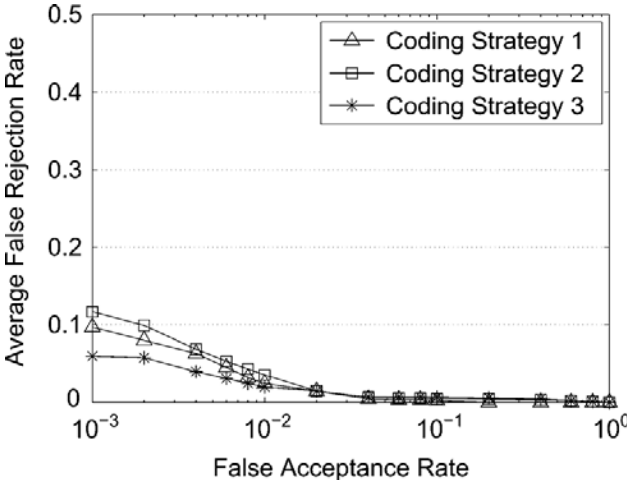

Fig 2: performance of proposed mechanism for possible combined minutiae templates

we examine the difference among all the combined minutiae templates that can be created for a set of 10 fingers based on our proposed system. That is to say, we evaluate the performance of our system when the database stores all the combined minutiae templates generated for 10 fingers. We randomly separate the 100 fingers (in FVC2002 DB2_A) into 10 groups with 10 fingers per group. For each group, there are in total 45 possible finger pairs. The first impressions of each finger pair are used to produce two combined minutiae templates for enrolment. The corresponding second impressions serve as the testing fingerprints. As such, 90 combined minutiae templates are generated and stored in the system database. There are 90 genuine tests for computing FRR and 90*89=8010 imposter tests for computing the FAR for each group, where the average FRR for the ten groups (with 10 fingers per group) is shown in Fig. 2. We can see that the error rates of our system increase a lot because some templates either share the same minutiae positions or the same orientation. Among the different coding strategies, the Coding Strategy 3 achieves the lowest error rates with FRR=6% at FRR=0.1%. While the corresponding FRR of using Coding Strategy 1 and Coding Strategy 2 are 9.67% and 11.67%, respectively.

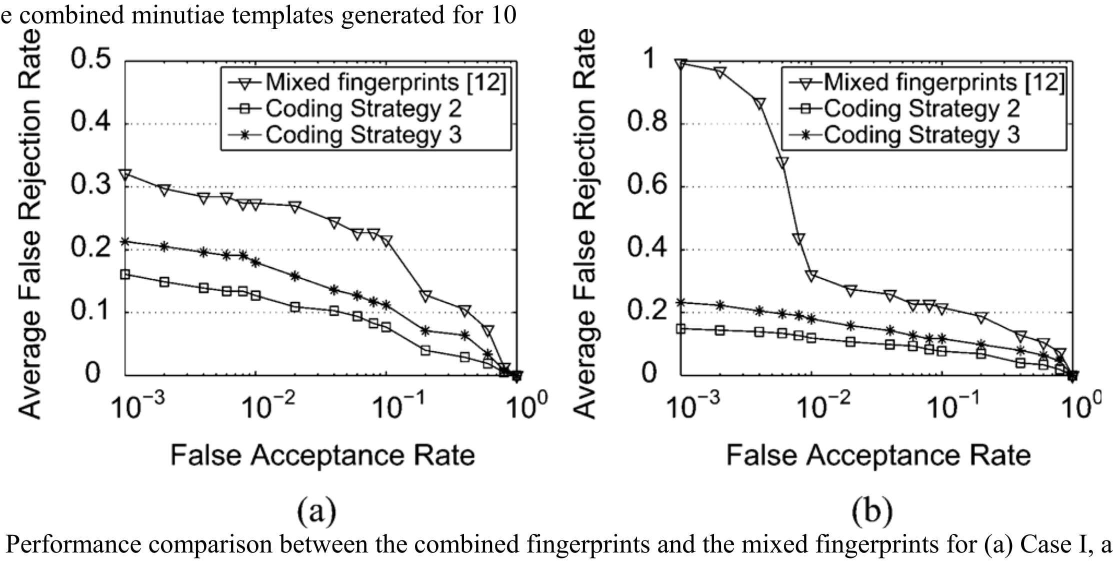

Fig.3 Performance comparison between the combined fingerprints and the mixed fingerprints for (a) Case I, and (b) Case II.

Fig. 3 shows the performance comparison between the combined fingerprints and the mixed fingerprints. It can be seen that our combined fingerprint achieves a lower error rate than the mix fingerprint. Especially for Case II, our work performs much better when the FAR is less than 1%. The combined fingerprints of *Coding Strategy* 2 perform the best, with FRR around 15% at FAR = 0.1% for the two cases.

5. Conclusions

In this paper, we introduce a novel system for fingerprint privacy protection by combining two fingerprints into a new identity. In the enrolment, the system captures two fingerprints from two different fingers. A combined minutiae template containing only a partial minutiae feature of each of the two fingerprints will be generated and stored in a database. To make the combined minutiae template look real as an original minutiae template, three different coding strategies are introduced during the combined minutiae template generation process. In the authentication process, two query fingerprints from the same two fingers are required. A two-stage fingerprint matching process is proposed for matching the two query fingerprints against the enrolled template. Our combined minutiae template has a similar topology to an original minutiae template. Therefore, we are able to combine two different fingerprints into a new virtual identity by reconstructing a real-look alike combined fingerprint from the combined minutiae template. The experimental results show that our system achieves a very low error rate with FRR=0.4% at FAR=0.1% .

References

[1] S. Li and A. C. Kot, “A novel system for fingerprint privacy protection,” in *Proc. 7th Int. Conf. Inform. Assurance and Security (IAS)*, Dec. 5–8, 2011, pp. 262–266.

[2] B. J. A. Teoh, C. L. D. Ngo, and A. Goh, “Biohashing: Two factor authentication featuring fingerprint data and tokenised random number,” *Pattern Recognit.*, vol. 37, no. 11, pp. 2245–2255, 2004.

[3] A. Kong, K.-H. Cheung, D. Zhang, M. Kamel, and J. You, “An analysis of biohashing and its variants,” *Pattern Recognit.*, vol. 39, no. 7, pp. 1359–1368, 2006.

[4] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle, “Generating cancelable fingerprint templates,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 561–72, Apr. 2007.

[5] A. Nagar, K. Nandakumar, and A. K. Jain, “Biometric template transformation: A security analysis,” in *Proc. SPIE, Electron. Imaging, Media Forensics and Security*, San Jose, Jan. 2010.

[6] K. Nandakumar, A. K. Jain, and S. Pankanti, “Fingerprint-based fuzzy vault: Implementation and performance,” *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 4, pp. 744–57, Dec. 2007.

[7] W. J. Scheirer and T. E. Boult, “Cracking fuzzy vaults and biometric encryption,” in *Proc. Biometrics Symp.*, Sep. 2007, pp. 34–39.

[8] S. Li and A. C. Kot, “Privacy protection of fingerprint database,” *IEEE Signal Process. Lett.*, vol. 18, no. 2, pp. 115–118, Feb. 2011.

[10] S. Li and A. C. Kot, “Attack using reconstructed fingerprint,” in *Proc. IEEE Int. Workshop on Inform. Forensics and Security (WIFS)*, Foz do Iguacu, Brazil, Nov. 29–Dec. 2, 2011.

[11] VeriFinger 6.3. [Online]. Available: http://www.neurotechnology.com

[12] L. Hong, Y. F. Wan, and A. Jain, “Fingerprint image enhancement: Algorithm and performance evaluation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 777–789, Aug. 1998.

[13] K. Nilsson and J. Bigun, “Localization of corresponding points in fingerprints by complex filtering,” *Pattern Recognit. Lett.*, vol. 24, no. 13, pp. 2135–2144, 2003.

[14] R. Cappelli, A. Lumini, D. Maio, and D. Maltoni, “Fingerprint image reconstruction from standard templates,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 9, pp. 1489–1503, Sep. 2007.

[15] J. Feng and A. K. Jain, “Fingerprint reconstruction: From minutiae to phase,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 2, pp. 209–223, Feb. 2011.

[16] Y. Wang and J. Hu, “Global ridge orientation modeling for partial fingerprint identification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 72–87, Jan. 2011.