IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 28, 2018, accepted May 11, 2018, date of publication May 28, 2018, date of current version June 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2840330

Cross-Sensor Fingerprint Matching Method Based on Orientation, Gradient, and Gabor-HoG Descriptors With Score Level Fusion

HELALA ALSHEHRI, MUHAMMAD HUSSAIN, HATIM A. ABOALSAMH, (Member, IEEE), AND MANSOUR A. AL ZUAIR

College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

Corresponding authors: Helala AlShehri (hm.shehri@gmail.com) and Muhammad Hussain (mhussain@ksu.edu.sa)

This work was supported by the Research Center of College of Computer and Information Sciences, King Saud University.

ABSTRACT The fingerprint is one of the oldest and most widely used biometric modality for person identification. Existing automatic fingerprint matching systems perform well when the same sensor is used for both enrollment and verification (regular matching). However, their performance significantly deteriorates when different sensors are used (cross-matching, fingerprint sensor interoperability problem). We propose an automatic fingerprint verification method to solve this problem. It was observed that the discriminative characteristics among fingerprints captured with sensors of different technology and interaction types are ridge orientations, minutiae, and local multi-scale ridge structures around minutiae. To encode this information, we propose two minutiae-based descriptors: histograms of gradients obtained using a bank of Gabor filters and binary gradient pattern descriptors, which encode multi-scale local ridge patterns around minutiae. In addition, an orientation descriptor is proposed, which compensates for the spurious and missing minutiae problem. The scores from the three descriptors are fused using a weighted sum rule, which scales each score according to its verification performance. Extensive experiments were conducted using two public domain benchmark databases (FingerPass and Multi-Sensor Optical and Latent Fingerprint) to show the effectiveness of the proposed system. The results showed that the proposed system significantly outperforms the state-of-the-art methods based on minutia cylinder-code (MCC), MCC with scale, VeriFinger—a commercial SDK, and a thin-plate spline model.

INDEX TERMS Biometrics, feature fusion, fingerprint cross-matching, fingerprint sensor-interoperability, score level fusion.

I. INTRODUCTION

Fingerprint is a prevalent biometric modality, which is widely used for person authentication. There has been an intensive research on the development of automatic fingerprint verification methods. However, existing methods were designed to work in the situation when a specific type of sensor is used for both enrollment and verification. Owing to advances in sensing technology, there is a plethora of low-cost and smart fingerprint sensors, which are being embedded in smart devices such as mobile phones and PCs. Matching an individual's fingerprints that originate from different sensors has been an important concern with the growing number of fingerprint applications. Law-enforcement departments, security agencies, and various service providers have developed huge fingerprint databases usually with a specific sensor, but a different type of sensor can be used at the time of authentication and verification. This has given rise to the fingerprint sensor interoperability problem. Fingerprint sensors are based on various technologies such as ultrasound, optical sensors, and solid-state devices [1]. The underlying physics of these technologies incorporate inconsistencies in the captured fingerprints, making the interoperability problem even more challenging.

Recent research has highlighted the need to study the impact of diverse fingerprint sensors in fingerprint-matching methods. Ross and Jain [1] demonstrated that a matching method's performance will decline drastically if a fingerprint is obtained using two different sensors. Lugini *et al.* [2] carried out statistical analysis of the sensor interoperability problem to measure the degree of change in match scores when using different sensors for enrollment and verification.

The literature on automatic fingerprint verification includes a variety of methods that are broadly categorized into minutiae-based, image-based, and hybrid techniques.

While minutiae-based methods primarily depend on minutiae points, image-based methods extract features using techniques such as Gabor filters (GFs) [3]–[5], curvelet transform [6], and radon transform [7]. For image-based fingerprint matching, GFs have been widely applied for two purposes: (1) to enhance fingerprints [8] and (2) to extract Gabor features from filter-bank responses in order to determine the fingerprint quality [9], and for core point detection [10], and matching [3]. GF variants such as curved GFs [11] and orthogonal curved-line GFs [12] have been employed to enhance fingerprints. In view of the success of GFs in fingerprint verification, we employed GFs together with histograms of gradients (HoG) for the description of fingerprints. Hybrid schemes employ either different features or different techniques based on minutiae and images and fuse the information using either feature-level-fusion or score-level-fusion methods [4]–[13]. In addition, image-based fusion has also been employed [14]. Although the most commonly used fusion approach is score-level fusion, to the best of our knowledge, weighted score-level-fusion based on the equal error rate (EER) has not been used.

Only few research efforts have been made to overcome the fingerprint sensor interoperability problem. Such research efforts focused only on non-linear distortion [15], [16], scaling of fingerprints [17]–[19], and fusion of existing fingerprint-recognition methods [20]. The non-linear distortion method [16] is based on modeling the deformation of fingerprint using the thin-plate spline (TPS) model to register a pair of fingerprints from different sensors. Fingerprint scaling methods are used to explore the effect of adding a step for scaling fingerprint using the fingerprint's average inter-ridge distance to compute the scale required to zoom in on two compared fingerprints [17]–[19]. Fusion-based methods explore the effect of combining existing fingerprint systems [20] or fingerprint features based on a classifier [21]. To the best of our knowledge, only a few methods have focused on fusion for sensor interoperability problem [10]–[13]. Alonso-Fernandez *et al.* [20], fused a minutiae-based matcher with a ridge-based matcher using a mean rule. Marasco *et al.* [21] fused some fingerprint characteristics, such as fingerprint quality and average gray-level with match scores using a classifier. In spite of these efforts, fingerprint sensor interoperability is still a challenging problem.

When sensors of different types are used, the fingerprints of the same subject have the same minutiae points and ridge patterns but they differ in details such as local microstructures and texture patterns, which indicate that the discriminative content in cross-sensor fingerprints consists of ridge orientations, minutiae, and local ridge patterns around a minutiae. These observations provide the motivation to explore ridge orientations and multiscale local ridge patterns around a minutia for cross-matching. In view of this, we propose an automatic fingerprint verification method based on three types of descriptors, which employ information about minutiae, multiscale local ridge patterns, and ridge orientations and reduce the effects of cross-matching. The minutiae based descriptors proposed in this paper use binary gradient pattern (BGP) and Gabor-based histograms of gradients (Gabor-HoG) to encode multi-scale local ridge structures and structural changes across ridge patterns around minutiae. To encode ridge orientations, an orientation descriptor is employed. To the best of our knowledge, the BGP-based minutiae descriptor has not yet been applied to the problem of fingerprint recognition or fingerprint sensor interoperability. We argue that BGP is suitable for fingerprint recognition, as it is able to encode ridge structures and structural changes across ridges. The Gabor-HoG descriptor extracts detailed information about local orientations and scales of ridges. Furthermore, the orientation descriptor is based on the computation of dominant ridge orientations, which provide a distinguishing description of the fingerprint ridge patterns and can be computed with acceptable accuracy, even from a noisy input image, as the ridge orientation is invariant to rotation and translation [22]–[24]. These descriptors focus on different characteristics and result in different similarity scores, which are fused according to their verification performance to take the final decision. The proposed method is robust against sensor dependent variations as it encodes local ridge structures at different scales with various orientations. It tolerates spurious and missing minutiae because the orientation descriptor does not depend on minutia information. Small feature extraction errors and local distortion errors are handled by BGP because it is a powerful descriptor that efficiently captures the local structure around minutiae points.

The proposed method has been thoroughly evaluated using benchmark datasets. The results and comparisons show that the proposed method outperforms state-of-the-art methods such as minutiae cylinder-code (MCC), MCC with scale, VeriFinger—a commercial SDK, and thin-plate spline (TPS) model.

The main contributions of this paper are as follows.

- An automatic fingerprint verification method that minimizes the impact of the fingerprint sensor interoperability problem is proposed.
- Two minutiae-based descriptors, which encode multi-scale local ridge patterns around minutiae points, and an orientation descriptor that encodes local ridge patterns and orientation field are presented.
- A fusion scheme that combines scores obtained using three different types of descriptors according to their verification performances is proposed.
- A comprehensive method evaluation using two benchmark datasets and standard evaluation tools is presented and compared with state-of-the-art methods.
- The impacts of different types of sensors are analyzed and recommendations are made for developing a cross-sensor automatic fingerprint verification method.

The remaining sections of this paper are organized as follows. Section II presents the motivation for this work whereas Section III details the proposed method. Section IV presents details of the experiments, results,

and discussion. The conclusions and future work are presented in Section V.

II. MOTIVATION

Previous studies on the problem of fingerprint sensor interoperability attempted to adjust the distortions resulting from the deformations of fingerprints captured using different sensors [15]–[19]. However, due attention has not been given towards developing discriminative features for improving cross-device matching performance; it is still a challenging area of research.

To illustrate this challenge, Fig. 1 shows magnified views of fingerprints of the same finger that were captured using different sensors. Each image depicts a different texture with the same ridge patterns and minutiae points. This indicates that ridge patterns and minutiae are the most evident structural characteristics of a fingerprint [25], which can be used for discrimination in case the fingerprints are captured with different sensors. For extracting features that are robust against cross-sensor variability, it is important to focus on the ridge patterns and minutiae and consider the types of features that adequately capture rotation and scale variations, local microstructures, and distortion patterns.

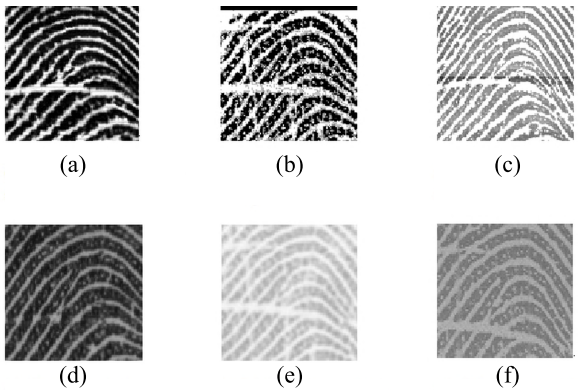

FIGURE 1. Magnified views of fingerprints captured with different sensors. (a) AES2501 optical sweep sensor. (b) AES3400 capacitive sweep sensor. (c) ATRUA capacitive press sensor. (d) FX3000 optical sweep sensor. (e) FPC1011C capacitive press sensor. (f) TCRU2C capacitive press sensor.

Orientation descriptors and Gabor-based histograms of oriented gradients extract discriminative information from ridge patterns and have been shown to give promising results for fingerprint-matching when the same sensor is used for enrolment and query [13]. On the other hand, the binary gradient pattern is a new descriptor, which encodes the local orientation information of an image, and can be employed to encode ridge patterns [26]. Based on the characteristics of these descriptors and our observation of the magnified views shown in Fig. 1, which indicate that for cross-sensor matching, discriminative information can be extracted from ridge patterns and minutiae, we argue that these descriptors can be effective in reducing the impact of sensor interoperability. As a proof of concept, we conducted initial experiments with three descriptors, and analyzed their discriminative potential for cross-sensor matching. After refining the ridge patterns, we encoded multi-scale ridge pattern information around the minutiae points using BGP and Gabor-HoG. In addition, the orientation descriptor was computed based on dominant ridge orientations. We conducted four cross-device matching experiments using multi-sensor optical and latent fingerprint (MOLF) database (the detailed description can be found in Section IV.A). Three experiments were conducted with the orientation descriptor, BGP, and Gabor-HoG using one descriptor for each. The fourth experiment was conducted with three descriptors using score level fusion with sum rule. The results of cross-sensor matching in terms of EER are presented in Table 1. The results indicate that although each descriptor performed relatively better for native matching than cross-sensor matching, the overall performance was poor; the score level fusion showed significant improvement.

Fig. 2 shows the distributions of genuine (in blue) and impostor (in red) scores obtained by cross-sensor matching with each descriptor and the score level fusion when the Lumidigm Venus IP65 Shell sensor is used for enrollment and the Secugen Hamster IV sensor is employed for verification (MOLF database). When using a single descriptor, the intersection region of the genuine and impostor score distributions is very large compared to that of the score distributions corresponding to the score level fusion. This finding is consistent across all pairs of sensors we analyzed.

Our observations are: (1) the discriminative characteristics of fingerprints captured using different sensors are ridge patterns and minutiae, and descriptors such as BGP, Gabor-HoG, and orientation, which encode these characteristics can be effective for cross-sensor matching; (2) the performance of each of the three descriptors (BGP, Gabor-HoG, and orientation) is almost similar for native and cross-sensor matching; (3) none of the descriptors when used alone is effective for cross-sensor matching; and (4) the fusion of the scores obtained by using the three descriptors minutiae result in significant improvement for both native and cross-sensor matching, i.e., the score level fusion minimizes intra-class variance and maximizes inter-class variance. These observations motivated us to further investigate the effectiveness of BGP, Gabor-HoG, and orientation descriptors and score level fusion for the fingerprint sensor interoperability problem.

III. PROPOSED CROSS-SENSOR FINGERPRINT MATCHING SYSTEM

Based on our observations described in the previous section, details of a cross-sensor fingerprint matching method based on three types of descriptors and score level fusion is proposed in this section. An overview of the system is shown in Fig. 3. In the enrollment phase, a gallery fingerprint is first enhanced and then processed to extract minutiae points. Next, three different descriptors are extracted and the template is stored in a database.

During the verification phase, the probe fingerprint is enhanced and processed to extract minutiae points. Then, the probe and template fingerprints are aligned to find

TABLE 1. Performance of (a) BGP (b) Gabor-HoG (c) orientation descriptors, and (d) score level fusion in terms of EER on MOLF database.

Template → Probe	(a)BGP Descriptor			(b)Gabor-HoG Descriptor			(c)Orientation Descriptor			(d)Fusion		
	DB1	DB2	DB3	DB1	DB2	DB3	DB1	DB2	DB3	DB1	DB2	DB3
DB1	**4.806**	6.656	6.677	**3.795**	7.693	5.163	**3.275**	4.84	7.605	**0.721**	2.085	1.344
DB2	6.656	**3.278**	7.697	7.693	**4.387**	5.836	4.84	**3.44**	7.479	2.085	**0.70**	1.28
DB3	6.677	7.697	**4.16**	5.163	5.836	**3.673**	7.605	7.479	**4.010**	1.34	1.28	**0.68**

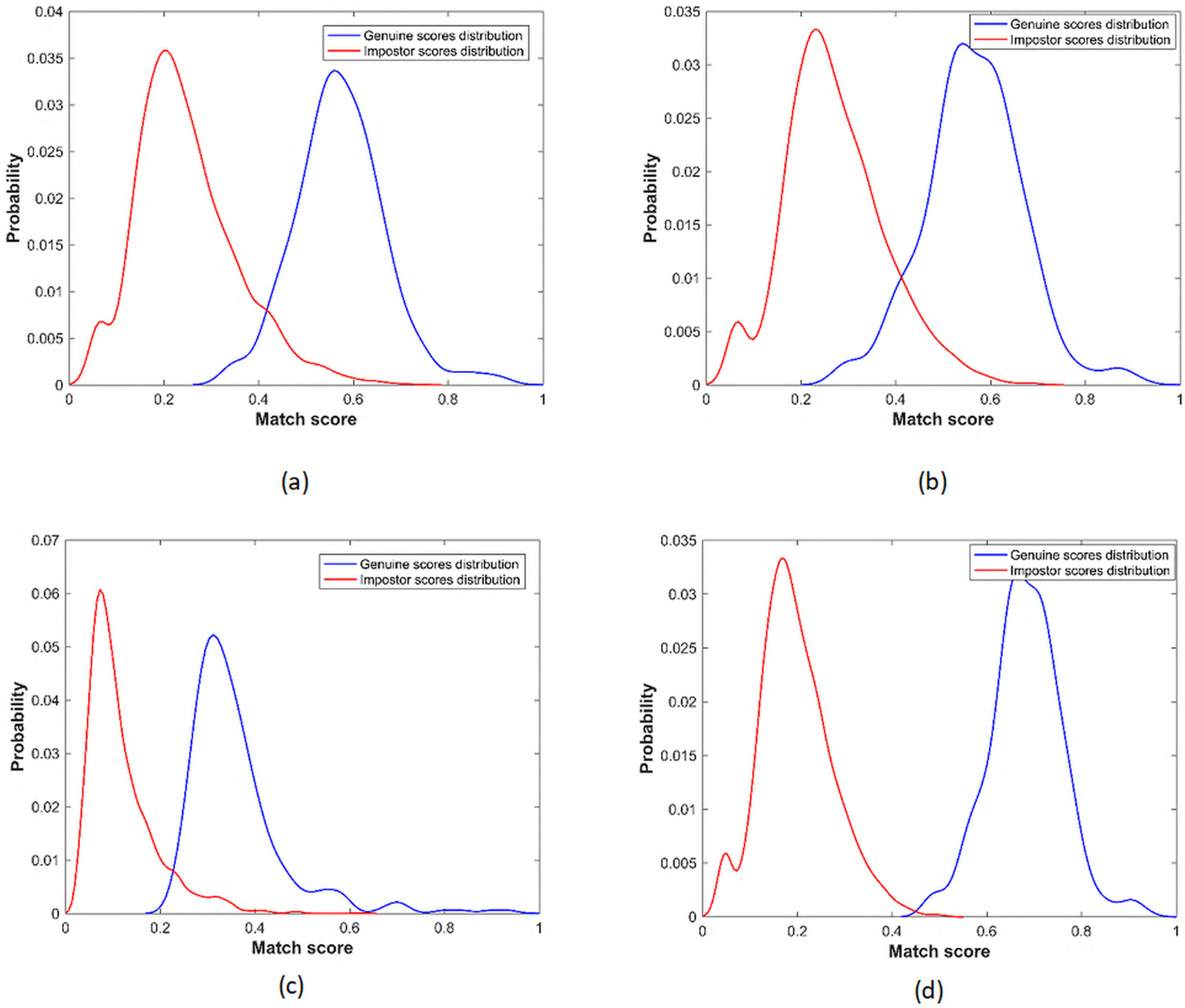

FIGURE 2. Distributions of genuine and imposter scores obtained by cross-sensor matching using (a) BGP, (b) Gabor-HoG, (c) orientation descriptors, and (d) score level fusion when Lumidigm Venus IP65 Shell sensor is used for enrollment and Secugen Hamster IV sensor (MOLF database).

minutiae correspondences, and three different types of features are extracted from the probe fingerprint. The matching process is performed separately for each descriptor by computing the similarities between the extracted features. Finally, the matching scores resulting from the three descriptors are fused to produce the final matching score. In the following subsections, details of different algorithms used for enhancement, minutia extraction, feature extraction, and matching are presented.

A. ENHANCEMENT AND MINUTIAE EXTRACTION

As observed in Fig. 1, the discriminative information consists of ridge patterns and minutiae. To emphasize these characteristics and suppress irrelevant details, fingerprints are preprocessed using a method based on short-time Fourier transform (STFT) [27], which enhances the furrows and ridge structures. In this method, a fingerprint is first divided into small overlapping windows, and STFT is applied to each window. The ridge frequency, ridge orientation, and block

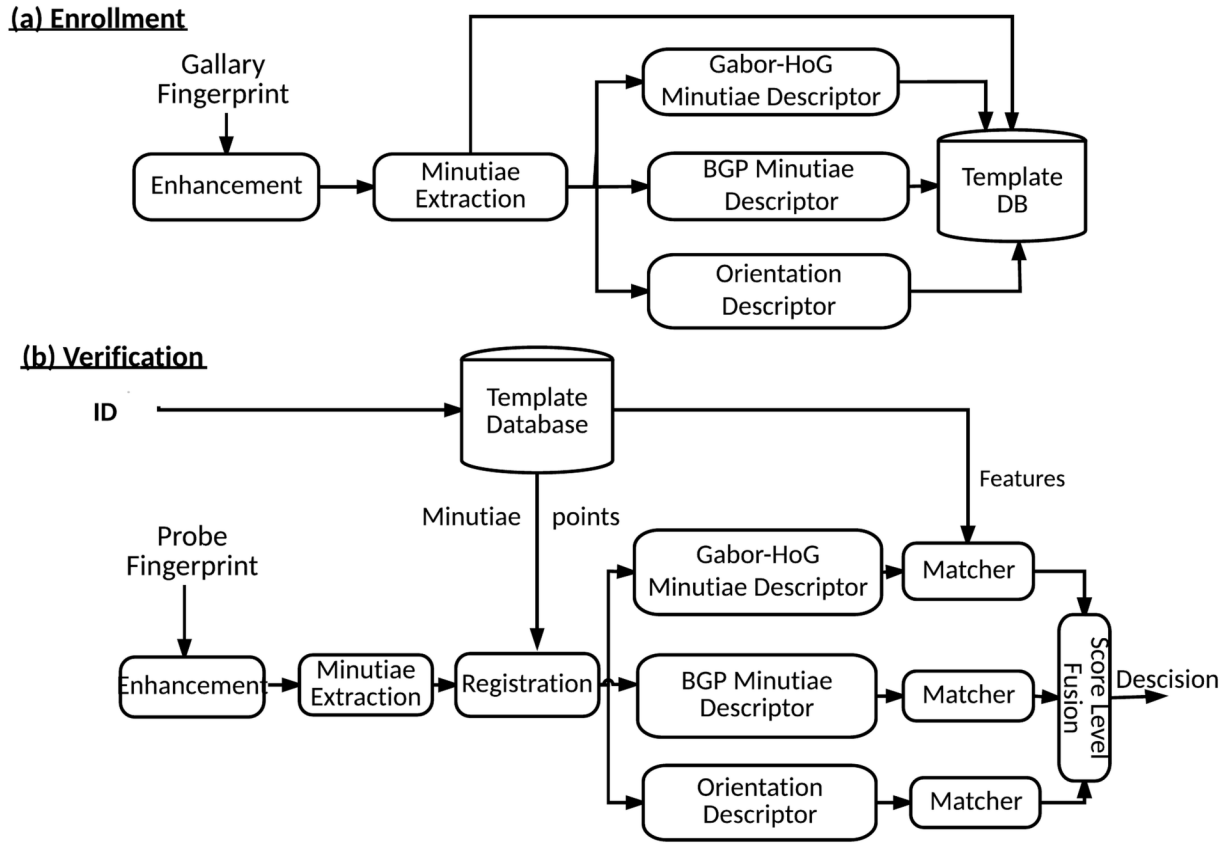

FIGURE 3. Overview of the proposed system: (a) enrollment phase, and (b) verification phase.

energy are estimated based on the Fourier spectrum. Next, contextual filtering is applied to enhance the fingerprints. The minutiae extraction method used in this work is based on the binarization and thinning approach proposed in [8].

B. MINUTIAE ALIGNMENT

Probe and template fingerprints are aligned using minutiae points before feature extraction from the probe fingerprint. For finding the correspondence between minutiae pairs, the method by Tico and Kuosmanen [18] is applied. In this method, the minutiae descriptor is constructed based on the choice of sampling points that are arranged in concentric circles around the minutiae; then, the orientation values for these sampling points are calculated. For each probe minutiae, $q_j \in Q = \{q_1, q_2, \ldots, q_n\}$ and for each template minutiae, $p_i \in P = \{p_1, p_2, \ldots, p_m\}$, the algorithm calculates the value of a parameter (possibility value), which expresses the probability of the two minutiae points, q_j and p_i being similar. Then, all minutiae pairs are sorted in descending order based on their possibility values. The probe minutiae are transformed such that the possibility values are maximum for all corresponding pairs in the template and the query fingerprints. Next, a greedy algorithm is applied to identify corresponding minutiae pairs that satisfy two conditions: 1) the Euclidean distance between the two minutiae does not exceed a specific threshold Δd, and 2) the difference between the two minutiae directions does not exceed the threshold $\Delta\theta$.

C. FEATURE EXTRACTION

For cross-sensor matching, the discriminative information includes ridge patterns and minutia points. To encode ridge patterns and ridge orientations, three types of descriptors were employed: Gabor-HoG, BGP and orientation descriptors. BGP and Gabor-HoG descriptors encode multiscale local ridge patterns and local ridge orientations around each minutia point whereas the orientation descriptor extracts ridge orientation from the foreground of the fingerprint. A detailed account of these descriptors is given in the following paragraphs.

1) ORIENTATION DESCRIPTOR

We argue that the ridge orientation is a robust fingerprint feature because it is invariant to rotation and translation and can be computed with acceptable accuracy, even from a noisy fingerprint. Ridge orientation offers a distinguishing description of a fingerprint that does not change even if fingerprints of the same subject are acquired from different sensors.

For computing the ridge orientation, the foreground of each enhanced fingerprint is divided into overlapping square blocks $S_i, i = 1, 2, , b$ with 50% overlap and a fixed

dimension $D = 25$. Each block is first normalized and then the ridge orientation is calculated using the method proposed in [8].

Normalization is carried out to minimize the variation in the gray level values along furrows and ridges. It helps in removing artifacts due to differences in finger pressure and sensor noise. It is carried out using the following formula:

$$N_i(x,y) = \begin{cases} M_0 - \sqrt{\dfrac{V_0 \times (S_i(x,y) - M_i)^2}{V_i}} \\ \qquad if\ I(x,y) > M_i \\ M_0 + \sqrt{\dfrac{V_0 \times (S_i(x,y) - M_i)^2}{V_i}} \\ \qquad otherwise. \end{cases} \quad (1)$$

where $S_i(x,y)$ denotes the intensity value of the pixel at position (x,y) of a block S_i, M_i and V_i denote the computed mean and variance of the block S_i, respectively, and $N_i(x,y)$ is the normalized gray level value at pixel (x,y). M_0 and V_0 are the desired mean and variance. For our experiments, we set M_0 to zero and V_0 to one.

The computation of the ridge orientation is affected by noise. To counter the effect of noise, the ridge orientation is computed using the least squares estimate of the local ridge orientation. First, each block is divided into non-overlapping windows of size $w \times w$ and a ridge orientation is assigned to each window using the gradients-based approach. The orientation of a window (i,j) is defined as follows:

$$\theta(i,j) = \tanh^{-1}\left(\frac{G_yy(i,j)}{G_xx(i,j)}\right) \quad (2)$$

where

$$G_yy(i,j) = \sum_{u=i-w/2}^{i+w/2}\sum_{v=j-w/2}^{j+w/2} 2(G_x(u,v)G_y(u,v)) \quad (3)$$

$$G_yy(i,j) = \sum_{u=i-w/2}^{i+w/2}\sum_{v=j-w/2}^{j+w/2} (G_x^2(u,v) - G_y^2(u,v)) \quad (4)$$

where G_x and G_y are the gradient magnitudes in the x and y directions, respectively. To compute G_x and G_y, Sobel operator is applied. The Sobel operator embeds Gaussian like behavior and is inherently equipped with denoising capabilities. For this reason, it has been extensively used for computing image gradient, in particular for fingerprints [28]–[30]. Owing to the existence of noise, the orientation of a window is smoothed using a Gaussian filter as follows:

$$\theta'(i,j) = \frac{1}{2}\tanh^{-1}\left(\frac{G(x,y)\sin(2\theta(i,j))}{G(x,y)\cos(2\theta(i,j))}\right) \quad (5)$$

where $G(x,y)$ is the Gaussian smoothing kernel.

For the orientation descriptor, a fingerprint is first divided into blocks, and the orientation field of each block is calculated. Then, the histogram of the orientation field of each block is computed. Finally, the orientation descriptor is formed by concatenating the histograms corresponding to all the blocks. Table 1 and Fig. 2 (c) present the effect of this descriptor on cross-sensor matching.

2) MINUTIAE-BASED DESCRIPTORS

The local descriptors that encode the local ridge structure around minutiae are characterized by invariance to global transformations such as rotation and translation. Employing BGP [26] and Gabor-HoG, we propose minutiae-based descriptors for encoding the local ridge structure. For this purpose, a square window of size $r \times r$ is defined around each minutia of the enhanced fingerprint and is normalized using the same method described in the orientation descriptor.

a: BINARY GRADIENT PATTERN MINUTIAE DESCRIPTOR

Based on image gradients, the BGP [26] represents the local orientation structure around a pixel in the form of a binary string that encodes directional variation of gray-level values in the local neighborhood of a pixel. BGP patterns can represent micro-edges along various directions such that it is suitable for representing the local structure around a minutia point.

The BGP considers a symmetric neighborhood around a pixel and converts the neighboring gray-level values into bits taking into account the differences among gray-level values along multiple (k) directions. The computation of the basic BGP code with eight neighbors and four directions is shown in Fig. 4.

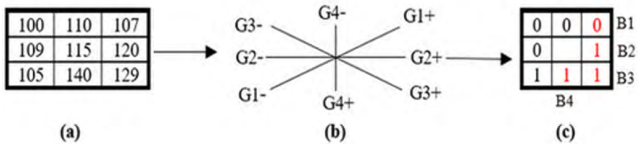

FIGURE 4. BGP operator: (a) a central pixel (of value 115) with eight neighbors, (b) the corresponding directions: G1, G2, G3, and G4, (c) principal (in red) and associated (in black) bits, the resulting BGP code is 0111 (in binary form) or 07 (in decimal form). .

For each pixel, two types of bits —the principal (B_i^+) and the associated (B_i^-) —are calculated using the following rules:

$$B_i^+ = \begin{cases} 1 & if\ G_i^+ - G_i^- \geq 0 \\ 0 & otherwise \end{cases}. \quad (6)$$

$$B_i^- = 1 - B_i^+ i = 1, 2, , k \quad (7)$$

where G_i^+ and G_i^- are the gray-level values of the pixels in the local symmetric neighborhood of the central pixel, as shown in Fig. 4 (b). The code of the central pixel is calculated from the resulting four principal bits using the following equation:

$$L = \sum_{i=1}^{k} 2^{(i-1)} B_i^+ \quad (8)$$

Patterns with zero or one-bit transition are defined as structural BGPs, such as 0000, 0001, and 0011, while other

patterns are referred to as non-structural patterns. Only structural BGP represents the local structure exploiting local edge orientations; it has the distinguishing property that assists not only in ruling out noise and outliers but also in reducing the dimension of the extracted feature vectors because there are only eight distinct structural BGP codes and only eight bins are needed when constructing a structural BGP histogram. The histogram of the structural BGP patterns of the window around a minutia form a local BGP descriptor. The effect of this descriptor on native and cross-sensor matching is shown in Fig. 2(a).

b: GABOR-HoG MINUTIAE DESCRIPTOR

The Gabor-HoG descriptor encodes the orientation field of an image by computing histograms of oriented gradients from the responses of Gabor filters; it offers a detailed description of the scales and the local orientation of the ridges. It was first adopted by Nanni and Lumini [13] in their work on fingerprint recognition. The difference between their work and this work is twofold: first, this work adopted the descriptor as a minutiae-based descriptor instead of applying it to encode the foreground of a fingerprint, and second, this work used eight orientations to obtain the filters instead of four. Using eight orientations, Gabor-HoG captures richer information about ridge orientation and ridge characteristics than when only four orientations are considered.

The 2D Gabor filter proved its effectiveness in fingerprint analysis to capture local ridge characteristics, including frequency information and the local orientation. In spatial domain, it is a Gaussian kernel modulated by a sinusoidal plane wave with the following general form:

$$G(x, y, f, \sigma, \theta) = exp(-\frac{x'^2 + y'^2}{2\sigma^2}) \cos(2\pi f x') \quad (9)$$

$$x' = x \sin\theta + y \cos\theta \quad (10)$$

$$y' = x \sin\theta - y \cos\theta \quad (11)$$

where f is the frequency of the sinusoidal wave, θ is the orientation, and σ is the standard deviation of the Gaussian envelope.

A filter bank is usually constructed using Gabor filters with different orientations and scales. We considered four scales and eight orientations for each scale ($\theta = 0°$, $22.5°, 45°, 67.5°, 90°, 112.5°, 135°, 157.5°$). While θ can be measured in radians or degrees, the use of degrees is convenient when dealing with ridge orientations, which is why these units have been extensively used in fingerprint recognition [3], [4], [31].

HoG was first proposed by Dalal and Triggs [32] as an image descriptor with the aim to represent an image using local histogram that counts the occurrences of gradient orientations in a local cell of the image. As the magnitude of the gradient is large near the ridges, HoG is useful in encoding ridge patterns.

For building the minutia-based Gabor-HoG descriptor, feature maps are first created by filtering the window around a minutia point with the Gabor filter bank and HoG is then computed from each feature map. Finally, HoG descriptors from all features maps are normalized to reduce the effect of variation in the gray level values along furrows and ridges and to rule out artifacts due to sensor noise, and then concatenated.

When fingerprints come from the same subject and are captured with different sensors, the fingerprints have the same ridge patterns, which differ in scales. In this scenario, an ideal choice must be a descriptor that is robust against scale. Gabor filter bank captures the ridge micro-patterns at different orientations at various scales, and HoG encodes orientation information from each feature map, making it robust against sensor dependent variations. In view of this discussion, we argue that the minutia based Gabor-HoG descriptor captures the local ridge pattern and is robust against sensor dependent variations. The effect of this descriptor on native and cross-sensor matching is shown in Fig. 2(b)

D. FINGERPRINT MATCHING

The similarity between two fingerprints is calculated (taking into account the corresponding descriptors) in the form of a matching score. An orientation descriptor is essentially a histogram, and the similarity between two descriptors can be measured using city-block distance, chi-square distance, and histogram intersection. In our experiments, we found that city-block distance and chi-square distance yield similar matching results, which are better than histogram intersection. Further, we employed city-block distance since it is computationally more efficient. The effects of different types of metrics are discussed in Section IV.B.

Each minutia-based descriptor is also a histogram; thus, city-block distance was employed for computing the matching score. The similarity between each minutia $m_i^T (i = 1, 2, \ldots, \text{n})$ of the template fingerprint T and its mate minutiae m_i^I (i = 1,2,...,n) of the probe fingerprint I is computed as city-block distance between the corresponding BGP descriptors. The matching score between T and I is the aggregate of the similarity scores of all pairs of minutiae points. The matching score using the Gabor-HoG descriptor is computed similarly.

E. SCORE-LEVEL FUSION

Table 1 and Fig. 2 indicate that the three types of descriptors result in different matching scores and the matching performance can be enhanced by fusion. There are three possibilities for information fusion: feature level fusion (combining the features), decision level fusion (combining the decisions), and score level fusion (combining the scores) [33]. The feature level fusion suffers from two problems: (1) the three types of features are not compatible, and (2) it leads to a high dimensional feature space. Fusion at the decision level has rigid and small information content. Score level fusion is easily adopted and can be utilized in an effective manner.

In view of this discussion, in this work, score level fusion was used for matching.

As the scores of matchers based on different descriptors have different ranges, it is necessary to normalize the scores such that they have the same range before they are fused. The scores are normalized using min-max normalization defined by the following formula:

$$nS_j = \frac{S_j - min_j}{max_j - min_j} \tag{12}$$

where the min and max are the minimum and maximum scores of the matcher j, S_j is the actual score, and nS_j is the normalized score.

The scores can be fused using different rules, such as minimum, maximum, product and sum. It is intuitive to take into account the matching performance of each matcher while matching the scores. In view of this, we compute the final score using the weighted sum rule defined by the following equation:

$$Score = w_1 S_o + w_2 S_b + w_3 S_h \tag{13}$$

where S_o, S_b, and S_h refer to the scores generated by the orientation, BGP, and Gabor-HoG descriptors, respectively, and w_1, w_2, andw_3 are their respective weights defined using the corresponding EER of each matcher as follows:

$$w_i = \frac{1}{EER_i} \tag{14}$$

The weight of each matcher is the reciprocal of its EER; this means that a matcher will get more weight if its EER value is less. The EER value of each matcher is computed using an independent dataset. The effect of the weighted sum rule is discussed in Section IV.B. The details of the enrollment module of the proposed method are summarized in Algorithm-1 whereas the details of the matching module are summerized in Algorithm-2.

Algorithm 1 Enrollment Module

Input:
T: Template fingerprint.
ID: subject ID.
Processing:
Extract minutiae from fingerprint T: $(m_1, m_2, ..., m_n)$
for *each minutiae* $m_j, j = 1, ..., n$ **do**
  Compute Bm_j, the BGP descriptor of minutia m_j.
  Compute Gm_j, the Gabor-HoG descriptor of minutia m_j.
Save m_j, Bm_j, Gm_j, $j = 1, ..., n$ to the subject with ID in the template database.
Compute OT, the orientation descriptor of the template fingerprint T.
Save OT to the subject with ID in the template database

Algorithm 2 Matching Module

Input:
I: Probe fingerprint.
ID: subject ID.
Output:
score: the matching score generated by the proposed method.
Processing:
Extract the minutiae from fingerprint I.
Retrieve the minutiae of r fingerprints of the subject with ID from the template database: $T_1, T_2, .., T_r$
for $i = 1 : r$ **do**
  Initialize S_i, s_i^o, s_i^b, s_i^g to zero.
  Align minutiae of I with $T_1 : (m_1^T, m_1^I), \ldots, (m_k^T, m_k^I)$, a set of matched minutiae pairs.
  for *each minutiae pair* $(m_j^T, m_j^I), j = 1, k,$ **do**
    Retrieve Bm_j^T, the BGP descriptor of minutia m_j^T
    Compute Bm_j^I, the BGP descriptor of minutia m_j^I
    Compute the similarity score $s_i^b = s_i^b + d(Bm_j^T, Bm_j^I)$
    Retrieve Gm_j^T, the Gabor-HoG descriptor of minutia m_j^T
    Compute Gm_j^I, the Gabor-HoG descriptor of minutia m_j^I
    Compute the similarity score $s_i^g = s_i^g + d(Gm_j^T, Gm_i^I)$
  Retrieve OT_i, the orientation descriptor of the r^{th} gallery fingerprint with ID
  Compute OI, the orientation descriptor of the probe fingerprint I
  Compute similarity score $s_i^o = s_i^o + d(OI, OT_i)$
  Normalize scores using Eq. (12)
  Fuse the scores $S_i = w_1 s_i^o + w_2\, s_i^b + w_3 s_i^h$
$Score = min\{S_1, S_2, S_r\}$
return *Score*

IV. MODEL SELECTION AND EVALUATION PROCEDURE

First, we present details of the databases, which were used to select the optimal parameters and validate the effectiveness of the system for cross-sensor fingerprint matching. Then, the results are presented, discussed, and compared with those of state-of-the-art methods.

It should be noted that the proposed method was implemented in the Matlab (R2016a) environment and experiments were performed on a PC (Intel Core i7-4702MQ processor, 2.2 GHz, 4 cores) with 14 GB RAM and the Microsoft Windows 10 × 64 operating system.

A. CROSS-SENSOR FINGERPRINT DATABASES

Experiments were conducted using two public domain cross-sensor databases: Multisensor Optical and Latent

Fingerprint (MOLF) [31] and FingerPass [34]. These databases contain a large number of fingerprints with variations in resolution, sensor type, and capture spectrum.

The MOLF database contains three datasets acquired with three optical sensors: (1) Lumidigm Venus IP65 Shell, (2) Secugen Hamster-IV, and (3) CrossMatch L-Scan Patrol. The resolution of the fingerprints captured with CrossMatch, Secugen, and Lumidigm is 500 dpi each, while the sizes are 1600×1500, 258×336, and 352×544 pixels, respectively. For each sensor, there are 1,000 fingerprint classes with four impressions for each. The fingerprints were captured in two sessions; two independent instances were captured in each session and the total number of fingerprints in each dataset is 4,000. Fig. 5 shows three impressions of the same fingerprint from the MOLF database; these impressions were captured with three sensors. The fingerprints show variations in resolution, quality, and noise pattern due to different technologies of the sensors.

FIGURE 5. Three impressions of the same fingerprint from the MOLF database. (a) Lumidigm Venus IP65 Shell optical sensor. (b) Secugen Hamster-IV optical sensor. (c) CrossMatch L-Scan Patrol optical sensor

The FingerPass database contains nine datasets acquired from nine different sensors. Table 2 presents details of each sensor. The fingerprints were captured using capacitive sensors and optical sensors, which are either sweep or press sensors. For each dataset, there are 720 fingerprint classes with 12 impressions for each fingerprint class. Therefore, there are 8,640 fingerprints in each dataset, and the total number of fingerprints in the FingerPass database is 77,760.

Fig. 6 shows fingerprints (from FingerPass captured with different sensors) of the same finger. It is a challenging database, as is obvious from the sample impressions.

B. EVALUATION PROTOCOL

For the evaluation of a matching system, the scores from two matching scenarios are of interest. These scenarios include 1) *regular matching* (also called native device or intra-device matching), which involves comparing two fingerprints acquired by the same sensor to generate a native equal error rate (EER), and 2) *cross matching* (cross-device or inter-device matching), which involves comparing two fingerprints captured with different sensors to generate an interoperable or cross-EER.

The performance is measured using the well-known EER measure, which is the operating point at which the false match rate (FMR) and false non-match rate (FNMR) are equal. The FMR is the rate at which the matching system falsely considers two different fingerprints to be from the same person while the FNMR is the rate at which the matching system considers two fingerprints from the same person to be different.

C. MODEL SELECTION

The proposed method involves four types of parameters: (i) block size for the orientation descriptor, (ii) window size for minutiae-based descriptors, (iii) a similarity measure for comparing the descriptors, and (iv) the weights for score-level fusion. The choice of a particular value of a parameter affects the performance of the matching system. In the following subsections, we discuss the effects of these parameters and suggest optimal choices.

1) EFFECT OF THE BLOCK SIZE IN ORIENTATION DESCRIPTOR AND THE METRIC FOR MATCHING

To assess the effect of the block size for the orientation descriptor, three block sizes (25, 50, and 70 pixels) were tested, and four similarity measures were used: chi-square, histogram intersection, city-block, and Euclidean distances. Fig. 7 reports the results of the orientation descriptor based only on randomly selected fingerprints from 200 subjects with 4 impressions for each obtained using two sensors from the MOLF database. The fingerprints captured with Lumidigm Venus IP65 Shell sensor are used for enrollment whereas the fingerprints captured with the Secugen Hamster IV sensor are used for verification.

Fig. 7 shows that the block sizes 25 and 50 produce similar results, which are better than the result produced by the larger block size of 70. This indicates that small block sizes are better for the performance of the system as the descriptor computed from smaller blocks captures the local structure in a better way than that calculated using a bigger block size. Based on these observations, the best choice for the block size is 25.

Further, Fig. 7 shows that city-block distance is the best of all four types of metrics. The effects of chi-square distance and Euclidean distances are close to those of city-block distance, but the performance of histogram intersection distance is the worst of all the four types of distances. This observation suggests that the best choice for the metric is city-block distance.

2) ANALYSIS OF THE EFFECTS OF DIFFERENT PARAMETERS IN MINUTIA-BASED DESCRIPTORS

Minutia-based descriptors involve different parameters: window size (w) around a minutia, radius (R) and the number of neighbors (P) in the BGP descriptor, the number of orientations (o) and scales (s) of the Gabor filters, and the number of cells (c) for the HoG in the Gabor-HoG descriptor. In the following paragraphs, we provide a detailed analysis of these parameters, and justify the selection of the most suitable set of values for these parameters using the same dataset that was used in the previous section. Firstly, experiments were conducted to estimate the window size for the minutiae-based descriptors. We tested three window sizes, i.e., $w = 25, 50,$

TABLE 2. Sensors and fingerprint details in the FingerPass cross-device-matching database.

Sub-dataset	Sensor	Technology Type	Interaction Type	Image Size (pixels)	Image Resolution
FXO	FX3000	Optical	Press	400×560	569 dpi
V3O	V300	Optical	Press	640×480	500 dpi
URO	URU4000B	Optical	Press	500×550	700 dpi
AEO	AES2501	Optical	Sweep	unfixed	500 dpi
ATC	ATRUA	Capacitive	Sweep	124×400	250 dpi
SWC	SW6888	Capacitive	Sweep	288×384	500 dpi
AEP	AES3400	Capacitive	Press	144×144	500 dpi
FPP	FPC1011C	Capacitive	Press	152×200	363 dpi
TCC	TCRU2C	Capacitive	Press	208×288	500 dpi

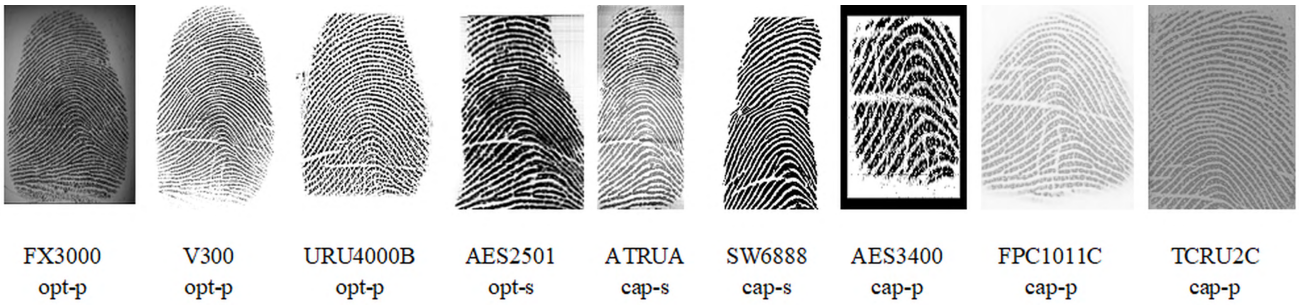

FIGURE 6. Nine impressions of the same finger from the FingerPass database; opt and cap means optical and capacitive sensors, respectively; p and s means press and sweep capture type, respectively.

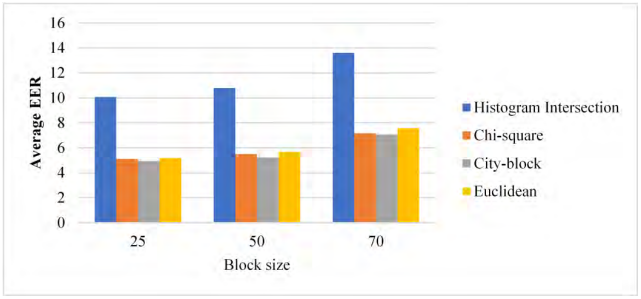

FIGURE 7. Average EER of the orientation descriptor on MOLF database with different block sizes and metrics.

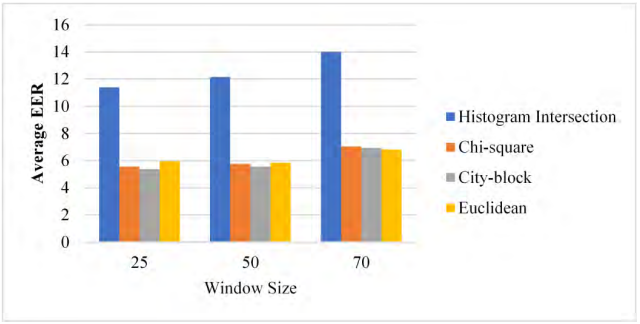

FIGURE 8. Average EER of the BGP descriptor on MOLF database showing the effect of window size.

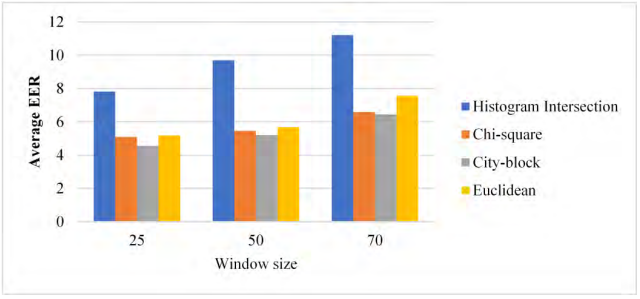

FIGURE 9. Average EER of the Gabor-HoG descriptor on MOLF database showing the effect of window size.

and 70 pixels using four metrics; Figs. 8 and 9 display the results for BGP with $R = 1$ and $P = 8$ and Gabor-HoG with $o = 8$, $s = 4$, and $c = 3$, respectively.

Figs. 8 and 9 show that the window size for minutia behavior is similar to the block size for the orientation descriptor. As the window size increases, the system performance deteriorates; the most suitable window size for the minutiae-based descriptors was determined to be 25 or 50. This indicates that a smaller size helps to capture the local structure more precisely, and our recommended best choice of w is 25. Figs. 8 and 9 also support the earlier conclusion from the orientation descriptor that city-block distance is the best choice. Again, the histogram intersection distance method performed the poorest among the four metrics studied.

BGP involves two parameters: radius (R) and the number of neighbors (P). Typically, R takes an integer value and P is usually set to $8R$. To assess the effects of (R, P), we tested three frequently used configurations: $(8, 1)$, $(16, 2)$ and $(24, 3)$, using block size $w = 25$ and the city-block distance as a similarity measure. Fig. 10 illustrates the average cross-EER results of the BGP descriptor on the MOLF database alone; the configuration (8,1) produces better results than the other configurations. This demonstrates that the best choice for the BGP descriptor is (8, 1).

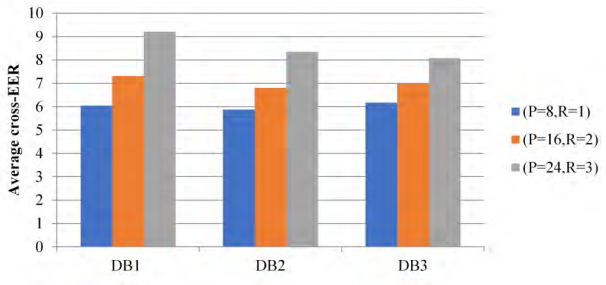

FIGURE 10. Average EER of the BGP descriptor on MOLF database showing the effect of radius (*R*) and the number of neighbors (*P*).

The minutia-based Gabor-HoG descriptor contains three parameters: the number of orientations (o), the number of scales (s), and the number of cells (c) in the HoG. For the best performance, suitable values for o, s and c must be selected. We first tested, three frequently used orientation configurations: $o = 2$ ($\theta = 0°, 90°$), $o = 4$ ($\theta = 0°$, 45°, 90°, 135°), and $o = 8$ ($\theta = 0°, 22.5°, 45°, 67.5°, 90°$, 112.5°, 135°, 157.5°) with three choices for scales $s = 2, 4$, and 8 and three frequently used options for the number of cells: c $= 3 \times 3$, 5×5, and 7×7. Figs. 11–13 shows the results using the Gabor-HoG descriptor alone on DB1, DB2, and DB3 as a gallery and the other database as probe on the MOLF database respectively. The results highlight that the Gabor filter bank, with 8 orientations, 4 scales, and with 3×3 HoG cells, captures the discriminating information about local orientations and the scales of the ridge patterns around a minutiae in a better way than other parameter values.

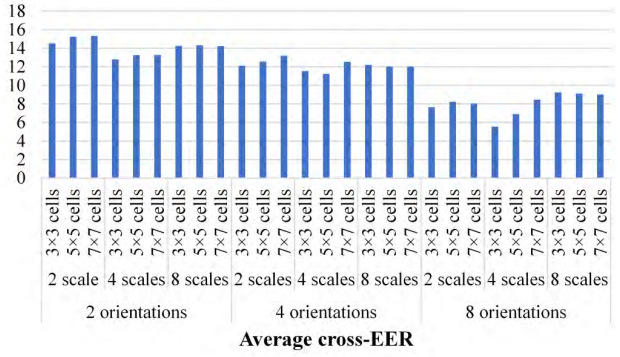

FIGURE 11. Average cross-EER of the Gabor-HoG descriptor on DB1 as a gallery from the MOLF database showing the effect of the number of orientations,scales, and cells.

In summary, the above results and discussion reveal that the selection of a specific parameter value has impact on the performance of the proposed system. In view of the analyses of the various parameters given above, the best set of parameters values is: $R = 1, P = 8$ for the BGP descriptor, and $o = 8$, $s = 4$, and $c = 3 \times 3$ for the Gabor-HoG descriptor, window size $w = 25$ and city-block distance for computing the similarity. The results provided and discussed in the following sections are based on this set of parameters.

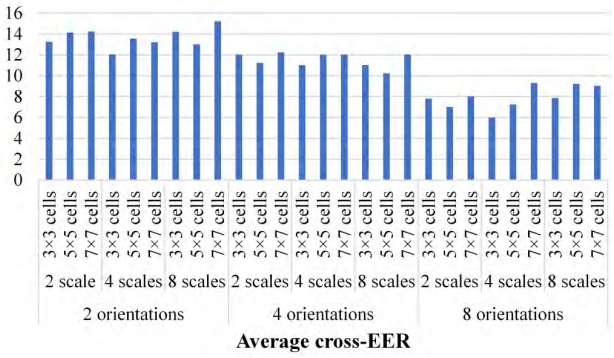

FIGURE 12. Average cross-EER of the Gabor-HoG descriptor on DB2 as a gallery from the MOLF database showing the effect of the number of orientations,scales, and cells.

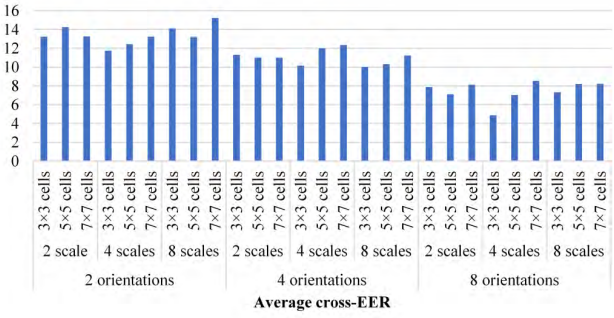

FIGURE 13. Average cross-EER of the Gabor-HoG descriptor on DB3 as a gallery from the MOLF database showing the effect of number of orientations,scales, cells.

3) EFFECT OF THE FUSION RULE

First, we select the fusion weights and then validate the effectiveness of the proposed weighted sum fusion rule. To select the weights, we treat the MOLF database as an independent set. Each matcher corresponding to the descriptors described in Section III was tested on the MOLF database, and the results are reported in Table 1. In these experiments, the block size for the orientation descriptor and the window size for the minutiae-based descriptors were 25; the city-block distance was used to measure the similarity.

As expressed in Section III, the weighted sum rule uses weights for fusing the scores from individual matchers and the reciprocal of the EER measure of each matcher is used as its weight. The weights are calculated using the average EER of each matcher. For example, w_1 is the weight that corresponds to the BGP descriptor and is computed by taking the average of the nine EER values ($[4.81 + 6.66 + 6.68 + 6.656 + 3.278 + 7.693 + 6.677 + 7.697 + 4.16]/9 = 6.034$) corresponding to the BGP descriptor (see Table 1), i.e., $w_1 = \frac{1}{EER} = 0.164$ similarly, w_2 (corresponding to the Gabor-HoG descriptor) is 0.183, and w_3 (corresponding to the orientation descriptor) is 0.177. These weights are used in subsequent experiments.

To validate the effectiveness of the weighted fusion rule, we examined commonly used fusion rules such as selecting

TABLE 3. EER obtained by the (a) minimum rule, (b) maximum rule, (c) product rule, (d) sum rule, and (e) the proposed weighted fusion rule.

Template → Probe	(a) Minimum rule			(b)Maximum rule			(c)Product rule			(d)Sum rule			(e)Weighted sum rule		
	DB1	DB2	DB3	DB1	DB2	DB3	DB1	DB2	DB3	DB1	DB2	DB3	DB1	DB2	DB3
DB1	**1.22**	2.89	2.10	**1.32**	3.45	3.32	**0.87**	2.19	1.51	**0.73**	2.06	1.34	**0.60**	1.99	1.19
DB2	2.89	**1.20**	2.18	3.45	**1.25**	3.51	2.19	**0.82**	1.61	2.06	**0.72**	1.28	1.99	**0.64**	1.24
DB3	2.10	2.18	**1.18**	2.32	3.51	**1.53**	1.51	1.61	**0.99**	1.34	1.28	**0.66**	1.19	1.24	**0.54**

the minimum or maximum score, the sum of scores, and the product of scores. The results are presented in Table 3; from the reported results, it can be observed that the minimum and maximum rules produce the worst results. The product rule performs better than the minimum and maximum rules, and the sum rule has the best results among the four rules. The proposed weighted fusion rule that gives the best results is based on the fact that this rule weighs each score according to the performance of the associated descriptor.

4) EFFECT OF PRECISION

We implemented the proposed method in MATLAB, which represents floating-point numbers in either double-precision or single-precision format. Double precision may improve the performance of the system, but at the expense of memory. All of the above-mentioned experiments were performed using single-precision floating-point numbers. To test the effect of the double-precision format on performance, we ran experiments in double precision. Table 4 reveals that there is no noticeable improvement in results when using double-precision floating-point numbers over single precision numbers.

TABLE 4. EERs obtained by the proposed method using (a) double-precision, and (b) single-precision floating-point numbers.

Template → Probe	(a) Double Precision			(b) Single Precision		
	DB1	DB2	DB3	DB1	DB2	DB3
DB1	**0.60**	1.99	1.19	**0.60**	1.99	1.19
DB2	1.99	**0.63**	1.24	1.99	**0.64**	1.24
DB3	1.19	1.24	**0.52**	1.19	1.24	**0.54**

D. RESULTS AND DISCUSSION

In this section, the results of the proposed method obtained on the two databases using the best choices of parameters (as discussed in previous subsections) are presented. Table 5 presents the EERs obtained on the MOLF database. It can be observed that in general, the EER is low when the probe and template fingerprints are acquired by the same sensor; on the other hand, the EER is relatively high, but acceptable, when the probe and template fingerprints are acquired by different sensors.

Table 6 presents the verification results in terms of the EER obtained by the proposed method on the FingerPass database. The native EER (i.e., when the same sensor is used for gallery and probe) is small (less than 1) for all sensors except AEP and FPP. Furthermore, it can be observed that

TABLE 5. Verification results in terms of EER by the proposed method on MOLF database.

Template → Probe	DB1	DB2	DB3
DB1	**0.60**	1.99	1.19
DB2	1.99	**0.64**	1.24
DB3	1.19	1.24	**0.54**

TABLE 6. Verification results in terms of EER by the proposed method on FingerPass database.

Template → Probe	FXO	V3O	URO	AEO	ATC	SWC	AEP	FPP	TCC
FXO	**0.18**	0.55	0.82	0.45	3.01	1.95	6.36	3.22	0.18
V3O	0.55	**0.00**	0.00	0.00	1.21	0.78	6.48	2.31	0.09
URO	0.82	0.00	**0.00**	0.55	2.93	2.74	8.80	1.52	0.82
AEO	0.45	0.00	0.55	**0.05**	2.09	0.67	4.98	2.82	0.18
ATC	3.01	1.21	2.93	2.09	**0.00**	1.83	6.56	2.33	0.59
SWC	1.95	0.78	2.74	0.67	1.83	**0.05**	4.25	2.98	0.41
AEP	6.36	6.48	8.80	4.98	6.56	4.25	**5.66**	9.02	3.00
FPP	3.22	2.31	1.52	2.82	2.33	2.98	9.02	**1.46**	0.50
TCC	0.18	0.09	0.82	0.18	0.59	0.41	3.00	0.50	**0.00**

when AEP and FPP are used for gallery or probe, the cross EER on average is highest among all sensors; FPP stands next to AEP. Both AEP (AES3400) and FPP (FPC1011C) are capacitive sensors with press interaction type and the image sizes are 144×144 and 152×200 pixels, respectively. Note that for FPP, the image size is comparatively higher, and the cross EER is also relatively higher. The poor performance of AEP and FPP when used for gallery or probe is likely because the image size is small. In the case of AEP, there is a significant difference between the ridge width and the valley width (see Fig. 6) and distortion in the image; for this reason, the descriptors do not capture the ridge structures properly.

If FXO (FX3000, an optical sensor with press interaction type) is used for gallery, the cross EER is small (less than 1), except when ATC, SWC, AEP, and FPP are used for probe. Similar results are obtained when FXO is used for probe and ATC, SWC, AEP, and FPP are used for gallery. One likely reason is that FXO is an optical sensor, whereas ATC, SWC, AEP, and FPP are capacitive sensors. Another reason is that for ATC, SWC, AEP, and FPP, the image size is lower and the distortion is more intensive than for FXO. The highest cross EER is with AEP as discussed above. In the case of URO (URU4000B, optical, press), the performance is almost similar when ATC, SWC, AEP, and FPP are used for gallery or probe.

When V3O (V300, optical, press) is used for gallery or probe, the cross EER is small, except with ATC, AEP,

and FPP. The likely reasons are the difference in the technology type and the distortion. In the case of AEO (AES2501, optical, sweep), the cross EER is almost similar when ATC, AEP, and FPP are used for gallery or probe.

In most of the cases, the cross EER is greater than 1 irrespective of whether ATC, SWC, AEP, and FPP are used for galley or probe; in case cross EER they can be ranked (lowest cross EER first) as SWC, ATC, FPP, and AEP. If we examine the image sizes of the corresponding fingerprints, these sensors can be ranked as SWC ($288 \times 384 = 111$Kpixels), ATC ($124 \times 400 = 48$Kpixels), FPP ($152 \times 200 = 30$Kpixels), and AEP ($144 \times 144 = 21$Kpixels). It seems that there is a correlation between the image size and the cross EER. However, this is not the case when we check the results for TCC. Interestingly, for TCC (TCRU2C, capacitive, press), the cross EER is less than 1, irrespective of whether it is used for gallery or probe, except AEP, despite resulting in smaller size fingerprints ($208 \times 288 = 60$Kpixels). A close look at the fingerprints generated with TCC (see Fig. 6) reveals that it creates less distortion than ATC, SWC, AEP, and FPP. It indicates that the important factor is the amount of distortion. It is interesting to note that the proposed system performs well even if either the gallery sensor or the probe sensor has distortion. The performance degrades when both gallery and probe sensors result in severe distortion.

Fig. 14 shows the average cross EERs obtained by the proposed system on each dataset of the FingerPass database. It can be observed that the cross EER for all datasets is under 5% and the cross EER is high for the AEP database owing to the reasons discussed above. The overall results indicate that the proposed descriptors are robust in capturing the ridge patterns and the system gives good performance when the amount of distortion is small. However, the performance degrades when the fingerprints are badly distorted as is in the case of AEP.

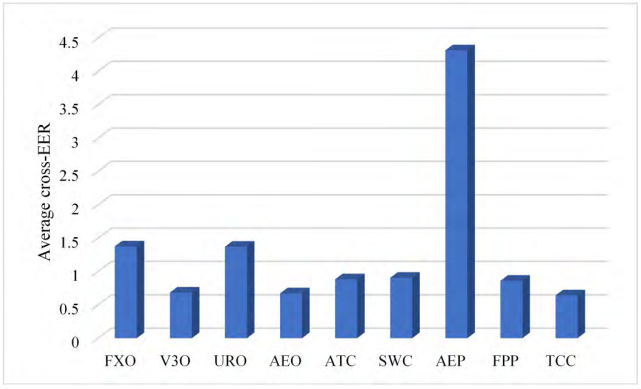

FIGURE 14. Average cross-EERs obtained by the proposed method on the datasets of FingerPass database.

The FingerPass database allows testing of the performances of a system based on the sensor technology type. The datasets can be categorized into two groups based on the sensor technology type: optical and capacitive. The optical group contains FXO to AEO (four sensors), while the capacitive group includes the remaining five sensors. The bar graph in Fig. 15 reports the average EER based on the sensor technology type produced by the proposed system. The native EER (when the sensors of the same technology type are used, i.e., optical vs. optical and capacitive vs. capacitive) of each group is computed as the mean of the cross EER of the same sensor type from Table 6, whereas the interoperable EERs (when sensors of different technology types are used, i.e., optical vs. capacitive and vice versa) are calculated as the mean of the cross EERs obtained when sensors of different types are employed. As can be seen from the graph, a lower native EER is obtained for the optical-vs-optical group compared to the capacitive-vs-capacitive group. The interoperable EER is higher in both cases. It is known that capacitive sensors are prone to noisy artifacts, which introduce deformations to the fingerprint images. It indicates that the proposed system gives the best performance for the optical-vs-optical group.

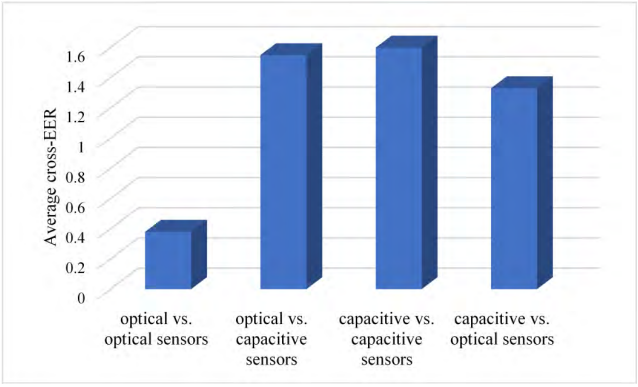

FIGURE 15. Average native and interoperable EERs obtained by the systems.

E. COMPARISON WITH THE STATE-OF-THE-ART METHODS

To evaluate the performance of the proposed system we compared it with four state-of-the-art methods: MCC [35], VeriFinger [36], MCC+Scale [22], and thin-plate spline (TPS) [16]. MCC is a state-of-the-art minutiae-based matching algorithm and MCC+Scale is an enhancement of MCC. VeriFinger is a well-known commercial-matching algorithm developed by Neurotechnology. MCC and VeriFinger are considered as the baseline for comparisons by various research works for regular-matching and cross-matching [22], [34].

1) RESULTS ON MOLF DATABASE

For comparison on MOLF, we used VeriFinger Extended SDK 9.0 and MCC SDK Version 2.0. It should be noted that VeriFinger uses its minutiae extraction algorithm. Minutiae extraction is an integral part of MCC; it computes cylinders using minutiae distances and angles. However, the MCC SDK developed by the authors does not contain a minutiae-extraction algorithm; consequently, in order to make a fair comparison, we used the same minutia-extraction

TABLE 7. Verification results in terms of EER by the three systems on MOLF database.

Template → Probe	(a) MCC Method			(b)VeriFinger Method			(c)Proposed Method		
	DB1	DB2	DB3	DB1	DB2	DB3	DB1	DB2	DB3
DB1	**11.14**	18.48	20.81	**3.16**	6.46	6.42	**0.60**	1.99	1.19
DB2	18.48	**16.82**	22.74	6.47	**3.20**	3.94	1.99	**0.64**	1.24
DB3	20.81	22.74	**13.83**	6.42	3.94	**3.51**	1.19	1.24	**0.54**

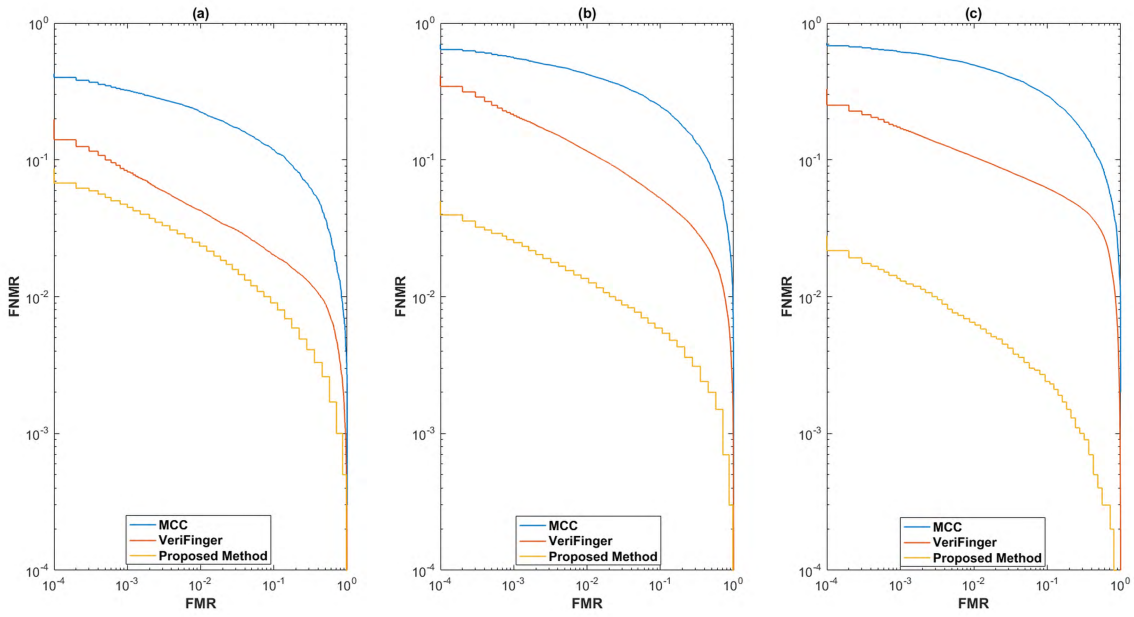

FIGURE 16. DET curves corresponding to the three systems on the MOLF database: (a) DB1 vs. DB1, (b) DB1 vs. DB2, and (c) DB1 vs. DB3.

algorithm for the MCC computation as was used in the proposed method (PM).

Table 7 presents the EERs obtained by the matching systems on the MOLF database. It can be observed that in general, the native EER is lower than the cross EER for the three methods. The overall performance of MCC is very poor for regular matching and cross-sensor matching, and for cross-sensor matching, the performance is worst. Although the results by VeriFinger are better than the results by MCC, it shows poor performance for case of cross-sensor matching. The proposed approach yields the lowest native and cross EERs compared to VeriFinger and MCC on all three datasets of the MOLF database. Figs. 16-18 show the graphs of DET curves for the three systems. The DET curves are consistent with the results presented in Table 7. The proposed method always achieves the best results in all cases in terms of DET curves, and with significant difference.

Table 8 summarizes the average matching times of the three systems on the MOLF database. The results show that, in terms of matching, VeriFinger is faster than both PM and MCC. Though PM is slower than VeriFinger and MCC, its matching time is acceptable for real matching scenarios. The overhead of the proposed method is associated with the alignment step; apart from this step, the average matching time is 1.0469×10^{-4} seconds. We intend to explore ways of removing this alignment step in future work.

TABLE 8. Average matching time (in seconds) on MOLF database.

Method	VeriFinger	MCC	PM
Matching time	0.0238	0.794	1.943

2) RESULTS ON FINGERPASS DATABASE

For a fair comparison with the state-of-the-art on FingerPass database, we selected four subsets from the database, which were considered in [22]: URO (optical, press), AEO (optical, sweep), SWC(capacitive, sweep), and TCC (capacitive, press). Table 9 presents a comparison of the four databases in terms of EER, FMR100, FMR1000, and ZeroFMR; FMR100 and FMR1000 are the minimum values of FNMR when FMR 1% and FMR 0.1%, respectively, and ZeroFMR is the lowest FNMR when no false match occurs.

The effects of using different sensors for gallery and probe on the performance of the compared methods are noticeable. For native matching, the proposed method (PM) outperforms MCC and VeriFinger for URO and TCC, both having the press interaction type; the performance of VeriFinger is slightly better, however, the performance of MCC is comparable with PM for AEO and SWC, both having the sweep interaction type. A close look at Fig. 6 reveals that the fingerprints have a small amount of distortion for URO and TCC, whereas the amount of distortion is high for AEO and SWC. This indicates that the performance of PM is

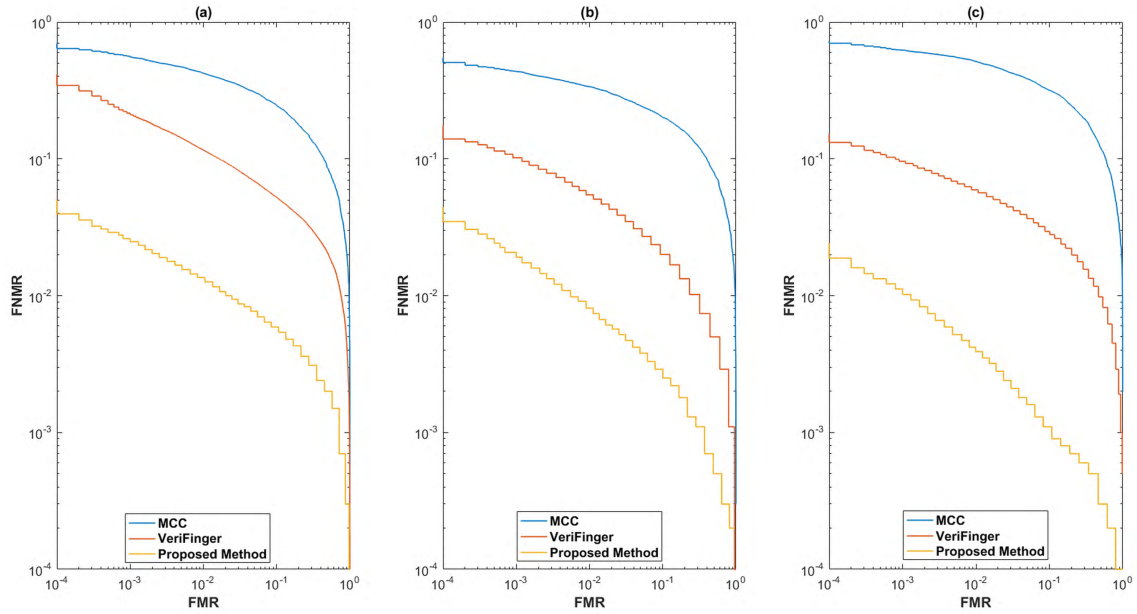

FIGURE 17. DET curves corresponding to the three systems on the MOLF database: (a) DB2 vs. DB1, (b) DB2 vs. DB2, and (c) DB2 vs. DB3.

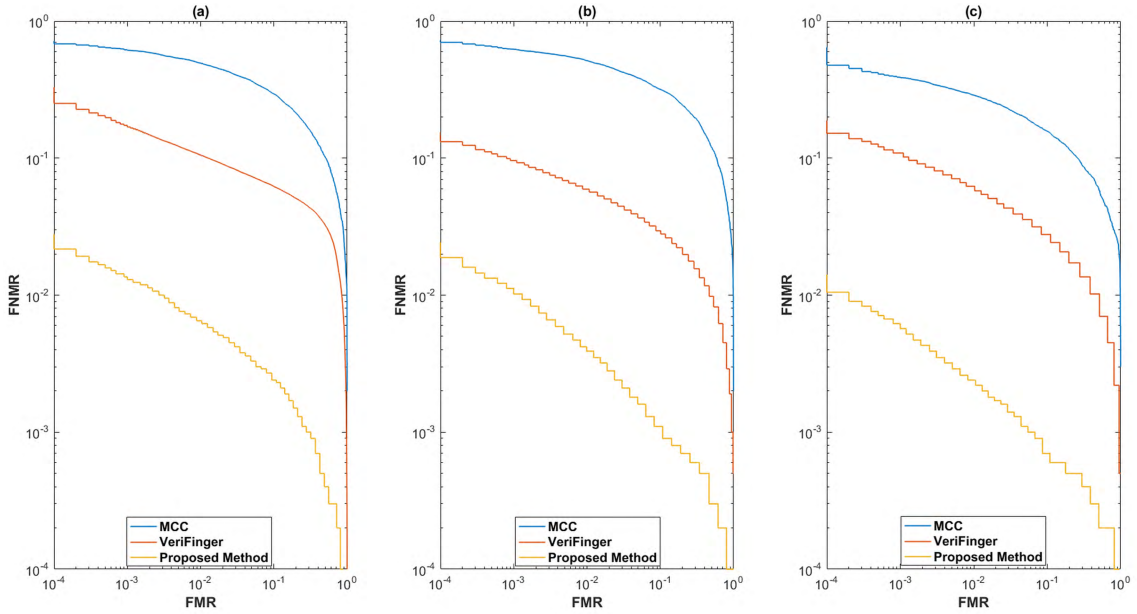

FIGURE 18. DET curves corresponding to the three systems on the MOLF database: (a) DB3 vs. DB1, (b) DB3 vs. DB2, and (c) DB3 vs. DB3.

better than that of VeriFinger and MCC when the amount of distortion is small.

For cross-sensor matching, the performance of PM is better than all the four methods with a large margin, except for URO vs. TCC, where the performance of PM is significantly better than MCC but shows slightly poor efficiency than VeriFinger, MCC with scale, and TPS. Please note that the VeriFinger system uses minutiae along with a number of proprietary algorithm solutions, such as ridge count, while MCC uses a neighborhood of fixed size around a minutia to encode directional and spatial relationships, which are represented by a cylinder whose height and base are related to the directional and spatial information, respectively. MCC with scale is based on scaling fingerprints before applying the MCC method. The TPS method uses the thin-spline model for registering a pair of fingerprints. The experimental results indicate that the features adopted by these systems are not robust against cross-sensor fingerprint matching. When using

TABLE 9. Performance comparison of the proposed method with state-of-the-art methods on four datasets from FingerPass database, PM means proposed method.

Template	Probe	Method	EER(%)	FMR100(%)	FMR1000(%)	ZeroFMR(%)
URO	URO	MCC	0.023	0.007	0.019	0.858
		VeriFinger	0.018	-	-	0.421
		PM	**0**	**0**	**0**	**0**
TCC	TCC	MCC	0.056	0.015	0.047	1.229
		VeriFinger	0.045	-	-	0.137
		PM	**0**	**0**	**0**	**0**
AEO	AEO	MCC	0.053	0.017	0.044	1.191
		VeriFinger	**0.014**	-	-	0.042
		PM	**0.05**	**0**	**0.0909**	**2.45**
SWC	SWC	MCC	0.073	0.028	0.061	2.517
		VeriFinger	**0.028**	-	-	0.109
		PM	**0.05**	**0.1818**	**3.909**	**1.909**
URO	TCC	MCC	27.41	89.23	98.65	99.99
		MCC+Scale	0.283	0.128	0.457	2.936
		TPS	0.298	0.128	0.512	1.552
		VeriFinger	0.272	-	-	0.714
		PM	**0.82**	**0.517**	**1.63**	**3.014**
AEO	SWC	MCC	2.482	3.67	7.681	24.93
		MCC+Scale	2.242	3.309	5.827	11.42
		TPS	2.017	3.546	7.328	15.63
		VeriFinger	1.083	-	-	3.511
		PM	**0.67**	**0.003**	**0.017**	**0.545**
URO	AEO	MCC	27.97	90.019	8.44	99.97
		MCC+Scale	2.432	4.186	10.89	31.06
		TPS	2.288	3.747	10.42	27.69
		VeriFinger	2.675	-	-	6.631
		PM	**0.55**	**1.090**	**1.182**	**4.818**
TCC	AEO	MCC	3.305	5.358	10.5	33.1
		MCC+Scale	2.632	3.581	6.137	15.26
		TPS	1.948	2.444	5.758	8.683
		VeriFinger	2.907	-	-	8.159
		PM	**0.18**	**0**	**1.091**	**1.727**
URO	SWC	MCC	26.41	87.26	97.39	99.86
		MCC+Scale	3.326	7.854	15.47	28.37
		TPS	3.158	7.329	13.56	25.73
		VeriFinger	3.487	-	-	10.92
		PM	**2.74**	**0**	**1.636**	**6.01**
TCC	SWC	MCC	5.21	9.701	18.59	47.19
		MCC+Scale	4.437	8.43	13.86	25.41
		TPS	4.382	8.592	13.94	25.83
		VeriFinger	4.263	-	-	19.58
		PM	**0.41**	**1.1667**	**4.833**	**17.83**

different sensors, the fingerprints of the same subject have the same ridge patterns, but they differ in details such as local micro-structures, rotations, and scales. These issues should be considered in designing a cross-sensor fingerprint recognition method; however, the designs used in other methods do not take this information into account.

A closer examination of the FMR100, FMR1000, and zeroFMR values in Table 9 reveals that the PM provides smaller values of TCC and URO for native matching, and outperforms the other methods for cross-matching, except when URO is compared to TCC. Accordingly, the PM improves cross-matching performance by reducing FNMR at different levels of FMR to a great extent; i.e. the rate of genuine-attempt denial is minimum at a certain security level. In particular, in most of the cases, it results in significantly lower FNMR at zero false match than the other methods.

The PM reduces the effects of fingerprint sensor interoperability significantly; overall, it outperforms VeriFinger, MCC, MCC with scale, and TPS. The results validate the capability of the proposed method to reduce the effects of the fingerprint sensor interoperability problem. The reason is that the proposed system is based on descriptors that are robust against variations in fingerprints due to using different sensors such as details of local micro-structures and rotations.

Although the proposed method outperforms other methods, overall, the best verification performance is given by the three methods in the case of the optical vs. optical group. Based on this observation, we recommend the use of the optical vs. optical group for best verification results.

V. CONCLUSION

In this paper, we proposed an automatic fingerprint verification method to deal with the fingerprint sensor interoperability problem. The discriminative characteristics of the fingerprints captured with different types of sensors were the ridge structures and minutiae. The proposed method relies on three types of descriptors: orientation, BGP, and Gabor-HoG descriptors. The orientation descriptor extracts ridge

orientations from the foreground of the fingerprint while the BGP and Gabor-HoG descriptors encode multi-scale local ridge patterns and local ridge orientations around a minutia. The scores obtained from matchers based on these descriptors were fused using a simple weighted sum, which assigns weight to each matcher in accordance with its matching performance. The proposed system was robust against sensor dependent structural variability, rotation, and scale variations. Missing and spurious minutiae are tolerated because the orientation descriptor does not depend on minutiae. Local distortion errors and small feature extraction errors are compensated by the BGP descriptor because it effectively captures the local structure. Intensive experiments were conducted to evaluate the performance of the system using two public domain databases: FingerPass and MOLF, which focus on the fingerprint sensor interoperability problem, and the system was compared with four state-of-the-art methods: MCC, MCC with scale, VeriFinger, and TPS. The results indicate that the proposed method significantly outperformed these methods. MCC showed the worst performance; VeriFinger gave a good performance for the optical vs. optical group; the proposed method gave a good performance for almost all cases. For best performance in the case of sensor interoperability, the recommendation of this study is to use optical vs. optical group.

Although the proposed system significantly overcomes the fingerprint sensor interoperability problem, its performance is not as good for different sensor technology types for the optical vs. optical group. It showed poor performance only when gallery or probe fingerprints were badly distorted; the solution to this problem is the use of a distortion correction method (for instance, the method by Tico and Kuosmanen [18] before applying the proposed method. One future direction to enhance the performance is to employ deep learning for feature extraction and to explore different feature levels and score level fusion techniques. In addition, we plan to extend our approach to address the compatibility between different types of interactions with sensors such as press and sweep.

ACKNOWLEDGMENT

This work was supported by the Research Center of College of Computer and Information Sciences, King Saud University. The authors are grateful for this support.

REFERENCES

[1] A. Ross and A. Jain, "Biometric sensor interoperability: A case study in fingerprints," in *Proc. Int. Workshop Biometric Authentication*, 2004, pp. 134–145.

[2] L. Lugini, E. Marasco, B. Cukic, and I. Gashi, "Interoperability in fingerprint recognition: A large-scale empirical study," in *Proc. IEEE/IFIP DSN-W*, Jun. 2013, pp. 1–6.

[3] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 846–859, May 2000.

[4] A. Ross, A. Jain, and J. Reisman, "A hybrid fingerprint matcher," *Pattern Recognit.*, vol. 36, no. 7, pp. 1661–1673, Jul. 2003.

[5] L. Nanni and A. Lumini, "A hybrid wavelet-based fingerprint matcher," *Pattern Recognit.*, vol. 40, no. 11, pp. 3146–3151, 2007.

[6] H. Guesmi, H. Trichili, A. M. Alimi, and B. Solaiman, "Curvelet transform-based features extraction for fingerprint identification," in *Proc. BIOSIG*, Sep. 2012, pp. 1–5.

[7] C.-L. Su, "r-Theta and orientation invariant transform and signal combining for fingerprint recognition," *Expert Syst. Appl.*, vol. 37, no. 7, pp. 5307–5313, 2010.

[8] L. Hong, Y. 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.

[9] L. Shen, A. Kot, and W. Koo, "Quality measures of fingerprint images," in *Proc. Int. Conf. Audio- Video-Based Biometric Person Authentication*, 2001, pp. 266–271.

[10] C.-J. Lee and S.-D. Wang, "A Gabor filter-based approach to fingerprint recognition," in *Proc. IEEE Workshop Signal Process. Syst.*, Oct. 1999, pp. 371–378.

[11] C. Gottschlich, "Curved-region-based ridge frequency estimation and curved Gabor filters for fingerprint image enhancement," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2220–2227, Apr. 2012.

[12] Y. Mei, B. Zhao, Y. Zhou, and S. Chen, "Orthogonal curved-line Gabor filter for fast fingerprint enhancement," *Electron. Lett.*, vol. 50, no. 3, pp. 175–177, Jan. 2014.

[13] L. Nanni and A. Lumini, "Descriptors for image-based fingerprint matchers," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 12414–12422, 2009.

[14] A. M. Olmos, G. Botella, E. Castillo, D. P. Morales, J. Banqueri, and A. García, "A reconstruction method for electrical capacitance tomography based on image fusion techniques," *Digit. Signal Process.*, vol. 22, no. 6, pp. 885–893, 2012.

[15] A. Ross, S. Dass, and J. Anil, "A deformable model for fingerprint matching," *Pattern Recognit.*, vol. 38, no. 1, pp. 95–103, 2005.

[16] A. Ross and R. Nadgir, "A thin-plate spline calibration model for fingerprint sensor interoperability," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 8, pp. 1097–1110, Aug. 2008.

[17] C.-X. Ren, Y.-L. Yin, J. Ma, and H. Li, "Fingerprint scaling," in *Proc. Int. Conf. Intell. Comput.*, 2008, pp. 474–481.

[18] Y. Zang, X. Yang, X. Jia, N. Zhang, J. Tian, and J. Zhao, "Evaluation of minutia cylinder-code on fingerprint cross-matching and its improvement with scale," in *Proc. ICB*, Jun. 2013, pp. 1–6.

[19] Y. Zang, X. Yang, X. Jia, N. Zhang, J. Tian, and X. Zhu, "A coarse-fine fingerprint scaling method," in *Proc. ICB*, Jun. 2013, pp. 1–6.

[20] F. Alonso-Fernandez, R. N. J. Veldhuis, A. M. Bazen, J. Fierrez-Aguilar, and J. Ortega-Garcia, "Sensor interoperability and fusion in fingerprint verification: A case study using minutiae-and ridge-based matchers," in *Proc. ICARCV*, Dec. 2006, pp. 1–6.

[21] E. Marasco, L. Lugini, B. Cukic, and T. Bourlai, "Minimizing the impact of low interoperability between optical fingerprints sensors," in *Proc. BTAS*, Sep./Oct. 2013, pp. 1–8.

[22] M. Tico and P. Kuosmanen, "Fingerprint matching using an orientation-based minutia descriptor," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 1009–1014, Aug. 2003.

[23] X. Xie, F. Su, and A. Cai, "Ridge-based fingerprint recognition," in *Proc. Int. Conf. Biometrics*, 2006, pp. 273–279.

[24] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*. London, U.K.: Springer-Verlag, 2009.

[25] D. R. Ashbaugh, *Quantitative-Qualitative Friction Ridge Analysis: An Introduction to Basic and Advanced Ridgeology*. Boca Raton, FL, USA: CRC Press, 1999.

[26] W. Huang and H. Yin, "Robust face recognition with structural binary gradient patterns," *Pattern Recognit.*, vol. 68, pp. 126–140, Aug. 2017.

[27] S. Chikkerur, C. Wu, and V. Govindaraju, "A systematic approach for feature extraction in fingerprint images," in *Biometric Authentication*. Berlin, Germany: Springer, 2004, pp. 344–350.

[28] J. Bigun, "Recognition of local symmetries in gray value images by harmonic functions," in *Proc. 9th Int. Conf. Pattern Recognit.*, May/Nov. 1988, pp. 345–347.

[29] J. V. Kulkarni, B. D. Patil, and R. S. Holambe, "Orientation feature for fingerprint matching," *Pattern Recognit.*, vol. 39, no. 8, pp. 1551–1554, 2006.

[30] C. Gottschlich, B. Tams, and S. Huckemann, "Perfect fingerprint orientation fields by locally adaptive global models," *IET Biometrics*, vol. 6, no. 3, pp. 183–190, 2017.

[31] A. Sankaran, M. Vatsa, and R. Singh, "Multisensor optical and latent fingerprint database," *IEEE Access*, vol. 3, pp. 653–665, 2015.

[32] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR*, Jun. 2005, pp. 886–893.

[33] T. Dunstone and N. Yager, *Biometric System and Data Analysis*. New York, NY, USA: Springer, 2009.
[34] X. Jia, X. Yang, Y. Zang, N. Zhang, and J. Tian, "A cross-device matching fingerprint database from multi-type sensors," in *Proc. ICPR*, Nov. 2012, pp. 3001–3004.
[35] R. Cappelli, M. Ferrara, and D. Maltoni, "Minutia cylinder-code: A new representation and matching technique for fingerprint recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2128–2141, Dec. 2010.
[36] VeriFinger. (2016). *Neurotechnology*. Accessed: Dec. 5, 2015. [Online]. Available: http://www.neurotechnology.com/verifinger.html
[37] X. Si, J. Feng, J. Zhou, and Y. Luo, "Detection and rectification of distorted fingerprints," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 555–568, Mar. 2015.

HELALA ALSHEHRI received the B.S. degree in computer science from Imam Muhammad ibn Saud University, Saudi Arabia, in 2007, and the M.S. degree in computer science from King Saud University, Riyadh, Saudi Arabia, in 2011, where she is currently pursuing the Ph.D. degree in computer science with the College of Computer Science and Information Systems.

MUHAMMAD HUSSAIN received the M.Sc. and M. Phil. degrees from the University of the Punjab, Lahore, Pakistan, in 1990 and 1993, respectively, and the Ph.D. degree in computer science from Kyushu University, Fukuoka, Japan, in 2003. He is currently a Professor with the Department of Computer Science, King Saud University, Saudi Arabia. His research has been funded by the Japan Science and Technology Agency and the National Science Technology and Innovation Plan of Saudi Arabia. His current research interests include image and signal processing, pattern recognition, and computer graphics. He is an Associate Editor of the *Journal of Computer and Information Sciences*, King Saud University, (Elsevier), and has served on the program committees of various international conferences.

HATIM A. ABOALSAMH (M'18) received the Ph.D. degree in computer engineering and science from the University of Miami, USA, in 1987. He is currently a Professor and the Chairman with the Department of Computer Science, King Saud University (KSA), Riyadh, Saudi Arabia. His research interests include pattern recognition, biometrics, probability modeling, and machine learning.

He was the Vice Rector for Development and Quality, KSA, from 2006 to 2009, and the Dean of the College of Computer and Information Sciences. He is a Fellow Member of the British Computer Society, and a Senior Member of the Association of Computing Machinery, USA, and the International Association of Computer Science and Information Technology. He was the Vice President of the Saudi Computer Society. He was the Editor-in-Chief of the *KSU-Journal of Computer Sciences*. He is currently the Editor-in-Chief of the Saudi Computer Society Journal.

MANSOUR A. AL ZUAIR received the B.S. degree from King Saud University, Riyadh, Saudi Arabia, and the M.S. and Ph.D. degrees from Syracuse University, all in computer engineering. He is currently an Associate Professor with the Department of Computer Engineering, College of Computer and Information Sciences (CCIS), King Saud University.

He served as the Vice Dean for CCIS from 2009 to 2011 and the Vice Dean for Academic Affairs, CCIS, from 2011 to 2016. He has been serving as the Dean of CCIS since 2016. His research interests include computer architecture, parallel processing, embedded systems, and signal processing.