Latent Palmprint Matching

Anil K. Jain, *Fellow*, *IEEE*, and Jianjiang Feng

Abstract—The evidential value of palmprints in forensic applications is clear as about 30 percent of the latents recovered from crime scenes are from palms. While biometric systems for palmprint-based personal authentication in access control type of applications have been developed, they mostly deal with low-resolution (about 100 ppi) palmprints and only perform full-to-full palmprint matching. We propose a latent-to-full palmprint matching system that is needed in forensic applications. Our system deals with palmprints captured at 500 ppi (the current standard in forensic applications) or higher resolution and uses minutiae as features to be compatible with the methodology used by latent experts. Latent palmprint matching is a challenging problem because latent prints lifted at crime scenes are of poor image quality, cover only a small area of the palm, and have a complex background. Other difficulties include a large number of minutiae in full prints (about 10 times as many as fingerprints), and the presence of many creases in latents and full prints. A robust algorithm to reliably estimate the local ridge direction and frequency in palmprints is developed. This facilitates the extraction of ridge and minutiae features even in poor quality palmprints. A fixed-length minutia descriptor, MinutiaCode, is utilized to capture distinctive information around each minutia and an alignment-based minutiae matching algorithm is used to match two palmprints. Two sets of partial palmprints (150 live-scan partial palmprints and 100 latent palmprints) are matched to a background database of 10,200 full palmprints to test the proposed system. Despite the inherent difficulty of latent-to-full palmprint matching, rank-1 recognition rates of 78.7 and 69 percent, respectively, were achieved in searching live-scan partial palmprints and latent palmprints against the background database.

Index Terms—Palmprint, forensics, latents, minutiae, MinutiaCode, matching, region growing.

---

1 INTRODUCTION

PALMPRINT is a combination of two unique features, namely, the palmar friction ridges and the palmar flexion creases (see Fig. 1). Palmar friction ridges are the corrugated skin patterns with sweat glands but no hair or oil glands [1]. Discontinuities in the epidermal ridge patterns are called the palmar flexion creases. These are the firmer attachment areas to the basal (dermis) skin structure. Flexion creases appear before the formation of friction ridges during the embryonic skin development stage, and both of these features are claimed to be immutable, permanent, and unique to an individual [1]. The three major types of flexion creases that are most clearly visible are distal transverse, proximal transverse, and radial transverse creases. Based on these major creases, three palmprint regions are defined: interdigital, thenar, and hypothenar (see Fig. 1). Various features in palmprints can be observed at different image resolutions. While major creases can be observed at less than 100 ppi, thin creases, ridges, and minutiae can be observed only at ∼400 ppi and resolutions greater than 500 ppi are needed to observe pores.

The use of palmprints for person identification traces back to Chinese deeds of sale in the 16th century [2]. Later in 1684, Grew introduced dermatoglyphics, a study of the epidermal ridges and their arrangement on the hand. The first systematic capture of hand, finger, and palm images for identification purposes was done by Herschel in 1858 [3]. Galton [4] discussed the basis of contemporary fingerprint science, and introduced palmar ridges and creases. He suggested that the ridges on the finger tips, palms, and soles are persistent and unique. Galton defined the peculiarities in the ridges as minutiae and introduced several different minutiae types. He also divided the palm into three regions and analyzed the correlation between the ridge flow and the major creases in each region. Cummins and Midlo [2] stated that the width of a palmar ridge is 18 percent larger compared to a finger. They also recognized the significance of the flexion creases, particularly palmar flexion creases, and established the basis of the present flexion crease based identification.

While the use of Automated Fingerprint Identification Systems (AFIS) in the forensic community is pervasive, the development of automated palmprint identification systems has lagged due to the limitations of live-scan technologies for palmprints, large number of creases present in palmprints, and large storage and computing capabilities needed for processing and matching palmprints. The first reported use of palmprints in a criminal case occurred in a British court in 1931. The first automated palmprint identification system became available in the early 1990s [5]. In recent years, with advances in live-scan techniques and increase in computational power, more and more law enforcement agencies are capturing palmprints of suspects and utilizing latent palmprints for suspect and victim identification. Surveys of law enforcement agencies indicate that at least 30 percent of the prints lifted from crime scenes, called latents,—from knife hilts, gun grips, steering wheels, and window panes—are of palms, not fingers [6]. A major component of the FBI's Next Generation Identification (NGI) system is the development of an integrated national palmprint identification system [7].

- *The authors are with the Department of Computer Science and Engineering, Michigan State University, 3115 Engineering Building, East Lansing, MI 48824-1226. E-mail: {jain, jfeng}@cse.msu.edu.*
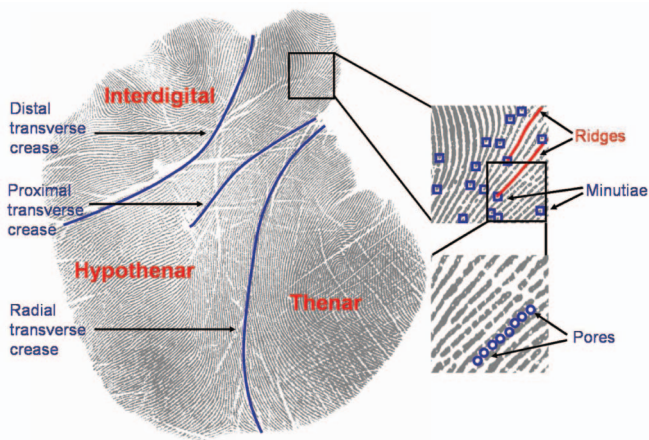
Fig. 1. Regions (interdigital, thenar, and hypothenar), major creases (distal transverse, proximal transverse, and radial transverse creases), ridges, minutiae, and pores in a palmprint.

(a)   (b)

Fig. 2. (a) Latent palmprint and (b) mated full palmprint.

Palmprint recognition systems have been developed for civilian (mainly access control) applications [8], [9]. But these systems typically utilize low-resolution (about 100 ppi) images and only support full-to-full palmprint matching. To facilitate palmprint matching, these systems use pegs to fix hand position and detect gaps between fingers for alignment. Matching is based on texture or crease information in palmprint images. In forensic applications, on the other hand, 500 ppi is the standard resolution and latent-to-full matching must be supported. When latent examiners match latent palmprints, they mainly use minutiae, whose accurate extraction requires a resolution of at least 400 ppi. Therefore, these low-resolution palmprint systems are not applicable for forensic applications. Recent work in [9] reports on a prototype image acquisition system to simultaneously acquire multispectral fingerprints and palmprints of a hand at 500 ppi. This will enable fusion of fingerprints and palmprints, which is also an objective of the FBI's NGI system in order to improve the matching accuracy.

Latent palmprint recognition shares some common problems with latent fingerprint recognition, which has been extensively studied. Some of the common attributes include complex background, poor ridge structures, and small image area. Although minutiae extraction and matching algorithms designed for fingerprints can be applied to palmprints directly, in order to achieve higher accuracy and faster matching, characteristics of palmprints should be taken into account. The first difference between fingerprints and palmprints is the presence of creases. Although creases are also frequently found in fingerprints, these creases are generally very thin and their number is small. Conventional direction field estimation algorithms [11] can reliably estimate the ridge direction in fingerprints, which is then used to remove the creases and recover the ridges. However, palmprints contain very wide creases (major creases) and a large number of thin creases, especially in the thenar area (see Fig. 3b). It is not a trivial problem to recover the ridge structure in the presence of a large number of creases. As shown in Fig. 3, VeriFinger 6.0 by Neurotechnology [12], which ranked high according to accuracy in two different
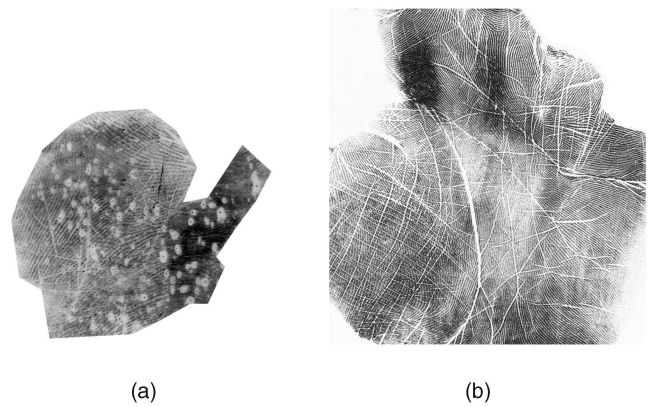
fingerprint competitions (FVC2000-2006 [13] and FpVTE 2003 [14]), and was the second best template generator in the MINEX test [15], produces many false ridges around the major crease in a palmprint (Fig. 3a), and totally fails in the palmprint area with dense thin creases (Fig. 3b). The second difference between fingerprints and palmprints is the image size. A typical full fingerprint image (500×500 pixels) contains about 100 minutiae, while a full palmprint image ($2,000 \times 2,000$ pixels) contains about 800 minutiae. A latent palmprint and its mated full palmprint are shown in Fig. 2. Assuming that the time complexity of a minutiae matcher is $O(n^2)$, where n denotes the number of minutiae in a fingerprint or a palmprint, matching palmprints will be about 64 times slower than matching fingerprints. Therefore, the computational efficiency of minutiae matching algorithm is critical for palmprint matching.

A partial-to-full palmprint matching system was proposed in [16] that used both SIFT [17] and minutiae features in matching. The system was evaluated using live-scan partial and full palmprint images. However, this system has the following limitations: 1) SIFT features cannot be consistently detected in latents and full prints, 2) the minutiae extractor and matcher (VeriFinger 4.2) used in [16] are not suitable for latent palmprint matching, and 3) latent images were not used to evaluate the algorithms.

We propose a minutiae-based latent-to-full palmprint matching system. To deal with creases in palmprints, a region growing algorithm is proposed to reliably estimate the ridge direction and frequency. To reduce the computational complexity of minutiae matching algorithm, a fixed-length minutia descriptor, MinutiaCode, is proposed, which captures information about the ridges and other minutiae in the neighborhood of a minutia. The proposed system has been evaluated by matching partial[1] palmprints (150 live-scan partial palmprints and 100 latent palmprints) against a background database of 10,200 full palmprints. Rank-1 recognition rates of 78.7 and 69 percent, respectively, were achieved in searching live-scan partial images and latents against the background database.

1. In our experiments, two types of partial palmprints, live-scan partial and latent palmprints, were used. Live-scan partial palmprints were captured using an optical palmprint scanner and latent palmprints were lifted from crime scenes. When we do not distinguish between live-scan partial and latent palmprints, they are referred to as partial palmprints.
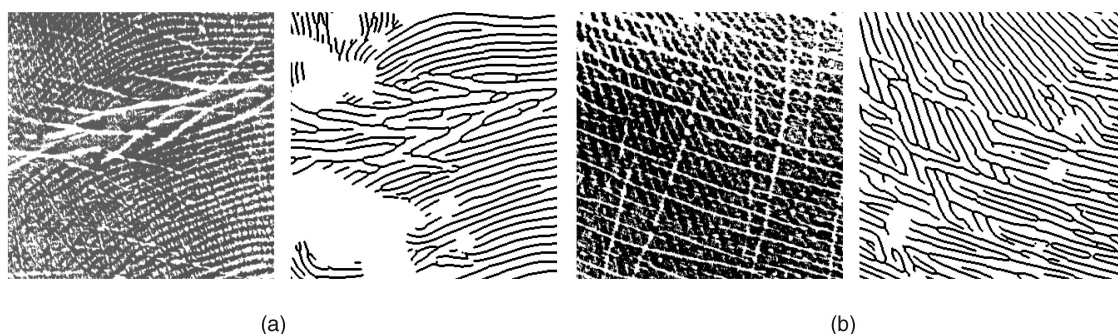
Fig. 3. Creases in palmprints. (a) A palmprint region with a major crease and its ridge skeleton image produced by VeriFinger and (b) a palmprint region with many thin creases and its ridge skeleton image produced by VeriFinger.

2   MINUTIAE EXTRACTION

The performance of a minutiae extraction algorithm relies heavily on the quality of the input palmprint images. In order to ensure that the minutiae extraction algorithm is robust with respect to the quality of the input palmprint images, an enhancement algorithm that improves the clarity of the ridge structures is necessary. Contextual filtering like 2D Gabor filters [18] has been very effective for fingerprint enhancement [19]. Two important parameters of 2D Gabor filters are local ridge direction and frequency. When these parameters are correct, Gabor filtering can connect broken ridges and separate joined ridges. However, when the parameters are incorrect, true ridges may be missed and spurious ridges may be produced after filtering. Hence, reliable ridge direction and frequency estimation is very important for minutiae extraction.

2.1   Ridge Direction and Frequency Estimation

As ridge frequency is often estimated based on ridge direction [19], reliable direction estimation is even more important. Most direction field estimation algorithms [11], [20], [21] consist of two steps: initial estimation using a gradient-based method, followed by smoothing. The smoothing may be done by a simple weighted averaging filter or more complicated model-based methods [20], [21]. These smoothing algorithms generally make two assumptions either explicitly or implicitly: 1) The direction field is smooth except for singular areas and 2) noise has a Gaussian distribution. But, for palmprints which contain a large number of creases, the initial direction field obtained by gradient-based methods significantly deviates from the true direction field and the noise cannot be modeled as Gaussian. Hence, it is very difficult for these algorithms to recover the true direction field in palmprints.

Funada et al. [22] proposed a palmprint enhancement approach, which performs image enhancement and local ridge direction and frequency estimation simultaneously. Local image blocks (8×8 pixels) are modeled by sine waves and the six strongest waves (according to amplitude) are found in each block. In the image formed by the first strongest wave in each block, continuous blocks are clustered into regions. Generally, a region contains only ridges (such region is called ridge region) or only creases (such region is called crease region). Based on certain properties, these regions are classified as ridge or crease regions, and ridge regions are

used as a single seed. A region growing algorithm is then used to grow the seed and obtain the enhanced image. The palmprint enhancement algorithm proposed in [22] has two main limitations: 1) Crease regions may be incorrectly classified as ridge regions and are grown in the region growing procedure. As a result, the objective of detecting only ridges in palmprints cannot be achieved. 2) The enhanced image is not smooth due to blocking effect and this produces spurious minutiae or leads to inaccurate estimation of the position and direction of minutiae.

We propose a palmprint enhancement approach by modifying the algorithm in [22] in the following ways: 1) Regions selected in the seed selection stage are treated as different seeds and are separately grown. Finally, one of the regions is selected as ridge region and the other regions which are compatible with the ridge region are merged with it. By postponing region classification to later stage, our algorithm can reliably remove creases and extract ridges. 2) To solve the blocking effect problem, we smooth the ridge direction and frequency obtained by the region growing algorithm and use Gabor filters to enhance the palmprint image. These two modifications significantly enhance the robustness of the minutiae extraction algorithm and lead to better recognition accuracy.

We now describe our ridge direction and frequency estimation algorithm, which is composed of four main steps.

2.1.1   Sine Wave Representation

A palmprint image $I(x, y)$ is divided into nonoverlapping blocks of 16×16 pixels. Let H and W denote the height and width of the image, and N_H and N_W denote the number of blocks in the vertical and horizontal directions, respectively. Since the ridge structure of a block can be approximated by a 2D sine wave, the task of estimating local ridge direction and frequency is transformed to estimating the parameters of sine wave in each block. Centered at each block, the local image in the 64×64 window is multiplied by a Gaussian function ($\sigma = 16$). The larger window (64×64 pixels) has the following two advantages over the smaller window (16×16 pixels): 1) it is more robust to noise and 2) the resolution in the frequency domain is higher. The Discrete Fourier Transform (DFT), $F(u, v)$, of the resulting image is computed and the amplitude of low-frequency components (points within 3 pixels from the center in the frequency domain) is set to 0. Three pixels in the frequency domain correspond to a ridge
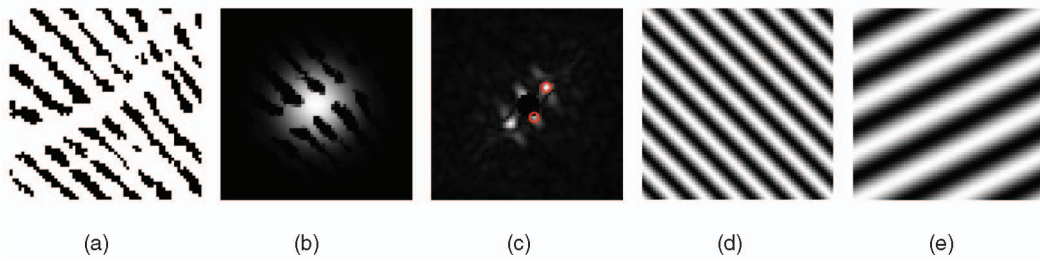
Fig. 4. Sine wave representation. (a) Local gray image (64×64 pixels), (b) local gray image multiplied by Gaussian function, (c) two points with the highest amplitude in the frequency image, (d) the first sine wave, and (e) the second sine wave.

period of 21.3 pixels in the palmprint image. In the frequency domain, six points with the maximum amplitude are found. Each of these points corresponds to a 2D sine wave $w(x, y) = a * sin(2\pi f(cos(\theta)x + sin(\theta)y + \phi)$, where a, f, θ, and ϕ represent the amplitude, frequency, direction, and phase, respectively. These waves are sorted in the decreasing order of amplitude and are referred to as the first wave, the second wave, ..., and the sixth wave. The above steps are shown in Fig. 4. The parameters of the sine wave at position (u, v) are computed as

$$a = |F(u, v)|, \tag{1}$$

$$f = \frac{\sqrt{u^2 + v^2}}{64}, \tag{2}$$

$$\theta = \arctan\left(\frac{u}{v}\right), \quad \text{and} \tag{3}$$

$$\phi = \arctan\left(\frac{Im(F(u, v))}{Re(F(u, v))}\right). \tag{4}$$

When a local image contains only ridges, the DFT has a single strong peak which corresponds to the ridges. When the local image contains both ridges and creases, the DFT has multiple strong peaks. Fig. 5 shows the six strongest

waves of three types of local palmprint images: 1) no crease, 2) creases with one direction, and 3) creases with two directions. As shown in Fig. 5, it is not easy to reliably determine which wave corresponds to ridges based on the local information alone, namely the amplitude. The basic idea of the proposed algorithm is to utilize the fact that waves corresponding to ridges form continuous and sufficiently large clusters.

Two adjacent waves (namely, waves in adjacent blocks) w_1 and w_2 are said to be continuous if the following three conditions are satisfied:

$$Angle(\theta_1, \theta_2) \leq \pi/6, \tag{5}$$

$$\left| \frac{1}{f_1} - \frac{1}{f_2} \right| \leq 3, \quad \text{and} \tag{6}$$

$$\frac{1}{16} \sum_{(x,y)\in L} \left| \frac{w_1(x, y)}{a_1} - \frac{w_2(x, y)}{a_2} \right| \leq 0.8, \tag{7}$$

where $Angle(\theta_1, \theta_2)$ computes the angle $\Delta\theta$ $(0 \leq \Delta\theta \leq \pi/2)$ between two directions θ_1 and θ_2, and L denotes the 16 pixels on the border of two adjacent blocks. The above three conditions measure the continuity of direction, frequency, and normalized gray-scale values between two adjacent waves, respectively. These thresholds were
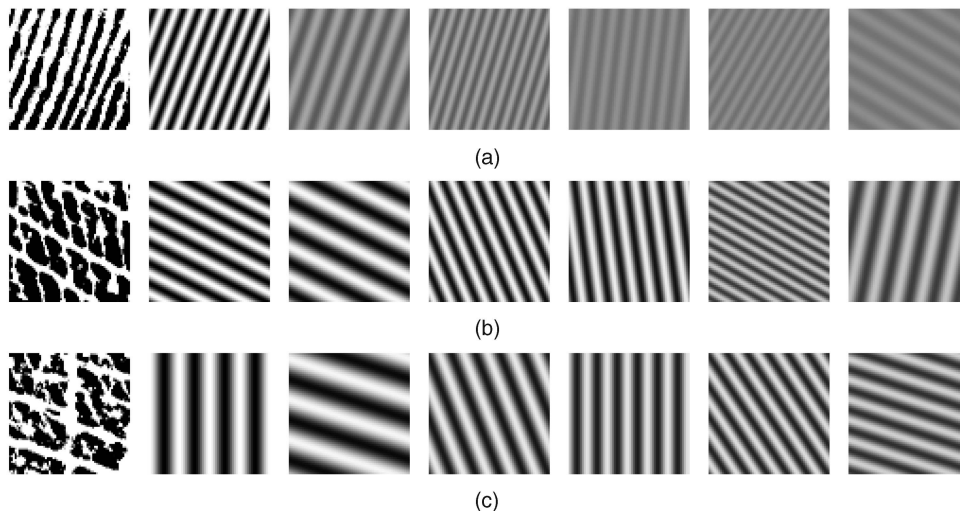

Fig. 5. The six strongest sine waves corresponding to three types of local regions (64×64 pixels) in a palmprint: (a) no crease, (b) creases with one direction, and (c) creases with two directions. In these three local regions, the sine wave corresponding to ridges is the first, the third, and the third one of the six waves, respectively.

Fig. 6. Continuity of adjacent waves. (a) Two waves are continuous, (b) the direction of two waves is discontinuous, (c) the frequency of two waves is discontinuous, and (d) the normalized gray-scale values of two waves are discontinuous.

set by analyzing the values of a set of positive samples manually selected, namely two adjacent waves both of which correspond to ridges. About 87 percent of positive samples satisfy the chosen thresholds. Very small threshold values will result in many small regions and therefore make it difficult to distinguish ridges from creases. Very large values of threshold will cause ridges and creases to merge and therefore also make it difficult to separate ridges and creases. A pair of continuous adjacent waves, which satisfies the above three conditions, and three pairs of discontinuous adjacent waves, which do not satisfy one of the three conditions above, are shown in Fig. 6.

2.1.2 Seed Selection

The reliability of the first wave of a block is computed as $a_1/(a_1 + a_2)$, where a_i denotes the amplitude of the ith wave. The first wave of a block is deemed as reliable if its reliability is greater than a predefined threshold (0.67). By observing a few images of low quality, a minimum reliability that guarantees that at least one correct seed, which is a ridge region, can be generated is selected as the threshold. A smaller threshold value may increase the number of seeds and thereby increase the computational complexity. On the other hand, a larger threshold value may miss some ridge regions. A reliable first wave is represented by a node in a graph. The adjacent nodes (waves) that are continuous are connected by edges. All connected components with more than 20 nodes in the graph are used as seeds and the seeds are sorted in the decreasing order of size (the number of blocks). An auxiliary image of $N_H \times N_W$ pixels, $I_S(m, n)$, is created to record the seed index of each block. $I_S(m, n)$ is 0 for the blocks that do not belong to any seed. The seed selection algorithm is illustrated in Fig. 7. Seeds selected in this step

may include both ridge and crease regions. For instance, one of the three seeds in Fig. 7 is a crease region.

2.1.3 Region Growing

Each seed is grown in turn by a region growing algorithm (see pseudocode RegionGrow). The three inputs to this algorithm are s_k, I_S, and I_W. s_k denotes the index of the current seed. I_S is an image of $N_H \times N_W$ pixels that is used to record the seed index of all blocks. I_W is an image of $N_H \times N_W$ pixels that is used to represent the selected waves in the current region. $I_W(m, n) = i, i = 1, 2, \ldots, 6$ indicates the ith wave is selected in block (m, n). $I_W(m, n) = 0$ indicates no wave is selected in block (m, n). Initially, the current region consists of only the current seed, namely, $I_W(m, n) = 1$ for blocks belonging to the current seed and 0 for the remaining blocks. The region growing algorithm iteratively selects waves in new blocks that are continuous with the current region, and adds them to the current region, until no more waves can be added. An example is given in Fig. 8 to show the region growing process for a seed that is a ridge region and a seed that is a crease region.

Function RegionGrow(s_k, I_S, I_W)
    $Q \leftarrow \emptyset$;
    for *each pixel (m, n) in image I_W* **do**
        if $I_W(m, n) > 0$ **then**
        FindCandidateWaves(I_W, m, n);
    end
    while $Q \neq \emptyset$ **do**
        Pop up a candidate wave $w_c = (m, n, i)$ from Q;
        if $I_W(m, n) > 0$ **then** continue;
        $I_W(m, n) = i$;
        FindCandidateWaves(I_W, m, n);
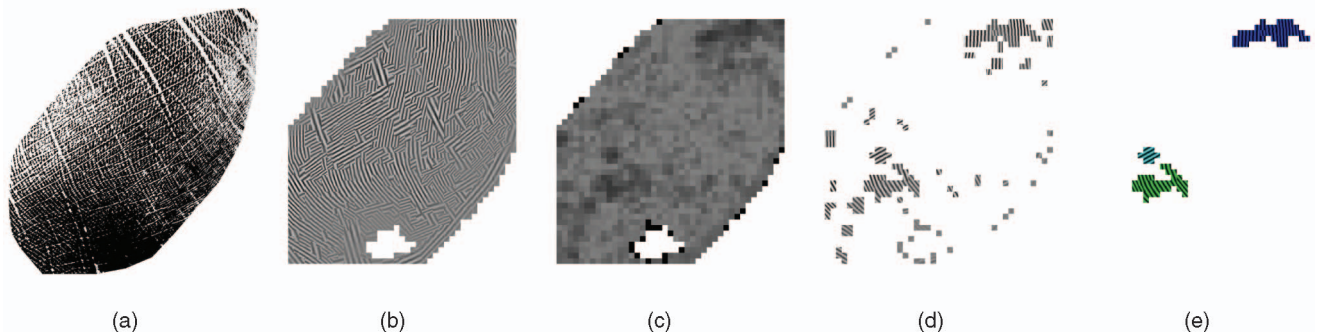   **if** $i = 1$ & $I_S(m, n) > s_k$ **then**

Fig. 7. Seed selection. (a) A live-scan partial print (height: 765 pixels, width: 717 pixels) from the thenar region, (b) first wave image, (c) reliability image (low gray value indicates high reliability), (d) reliable first waves, and (e) three seeds (two ridge regions and one crease region) extracted from (d).
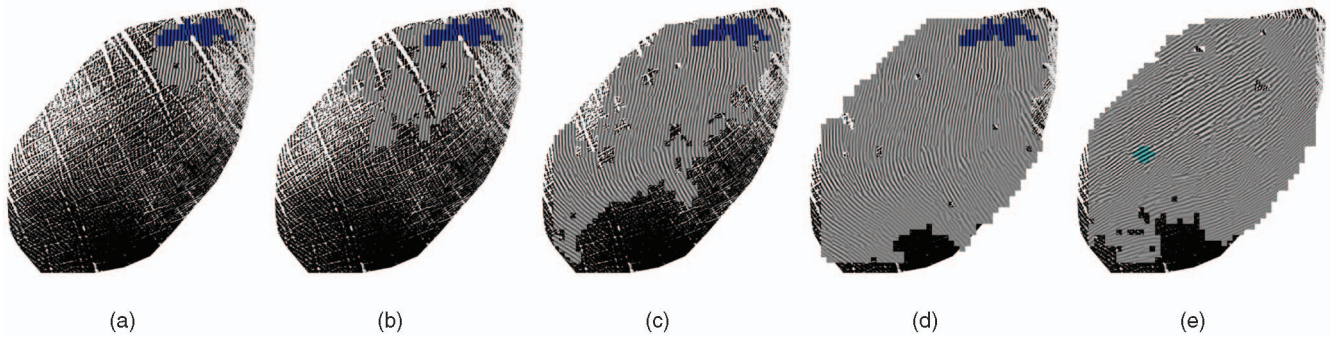
Fig. 8. Region growing process of the image in Fig. 7. (a), (b), (c), and (d) are four intermediate steps in growing one seed that is a ridge region and (e) is the region grown from another seed that is a crease region.

Merge seed $I_S(m, n)$ with the current region;
   end
  end

The region growing algorithm starts by finding candidate waves (see pseudocode FindCandidateWaves). For each block (m, n) of the current region, candidate waves are found in its 4-connected neighbors which do not belong to the current region. In a neighboring block, (m', n'), each of the six waves is checked in the decreasing order of amplitude if it is continuous with the wave of block (m, n). If the ith wave in block (m', n') is continuous with the wave of block (m, n), it is referred to as a candidate wave. A record about this candidate wave, $w_c = (m', n', i)$, is added to a priority queue, Q, where i is the priority value and the first wave has the highest priority.

The algorithm iteratively pops up a candidate wave $w_c = (m, n, i)$ from Q and processes it until Q is empty. If the wave of block (m, n) has been selected, pop up and process the next candidate wave in Q; otherwise, the ith wave is selected for block (m, n) and we find candidate waves in its 4-connected neighbors. In addition, we check whether $i = 1$ and $s_l = I_S(m, n) > s_k$. If yes, this wave also belongs to another seed s_l and we merge seed s_l with the current region by performing the following steps: 1) all pixels of I_W corresponding to seed s_l are set as 1, 2) seed s_l is made invalid by setting all pixels in I_S corresponding to seed s_l to 0, and 3) candidate waves are found based on the blocks of seed s_l.

Function FindCandidateWaves(I_W, m, n)
  for *each 4-connected block (m', n') of block (m, n)*
  do
     if $I_W(m', n') > 0$ **then** continue;
     for $i \leftarrow 1$ **to** 6 **do**
       if *the ith wave is continuous with the wave in block (m, n)* **then**
         // Assume Q can be accessed in this function
         Add candidate wave $w_c = (m', n', i)$ to priority queue Q;
         break;
       end
     end
  end

2.1.4 Region Merging

After region growing is performed for each seed, a set of regions is obtained. These regions are merged into a final region by first sorting in the decreasing order of the number of reliable first waves. The first region is deemed as a ridge region and copied to the final region. Then, the other regions are checked in turn to see if they have different waves in the overlapped blocks with the final region. If waves are not different, this region is deemed compatible with the final region and is copied to the final region; otherwise next region is checked.

For the example shown in Fig. 8, the region grown from the seed that is a ridge region is correctly selected as it contains more reliable first waves than the region grown from the seed that is a crease region. Fig. 9 compares the ridges extracted by VeriFinger 6.0 [12], the algorithm in [22], and the proposed algorithm for a live-scan partial print from the thenar region and a latent print, respectively. This comparison shows that two region growing based algorithms (the algorithm in [22] and the proposed algorithm) are more robust than VefiFinger 6.0 for ridge direction estimation in the presence of creases. The algorithm in [22] failed to remove some creases which the proposed algorithm successfully removed (see Figs. 9c and 9d) and produced more spurs than the proposed algorithm (see Figs. 9g and 9h).

2.2 Minutiae Extraction

Given local ridge direction and frequency, a sequence of image processing steps is performed to extract the minutiae: enhancement, binarization, thinning, and ridge and minutia extraction. The extracted minutiae include many spurious minutiae due to image noise, which are removed in the following way: The ridge validation procedure in [23] is used to classify ridges as reliable or unreliable and the minutiae associated with unreliable ridges are removed. The remaining minutiae are further classified as reliable or unreliable minutiae. A minutia is deemed as unreliable if it forms an opposite pair with other minutia in the neighborhood; otherwise, it is deemed as reliable. An opposite pair is a pair of minutiae that are close to each other but have opposite directions. Both reliable and unreliable minutiae are used in the proposed matching algorithm, but treated differently. The results of different steps in minutiae extraction are shown in Fig. 10. It should be noted that due to complex background and multiple
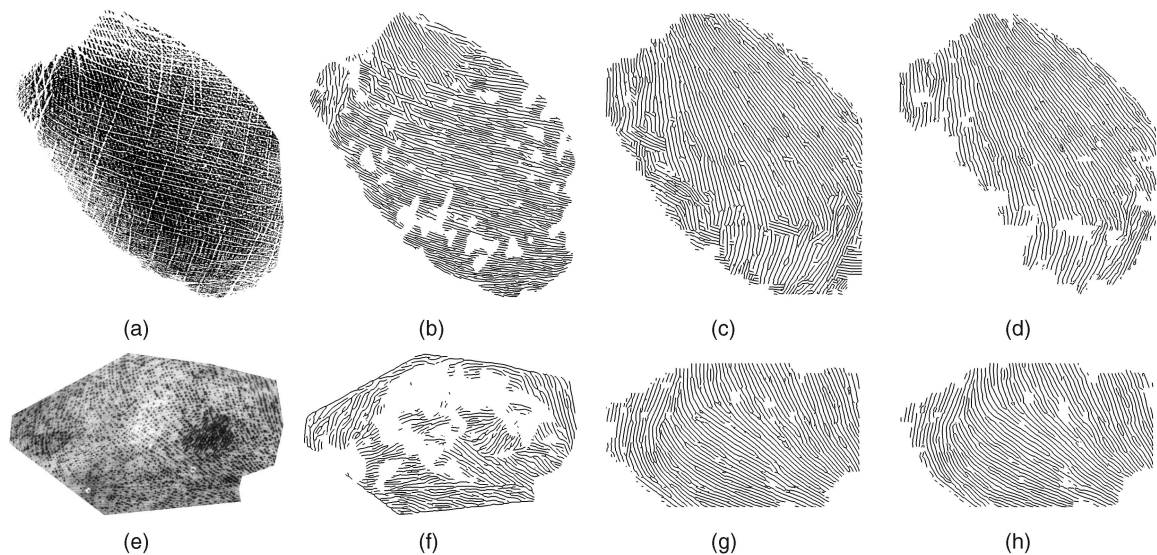
Fig. 9. Comparison of VeriFinger 6.0, the algorithm in [22], and the proposed algorithm for ridge detection. (a) A live-scan partial print (height: 973 pixels, width: 893 pixels) from the thenar region, (b) skeleton image of (a) by VeriFinger, (c) skeleton image of (a) by the algorithm in [22], (d) skeleton image of (a) by the proposed algorithm, (e) a latent print (height: 523 pixels, width: 886 pixels), (f) skeleton image of (e) by VeriFinger, (g) skeleton image of (e) by the algorithm in [22], and (h) skeleton image of (e) by the proposed algorithm.

overlapping latent prints in a single latent image, the region of interest (ROI) is manually marked for latent palmprints. This is a common practice in forensics. But, for other images (full and live-scan partial palmprints), no manual intervention is needed.

3 MINUTIAE MATCHING

Given the minutiae features of two palmprints, the matching algorithm consists of 1) local minutiae matching—the similarity between each minutia of a partial print and each minutia of a full print is computed, 2) global minutiae matching—using each of the five most similar minutia pairs in step 1) as an initial set, a greedy matching algorithm is used to find additional matching minutia pairs, and 3) matching score computation—a matching score is computed for each set of matching minutia pairs and the

maximum score is used as the matching score between two palmprints.

3.1 Local Minutiae Matching

A minutia is generally tagged with the following features: location, direction, type (ending or bifurcation), and quality (reliable or unreliable) [23]. Since the relative transformation between the two palmprints to be matched is not known a priori and considering the large size of palmprint images, the minutiae correspondence problem is very challenging. To reduce the ambiguity in matching, we attach additional distinguishing information to a minutia in the form of a minutia descriptor. In the fingerprint recognition literature, four types of information have been widely used as minutia descriptors, namely image intensity [24], texture [25], ridge information [26], and neighboring minutiae [27], [28]. Among these four types of descriptors, texture and minutiae-based descriptors are known to
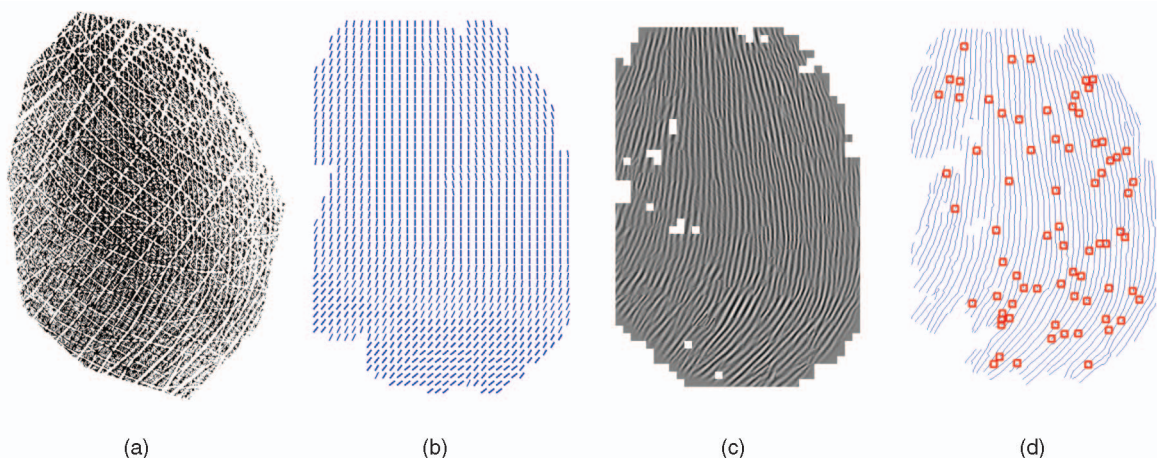

Fig. 10. Minutiae extraction. (a) A live-scan partial print (height: 636 pixels, width: 578 pixels) from the thenar region, (b) direction field, (c) enhanced image, and (d) extracted ridge and minutiae.
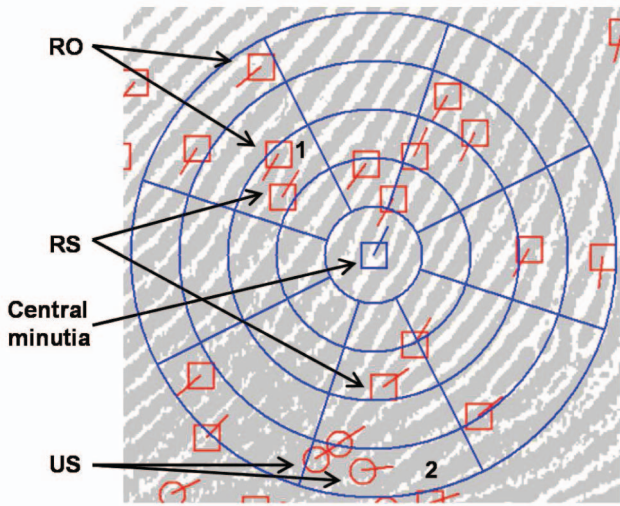
Fig. 11. The configuration of a MinutiaCode. The numbers of four types of neighboring minutiae, RS, US, RO, and UO, in sectors 1 and 2 are [1 0 1 0] and [0 2 0 0], respectively. Square indicates reliable minutiae and circle indicates unreliable minutiae.

provide good performance and a combination of texture and neighboring minutiae information can achieve higher accuracy [29]. However, the length of the neighboring minutiae-based descriptor in [29] is variable, depending on the number of neighboring minutiae. Computing the similarity between two variable-length minutiae descriptors is not very efficient. Therefore, a fixed-length minutia descriptor, called MinutiaCode, that captures neighboring texture and minutiae information is proposed here.

The MinutiaCode of a minutia (referred to as central minutia) is constructed as follows. The circular region around a central minutia is divided into $(R-1) \times K$ sectors by $R = 5$ concentric circles and $K = 8$ lines as illustrated in Fig. 11. The radius of the rth circle, $1 \leq r \leq R$, is $20 * r$ pixels. The direction of the kth line, $1 \leq k \leq K$, is $\theta + (k-1) \cdot \pi/K$, where θ denotes the direction of the central minutia. For each sector, a set of features is computed, including the quality (1: foreground, 0: background), mean ridge direction, mean ridge period, and the numbers of four types of neighboring minutiae. These four types of neighboring minutiae are defined as

1. reliable and with the same direction as the central minutia (RS),
2. unreliable and with the same direction as the central minutia (US),
3. reliable and with the opposite direction to the central minutia (RO), and
4. unreliable and with the opposite direction to the central minutia (UO).

Whether a neighboring minutia has the same or opposite direction to the central minutia is determined by the angle between the direction of the neighboring minutia and the direction of the central minutia. If the angle is less than $\pi/2$, the neighboring minutia has the same direction to the central minutia; otherwise, it has opposite direction to the central minutia. See Fig. 11 for the numbers of four types of

neighboring minutiae in two of the 32 sectors (excluding the central part).

The similarity s between two MinutiaCodes is defined as the weighted average value of the similarities of all valid sectors. A pair of corresponding sectors is deemed valid if both sectors are in the foreground. If the number of the valid sectors is less than 16, s is set to 0; otherwise s is computed by

$$s = \frac{1}{\sum_{i=1}^{32} w_i} \sum_{i=1}^{32} w_i s_i, \tag{8}$$

where s_i denotes the similarity of the ith sector and w_i denotes the weight of the ith pair of corresponding sectors. To assign a larger weight to sectors containing more reliable minutiae, w_i is defined as $(max(n_1, n_2) + w_0)$, where n_1 and n_2 are the number of reliable minutiae in the two corresponding sectors and w_0 is a weight for sectors without reliable minutiae (set to 0.2 in our experiments).

The similarity s_i between two corresponding sectors is computed as follows. If the difference between ridge directions or the difference between ridge periods is greater than the corresponding threshold ($\pi/6$ and 3 pixels), s_i is set to 0; otherwise, s_i is computed using the following formulas:

$$s_i = \frac{n_M}{n_S}, \tag{9}$$

$$n_M = n_{MS} + n_{MO}, \tag{10}$$

$$n_S = n_{SS} + n_{SO}, \tag{11}$$

$$n_{MS} = min(n_{RS1} + n_{US1}, n_{RS2} + n_{US2}), \tag{12}$$

$$n_{MO} = min(n_{RO1} + n_{UO1}, n_{RO2} + n_{UO2}), \tag{13}$$

$$n_{SS} = max(n_{RS1}, n_{RS2}, n_{MS}), \quad \text{and} \tag{14}$$

$$n_{SO} = max(n_{RO1}, n_{RO2}, n_{MO}), \tag{15}$$

where the description of the symbols is given in Table 1. The range of s_i is [0, 1]. If n_M is equal to n_S, s_i is maximum (1). If $n_M = 0$ and $n_S > 0$, s_i is minimum (0). If $n_M = 0$ and $n_S = 0$ (namely, there is no minutiae which should be matched in the two sectors), s_i is set to 1. To evaluate the robustness of MinutiaCode to noise, the matching results of five correctly identified latents of low quality are examined. The lowest quality image where MinutiaCode still succeeds is when there are five correctly detected minutiae and seven spurious minutiae in the neighborhood. This successful matching should be attributed to the use of texture information and the different ways we deal with reliable and unreliable minutiae.

3.2 Global Minutiae Matching

Given the similarity of all minutia pairs, the one-to-one correspondence between minutiae is established in this stage. All minutia pairs are sorted in the decreasing order of normalized similarity defined in [23] and each of the top-five minutia pairs is used to align the two sets of minutiae. Minutiae are examined in turn and minutiae that

TABLE 1
Symbols Used in the Computation of the Similarity between Two MinutiaCodes

Symbol	Description
s_i	Similarity between the ith pair of corresponding sectors
n_M	Number of the matched minutiae
n_S	Number of the minutiae that should be matched
n_{MS}	Number of the matched minutiae of type RS and US
n_{MO}	Number of the matched minutiae of type RO and UO
n_{SS}	Number of the minutiae of type RS and US that should be matched
n_{SO}	Number of the minutiae of type RO and UO that should be matched
n_{RSj}	Number of the minutiae of type RS in partial ($j = 1$) or full ($j = 2$) palmprint
n_{USj}	Number of the minutiae of type US in partial ($j = 1$) or full ($j = 2$) palmprint
n_{ROj}	Number of the minutiae of type RO in partial ($j = 1$) or full ($j = 2$) palmprint
n_{UOj}	Number of the minutiae of type UO in partial ($j = 1$) or full ($j = 2$) palmprint

are close in both location and direction, and have not been matched to other minutiae are deemed as matching minutiae. After all the minutia pairs have been examined, a set of matching minutiae is obtained.

3.3 Matching Score Computation

The matching score S between two palmprints is computed as

$$S = W_m * S_m + (1 - W_m) * S_d, \qquad (16)$$

where S_m and S_d denote the minutiae-based matching and the direction field based matching scores, respectively; the weight W_m is empirically set to 0.8.

The minutiae-based matching score S_m is the product of a quantitive score S_{mn} and a qualitative score S_{mq}. The quantitive score measures the quantity of evidence and the qualitative score measures the consistency in the common region between two palmprints. The quantitive score S_{mn} is computed as $M/(M + 20)$, where M denotes the number of matched minutiae and the value 20 is an estimate of the minimum number of matching minutiae for genuine matches. The qualitative score is computed as

$$S_{mq} = S_D \times \frac{M}{M + N_L} \times \frac{M}{M + N_F}, \qquad (17)$$

where S_D is the average similarity of descriptors for all the matching minutiae, and N_L and N_F, respectively, denote the number of unmatched minutiae in latent and full prints that are reliable and belong to the common region of the two palmprints.

The direction field based matching score S_d is the product of a quantitive score S_{dn} and a qualitative score S_{dq}. The quantitive score S_{dn} is computed as $N_b/(N_b + 900)$, where N_b is the number of blocks where the difference of direction between latent and full print is less than $\pi/8$, and the number 900 (of 16×16 blocks) is an estimate of the minimum common (or overlapping) area for genuine matches. The qualitative score S_{dq} is computed as $(1 - 2 * D_d/\pi)$, where D_d is the mean of the difference of direction values of all of the blocks.

4 EXPERIMENTS

4.1 Palmprint Database

There is no public domain latent and mated full palmprint database available. Further, to our knowledge, while there have been several large-scale performance evaluations organized by NIST for fingerprint (FpVTE [14] and ELFT [30]), face (FRVT [31]) and iris (ICE [32]), such performance evaluation of latent/partial palmprint matching algorithms has not yet been conducted. The announcement of the FBI's NGI program has created a substantial interest in palmprint matching and it is likely that a similar evaluation for palmprint recognition will be conducted in the near future. In our experiments, we used two sets of latent palmprints provided to us by Noblis [33] and the Forensic Science Division of Michigan State Police (MSP). The Noblis latent database consists of 46 latent palmprints which correspond to eight different palms. The MSP latent database consists of 54 latent palmprints which correspond to 22 of the 36 different palms. The latents from Noblis and MSP have been merged to form a database of 100 latents. Michigan State Police also provided us with 10,040 full palmprints that are used to form a background database for latent matching. Due to the limited number of latent palmprints available to us, we also collected live-scan partial palmprints and their mated full palmprints using a CrossMatch L SCAN 1000P optical scanner in our laboratory. Live-scan partial images were collected from 50 unique palms (25 subjects who provided images of both left and right palms) with three impressions per palm, one impression each from the thenar, hypothenar, and interdigital regions. Full prints of these 50 palms and other 66 palms were also scanned. The live-scan partial images and the latent images were not merged, since they are quite different, both in size and quality. The 116 live-scan full palmprints, 44 (eight from Noblis and 36 from MSP) mated full palmprints of latents, and 10,040 full palmprints from the Michigan Forensic Laboratory were merged to form a background database of 10,200 full palmprints. In our databases, most images (the 10,040 full prints from the Michigan Forensic Laboratory) are at 500 ppi; remaining images were either downsampled or upsampled to 500 ppi using bicubic interpolation. Our partial and full palmprint databases are summarized in Tables 2 and 3, respectively.

TABLE 2
Partial Palmprint Databases

	Noblis latent database	MSP latent database	MSU live-scan partial database
Resolution (ppi)	1000	400	1000
No. of partial prints	46	54	150
No. of palms	8	36	50
Interdigital region	23	24	50
Thenar region	3	14	50
Hypothenar region	21	22	50

The sum of the latents from the three regions (interdigital, thenar, and hypothenar) may be greater than the total number of the latents, as some latents contain data from more than one region.

TABLE 3
Full Palmprint Databases

	Noblis mated full database	MSP mated full database	MSP full database	MSU live-scan full database
Resolution (ppi)	1000	400	500	1000
No. of palms	8	36	10,040	116

4.2 Matching Performance on Full Background Database

Due to the differences in the nature and quality of live-scan partial and latent palmprints, we separately searched them against the full background database. Since a large number of full prints are not oriented properly, no rotation constraint is used in the minutiae matching algorithm. Hand type information (left or right hand) is utilized if this information can be reliably estimated from partial palmprints. 55 latents among all 100 latents have hand type information and all 150 live-scan partial images have hand type information. The hand type information for all the full prints was already available with the images. The number of left and right hands in the background database is roughly equal. The CMC curves for searching 100 latents and 150 live-scan partial images against 10,200 full prints are shown in Fig. 12. The rank-1 recognition rates of 78.7 and 69 percent, respectively, were achieved for live-scan partial and latent palmprints. As
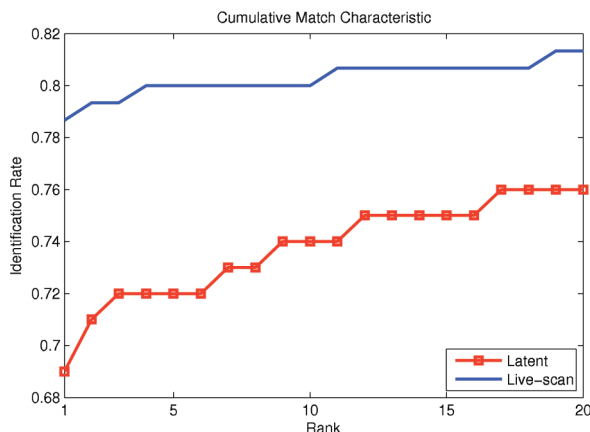

Fig. 12. CMC curves for latent and live-scan partial palmprint identification with a background database of 10,200 full prints. The number of latents is 100 and the number of live-scan partial palmprints is 150. The curves are not smooth due to the small number of partial images.

expected, the performance for live-scan partial images is much better than that for latents due to better image quality and larger image size of the former. There are two things that should be noted. In forensic applications, latent experts generally manually correct minutiae extracted by algorithms. With intervention of latent experts, the matching accuracy can be significantly improved. In practice, latent experts generally examine top 20 candidates provided by the automated system, and in high profile cases such as murder, latent experts may examine as many as 100 candidates. As shown in Fig. 12, the rank-20 recognition rates of 81.3 and 76 percent, respectively, were achieved for live-scan partial and latent palmprints.

4.3 Comparison to Other Algorithms

The proposed palmprint enhancement algorithm has two main improvements over the original algortihm of Funada et al. [22], namely, more robust direction field estimation and elimination of blocking effect, which have been qualitatively shown in Fig. 9. To evaluate the proposed improvements quantitatively, we combined the two enhancement algorithms with the same minutiae extraction and matching algorithms proposed in this paper. An experiment was conducted by searching 100 latents and 150 live-scan partial palmprints against a background database of 160 full prints which consists of the 44 mated full prints of latents and 116 live-scan full prints. Hand type information was not used in matching. The CMC curves of the two enhancement algorithms for two types of partial images are given in Fig. 13. This figure indicates that the improved algorithm provides higher palmprint matching accuracy than the original algorithm in [22].

Since, to our knowledge, there is no partial-to-full palmprint matching algorithm available in the open literature, we compared our matching algorithm to a commercial fingerprint SDK, Neurotechnology VeriFinger. However, VeriFinger cannot be directly used for palmprint matching because it has a limit on the number of minutiae
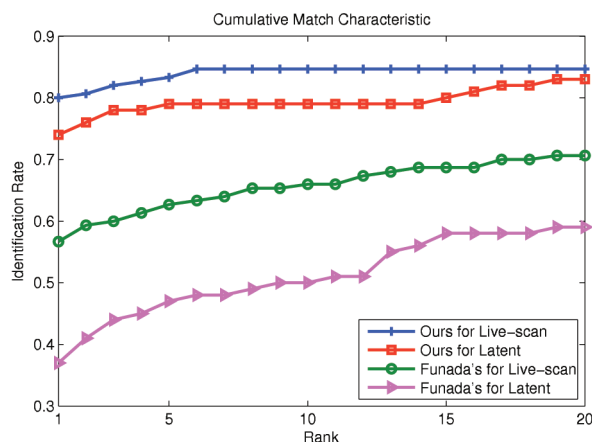
Fig. 13. CMC curves of using two different palmprint enhancement algorithms (Funada's [22] and ours) in searching 100 latents and 150 live-scan partial palmprints against a background database of 160 full prints.

Fig. 14. Region map. (a) A full palmprint and (b) its region map.

that can be dealt with in feature extraction and matching and this limit is smaller than the number of minutiae observed in full palmprints. In [16], full palmprints are split into five sectors and minutiae are extracted separately for each sector using VeriFinger. No minutiae are extracted from the central part of the palms, as this part is less frequently found in latents. Major creases are extracted and minutiae around the major creases are removed. After these steps, VeriFinger matcher can be used for partial-to-full palmprint matching. A rank-1 recognition rate of 67.5 percent was reported in [16] when matching 240 live-scan partial (which are from 20 palms of all of the 50 palms in the MSU live-scan partial database) against 100 live-scan full prints (which is a subset of the MSU live-scan full database). The rank-1 rate (67.5 percent) of VeriFinger on a small background database (100 full prints) is much lower than the rank-1 rate (78.7 percent) of the proposed algorithm on a much larger background database (10,200 full prints).

4.4 Utilization of Ancillary Information

Given a latent palmprint, proficient latent examiners can often reliably estimate (depending on the quality of the latent image) the hand (left or right) that made the latent, the part of the palm that the latent was from, and the orientation of the latent [34]. To determine the matching performance gain in the presence of such information, an experiment was conducted by searching 100 latents against a small background database of 160 full prints which consists of the 44 mated full prints of latents and 116 live-scan full prints. A small background database is selected as it is not a trivial task to automatically extract ancillary information for the full background database where a large number of palmprints are of poor quality and not in upright position. For the 160 full prints, region map and palm orientation were manually marked by the authors. The region map of a full palmprint is shown in Fig. 14. The region map is a 3-bit-depth image of the same size as the palmprint image, where one of the three bits of each pixel is used to record which of the three palmprint regions it belongs to. The different regions are allowed to have some overlap in order to account for errors in
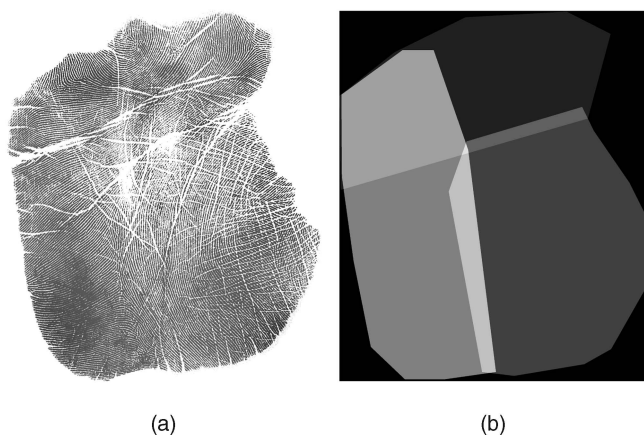
marking the region map for latents. For the 100 latents, hand type, region map, and orientation are estimated by the authors using the methods described in [34]. Due to the poor quality of latents, hand type cannot be estimated for 45 latents and palm orientation cannot be estimated for 27 latents. Fig. 15 shows one example for each of the following three situations: 1) The ancillary information can be reliably estimated, 2) no ancillary information can be reliably estimated, and 3) partial ancillary information can be reliably estimated. The ancillary information is utilized in the minutiae matcher in the following way: 1) the similarity between palmprints of different hand types (left versus right) is 0; 2) the similarity between two minutiae of
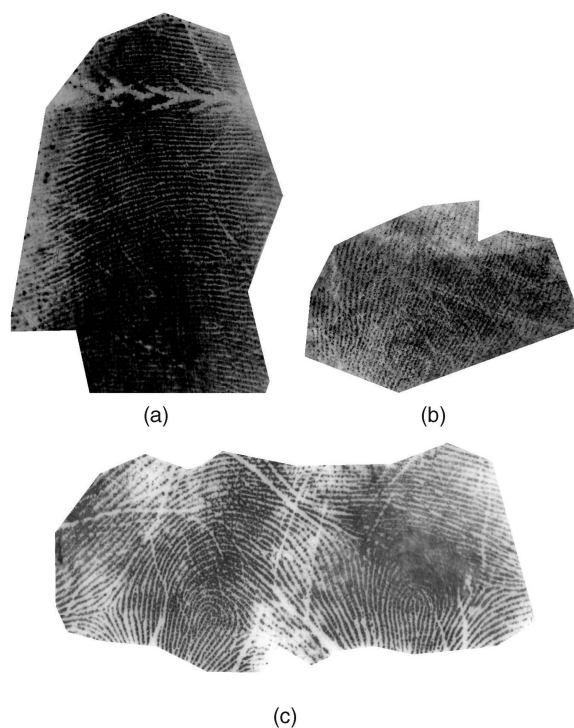

Fig. 15. Estimating latent palmprint ancillary information. (a) Ancillary information can be reliably estimated (left hand, hypothenar region, upright orientation), (b) none of the ancillary information can be reliably estimated, and (c) partial ancillary information can be reliably estimated (unknown hand, interdigital region, reverse orientation).
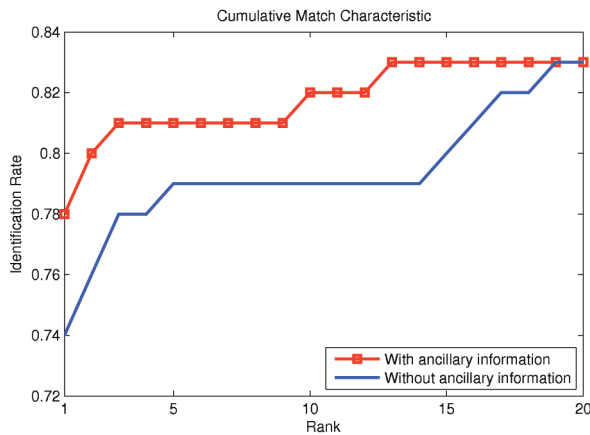
Fig. 16. CMC curves for latent palmprint matching (100 latents) with and without ancillary information against a background database of 160 full prints.
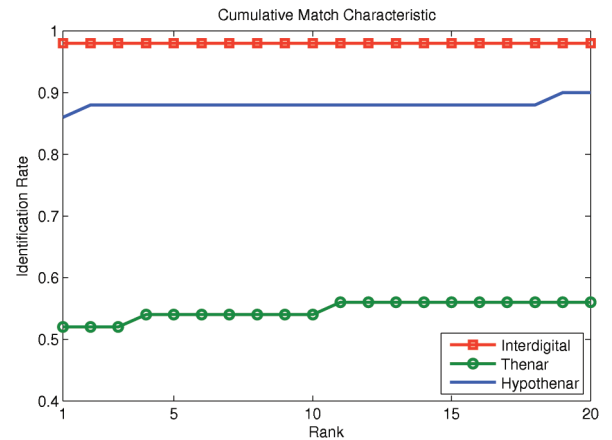

Fig. 17. CMC curves for matching live-scan partial images from three different palm regions against the background database of 10,200 full prints. The numbers of the partial images from the three palm regions are the same (50).

different palm regions (e.g., interdigital region versus thenar region) is 0; 3) the similarity between two minutiae whose direction (with respect to palm orientation) difference is greater than a threshold ($\pi/3$) is 0. Fig. 16 shows the CMC curves demonstrating the performance improvement in the presence of ancillary information. The improvement of rank-1 identification rate due to the use of ancillary information indicates that such information is quite useful. The matching speed with ancillary information is also about 2.5 times faster than without such information. The rank-20 identification rates with and without ancillary information in Fig. 16 are the same due to the small size of the background database.

4.5 Different Palm Regions

To examine the identification performance of different palm regions, we computed three separate CMC curves (see Fig. 17) for matching images from the three palm regions in the 150 live-scan partial images against the background database of 10,200 full prints. The thenar region was found to be the most challenging palmprint region with a rank-1 recognition accuracy of only 52 percent, which is much lower than the accuracy of the interdigital region (98 percent) and the hypothenar region (86 percent). The low accuracy for the thenar region is due to the presence of a large number of creases in the thenar region and the smaller size of the images from the thenar region. During our collection of live-scan partial palmprints, we intended to exclusively capture each of the three regions. However, it is not easy to scan the thenar region alone without interference of the other two regions due to the structure of the thenar region. Therefore, only a part of the thenar region, which is characterized by a large number of creases, is scanned. As a result, the size of the images from the thenar region is smaller than that of the images from the other two regions. The superior performance of the interdigital to the hypothenar is due to 1) the direction field in the interdigital region is more distinctive than that in the hypothenar region) and 2) some of the hypothenar images in our database contain the edge of the palm where the ridge pattern is not present. Since just using partial palmprints from the interdigital region can achieve a rank-1

recognition rate of 98 percent, we can predict that the rank-1 recognition accuracy of full-to-full palmprint matching (searching the full prints of the live-scan partial palmprints against the full background database) using the proposed algorithm should be greater than or equal to 98 percent.

4.6 Quality of Latents

We manually classified the 100 latent palmprints into three different quality levels: good (45 latents), bad (34 latents), and ugly (21 latents). This terminology for latent palmprint quality is adapted from NIST SD27 [35] where latent fingerprints were assigned the same labels. The average number of reliable minutiae extracted in good, bad, and ugly latents is 77, 56, and 45, respectively. An example image with each quality level is shown in Fig. 18. Fig. 19 shows the CMC curves for matching latents with these three quality levels against the background database of 10,200 full prints. As expected, the matching performance of latents with different quality levels is dramatically different. These results indicate that, while the proposed system can deal with latents of good quality satisfactorily, the intervention of latent experts is still necessary in the case of latents with bad and ugly quality.

The three latents (same as those in Fig. 18) and their mated full prints, which were all correctly identified at rank-1 by the proposed algorithm, are shown in Fig. 20. The latents have been aligned with the mated full prints. Two examples of unsuccessful match are shown in Fig. 21, where the poor quality of both the latents and full print is the reason for the matching failure. Unsuccessful matches can be classified into two categories: alignment failure and match failure. Alignment failures occur in the case of very low-quality prints (e.g., Fig. 21a) and match failures occur in the case of moderately low-quality prints (e.g., Fig. 21b). In all the 31 unsuccessful matches, 67.7 percent of them belong to alignment failure.

4.7 Fusion of Latent Palmprints

At crime scenes, multiple latent palmprints from the same palm can be frequently found. Based on image and nonimage information (such as the position of the latents),
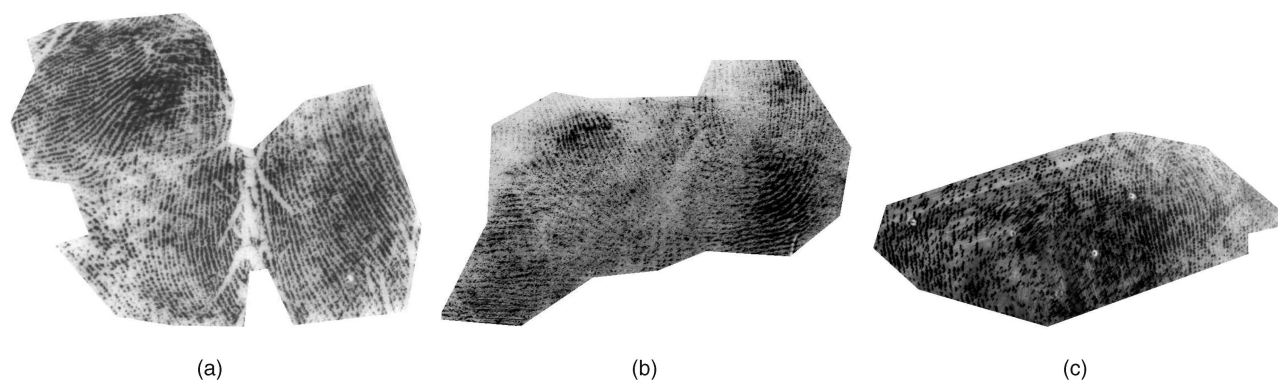
Fig. 18. Latents with three different quality levels. (a) Good (height: 552 pixels, width: 726 pixels), (b) bad (height: 511 pixels, width: 905 pixels), and (c) ugly (height: 473 pixels, width: 999 pixels).

latent experts can often reliably determine whether two latents are from the same palm. To determine the performance gain of fusing multiple latent palmprints, the following experiment was conducted. The 100 latents in our database were merged into 30 groups, each of which consists of multiple latents from the same palm. If any latent of a group leads to a successful match (identified at rank-1), this group is deemed as a successful match. This 'OR' rule is consistent with the practice in forensics. Based on this rule, the rank-1 recognition rate of searching 30 groups against the full background database is 90 percent, which is much higher than the rank-1 rate (69 percent) without fusion. All three (10 percent) latent groups that failed to identify contain only one latent that is of poor quality and cannot be improved by fusion.

4.8 Computational Requirements

The computational requirements of different modules of the proposed system on a PC with Intel 3 GHz CPU and Windows XP operating system are as follows. The average feature extraction time is 7 seconds for partial palmprints and 22 seconds for full palmprints. The DFT and Gabor filtering are the most computationally demanding parts of the feature extraction algorithm. The average matching time between a partial and a full palmprint is 0.34 seconds.
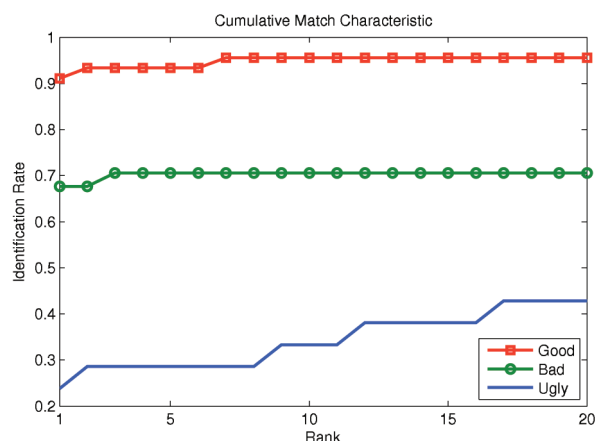

Fig. 19. CMC curves for matching latents with three different quality levels: good (45 latents), bad (34 latents), and ugly (21 latents) against the background database of 10,200 full prints.

Considering that a typical full palmprint has about 800 minutiae and a typical partial palmprint in our database has about 150 minutiae and no prealignment stage has been used prior to minutiae matching, this matching speed is reasonable. We have also tested the combined descriptor in [29] on a subset of palmprint images; its matching speed was found to be more than 10 times slower than MinutiaCode proposed here.

5 CONCLUSION AND FUTURE WORK

We have developed a prototype latent-to-full palmprint matching system. A region growing algorithm was developed to robustly estimate the local ridge direction and frequency even in the presence of overwhelming amount of noise. Our minutiae matching algorithm is based on a new fixed-length minutia descriptor which captures texture and neighboring minutiae information. The proposed system achieves rank-1 recognition rates of 78.7 and 69 percent, respectively, in searching 150 live-scan partial and 100 latent palmprints against a background database of 10,200 full palmprints. Partial palmprints from the thenar region are most difficult to match among the three palm regions. Quality of latents has a significant effect on the matching accuracy. Ancillary information in the form of hand type, palm region, and palm orientation can significantly improve both the matching accuracy and matching speed. A simple 'OR' rank level fusion of multiple latents from the same palm can improve the matching accuracy from 69 to 90 percent. Both the region growing algorithm and MinutiaCode can be applied for fingerprint matching as well. But, the current region growing algorithm may not be the best choice for the high curvature region that is very important region in fingerprints. MinutiaCode has been proposed to increase the matching efficiency and it is not as robust to distortion and disturbance of central minutiae as the combined descriptor in [29].

Designing a robust latent palmprint segmentation algorithm is our ongoing work. While the proposed region growing algorithm can recover a flat direction field even when the noise is overwhelming, it needs to be improved to deal with noisy high curvature areas of palmprints. For high-resolution (1,000 ppi) palmprints that are becoming available, we are exploring how to reliably extract and
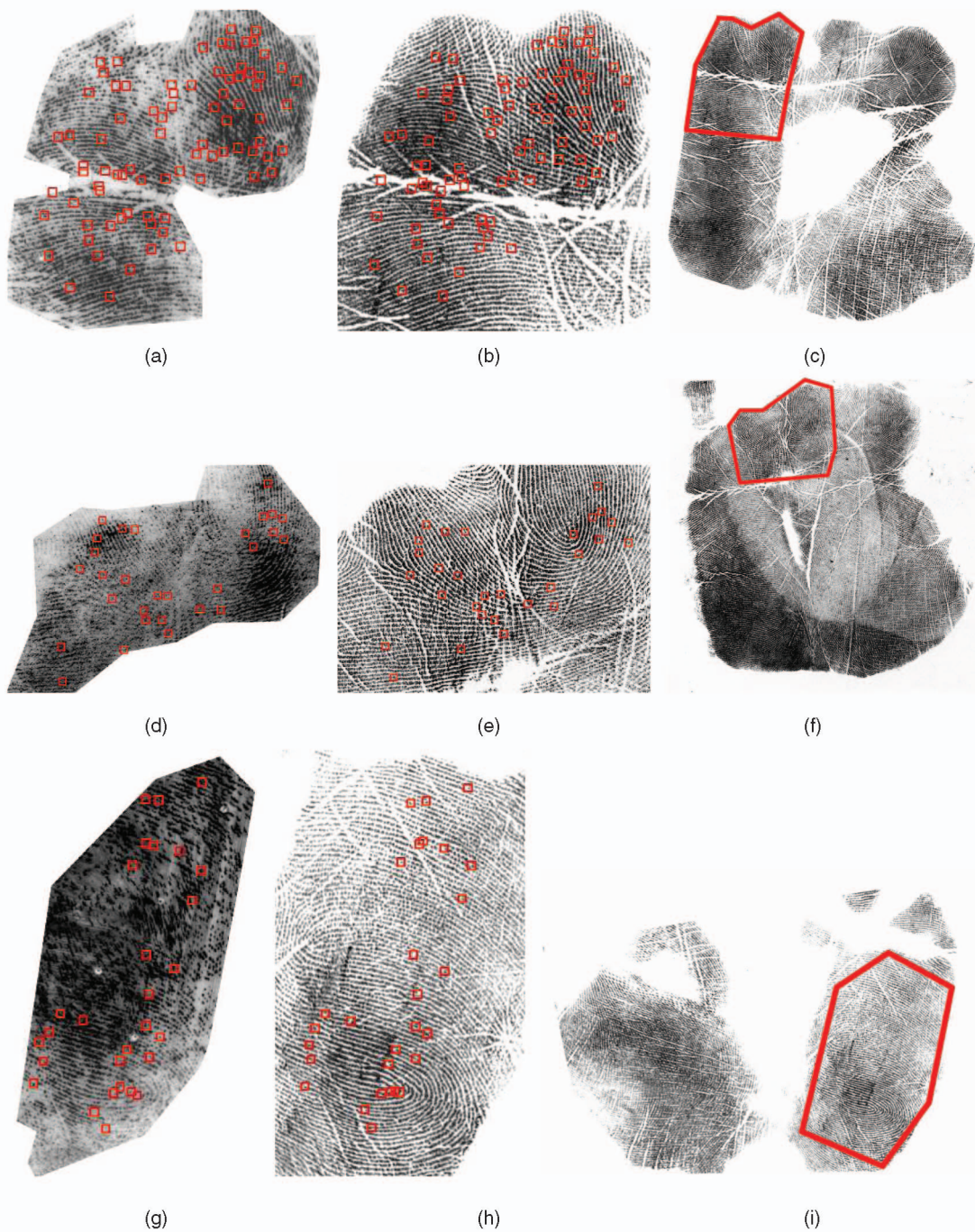
Fig. 20. Examples of successful latent match. (a) Latent of good quality, (b) corresponding region in the mated full print of (a), (c) the mated full print of (a), (d) latent of bad quality, (e) corresponding region in the mated full print of (d), (f) the mated full print of (d), (g) latent of ugly quality, (h) corresponding region in the mated full print of (g), and (i) the mated full print of (g). The matching minutiae are overlaid on the images. The corresponding regions are marked on the full prints.

utilize various types of extended features [36], especially creases, as palmprints often contain a large number of stable creases. Utilizing creases in latent palmprint matching is more likely to improve the matching accuracy for latents from the thenar region. For efficient search on a large background database (of the order of millions), our matching algorithm needs to be made more efficient. One approach is to use an indexing technique based on minutia triplets [37]. Another approach is to utilize the ancillary information, i.e., hand type, palm region, and palm orientation, since we have shown that such information can improve both the matching accuracy and matching speed. We plan to develop an algorithm to estimate the ancillary information from latent palmprints. Besides the interdigital, the thenar and the hypothenar regions, the writer's palm (the edge of palm opposite the thumb) is also frequently found at crime scenes. As the joint part of the palm and the back of the hand, the writer's palm generally contains very few minutiae. We plan to collect images of writer's palm and design a matcher which takes into account the characteristics of the writer's palm. It is generally believed that fusion at feature level can lead to
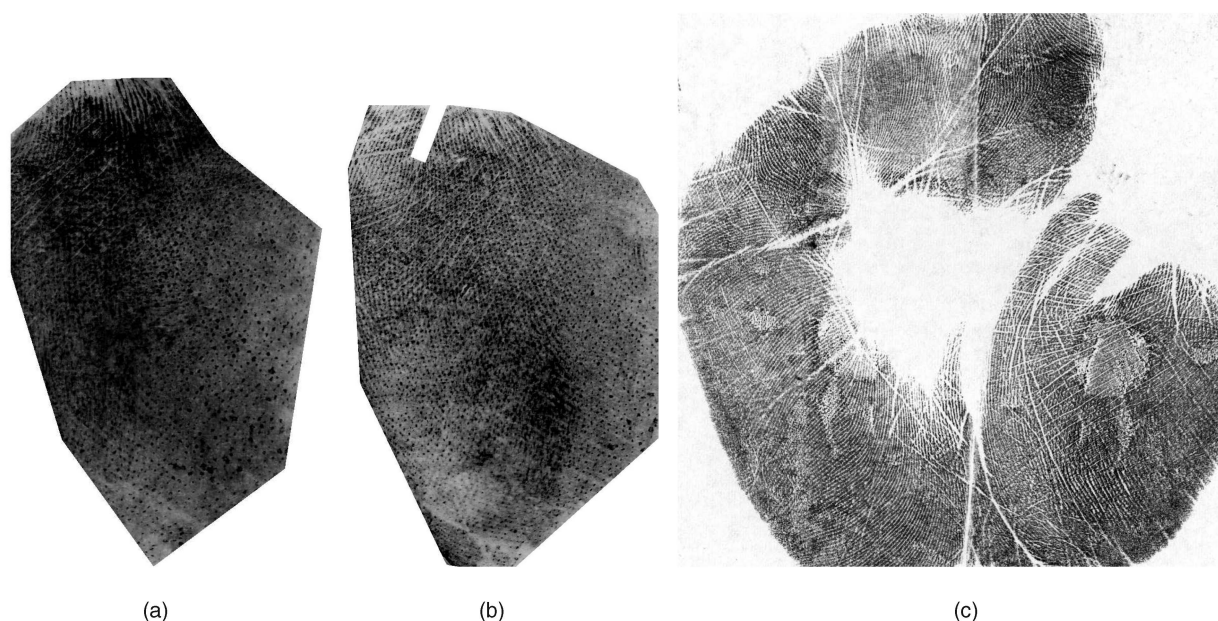
(a)   (b)   (c)

Fig. 21. Examples of unsuccessful latent match. In searching latents in (a) and (b) (from the thenar region of the same palm) against the background database of 10,200 full prints, the mated full print in (c) ranks 2,108 and 102, respectively. Correct alignment cannot be found for the latent in (a) due to unclear ridge structure. Although the correct alignment is found for the latent in (b), the matching score is low due to missing and spurious minutiae.

better accuracy than fusion at score/rank level. We plan to explore how to merge multiple fragmental latents from the same palm into a single latent palmprint with larger image size and better quality [38]. To automatically determine whether two latents are from the same palm, we need to develop a latent-to-latent palmprint matching algorithm.

ACKNOWLEDGMENTS

REFERENCES

[1] D.R. Ashbaugh, *Quantitative-Qualitative Friction Ridge Analysis: Introduction to Basic Ridgeology.* CRC Press, 1999.

[2] H. Cummins and M. Midlo, *Finger Prints, Palms and Soles: An Introduction to Dermatoglyphics.* Dover Publications, 1961.

[3] P. Komarinski, *Automated Fingerprint Identification Systems (AFIS).* Academic Press, 2004.

[4] F. Galton, *Fingerprints (reprint).* Da Capo Press, 1965.

[5] NSTC Subcommittee on Biometrics, "Palm Print Recognition," http://www.biometrics.gov/Documents/PalmPrintRec.pdf, 2009.

[6] S.K. Dewan, "Elementary, Watson: Scan a Palm, Find a Clue," *The New York Times,* http://www.nytimes.com/, Nov. 2003,

[7] The FBI's Next Generation Identification (NGI), http://fingerprint.nist.gov/standard/presentations/archives/NGI_Overview_Feb_2005.pdf, June 2008.

[8] D. Zhang, W.K. Kong, J. You, and M. Wong, "Online Palmprint Identification," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 9, pp. 1041-1050, Sept. 2003.

[9] Z. Sun, T. Tan, Y. Wang, and S.Z. Li, "Ordinal Palmprint Representation for Personal Identification," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. I: 279-284, 2005.

[10] R.K. Rowe, U. Uludag, M. Demirkus, S. Parthasaradhi, and A.K. Jain, "A Multispectral Whole-Hand Biometric Authentication System," *Proc. Biometric Symp. (BSYM), Biometric Consortium Conf.,* pp. 1-6, Sept. 2007.

[11] D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition.* Springer-Verlag, 2003.

[12] Neurotechnology Inc., VeriFinger, http://www.neurotechnology.com, 2009.

[13] FVC2006: the Fourth International Fingerprint Verification Competition, http://bias.csr.unibo.it/fvc2006/, 2009.

[14] C. Wilson et al., "Fingerprint Vendor Technology Evaluation 2003: Summary of Results and Analysis Report," NISTIR 7123, http://fpvte.nist.gov/report/ir_7123_analysis.pdf, June 2004,

[15] NIST Minutiae Interoperability Exchange Test (MINEX), http://fingerprint.nist.gov/minex/Results.html, 2009.

[16] A.K. Jain and M. Demirkus, "On Latent Palmprint Matching," technical report, Michigan State University, http://biometrics.cse.msu.edu/Publications/Palmprints/OnLatentPalmprintMatchingJainDemirkus08.pdf, 2008.

[17] D. Lowe, "Distinctive Image Features from Scale Invariant Keypoints," *Int'l J. Computer Vision,* vol. 20, pp. 91-110, 2003.

[18] J.G. Daugman, "Uncertainty Relation for Resolution in Space, Spatial Frequency, and Orientation Optimized by Two-Dimensional Visual Cortical Filters," *J. Optical Soc. Am. A,* vol. 2, no. 7, pp. 1160-1169, 1985.

[19] L. Hong, Y. Wan, and A.K. Jain, "Fingerprint Image Enhancement: Algorithm and Performance Evaluation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 8, pp. 777-789, Aug. 1998.

[20] J. Zhou and J. Gu, "A Model-Based Method for the Computation of Fingerprints' Orientation Field," *IEEE Trans. Image Processing,* vol. 13, no. 6, pp. 821-835, 2004.

[21] Y. Wang, J. Hu, and D. Phillips, "A Fingerprint Orientation Model Based on 2D Fourier Expansion (FOMFE) and Its Application to Singular-Point Detection and Fingerprint Indexing," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 4, pp. 573-585, Apr. 2007.

[22] J. Funada, N. Ohta, M. Mizoguchi, T. Temma, K. Nakanishi, A. Murai, T. Sugiuchi, T. Wakabayashi, and Y. Yamada, "Feature Extraction Method for Palmprint Considering Elimination of Creases," *Proc. 14th Int'l Conf. Pattern Recognition,* pp. 1849-1854, 1998.

[23] A.K. Jain, J. Feng, A. Nagar, and K. Nandakumar, "On Matching Latent Fingerprints," *Proc. CVPR Workshop Biometrics,* June 2008.

[24] A.M. Bazen, G.T.B. Verwaaijen, S.H. Gerez, L.P.J. Veelenturf, and B.J. van der Zwaag, "A Correlation-Based Fingerprint Verification System," *Proc. 11th Ann. Workshop Circuits Systems and Signal Processing,* pp. 205-213, Nov. 2000.

[25] M. Tico and P. Kuosmanen, "Fingerprint Matching Using an Orientation-based Minutia Descriptor," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 8, pp. 1009-1014, Aug. 2003.

[26] A.K. Jain, L. Hong, and R.M. Bolle, "On-Line Fingerprint Verification," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 4, pp. 302-314, Apr. 1997.

[27] X. Jiang and W.Y. Yau, "Fingerprint Minutiae Matching Based on the Local and Global Structures," *Proc. 15th Int'l Conf. Pattern Recognition,* pp. 1038-1041, 2000.

[28] X. Chen, J. Tian, and X. Yang, "A New Algorithm for Distorted Fingerprints Matching Based on Normalized Fuzzy Similarity Measure," *IEEE Trans. Image Processing,* vol. 15, no. 3, pp. 767-776, 2006.

[29] J. Feng, "Combining Minutiae Descriptors for Fingerprint Matching," *Pattern Recognition,* vol. 41, no. 1, pp. 342-352, 2008.

[30] Evaluation of Latent Fingerprint Technologies, http://fingerprint.nist.gov/latent/elft07/, 2007.

[31] Face Recognition Vendor Test, http://www.frvt.org/, 2009.

[32] Iris Challenge Evaluation, http://iris.nist.gov/ICE/, 2009.

[33] Noblis, http://www.noblis.org/, 2009.

[34] Ron Smith and Assoc., Inc., "Demystifying Palm Prints," http://www.ronsmithandassociates.com/, 2009.

[35] NIST Special Database 27, http://www.nist.gov/srd/nistsd27.htm, 2009.

[36] CDEFFS: the ANIS/NIST Committee to Define an Extended Fingerprint Feature Set, http://fingerprint.nist.gov/standard/cdeffs/index.html, 2009.

[37] R.S. Germain, A. Califano, and S. Colville, "Fingerprint Matching Using Transformation Parameter Clustering," *IEEE Computational Science and Eng.,* vol. 4, no. 4, pp. 42-49, 1997.

[38] A.K. Jain and A. Ross, "Fingerprint Mosaicking," *Proc. Int'l Conf. Acoustic Speech and Signal Processing,* vol. 4, pp. 4064-4067, May 2002.

Anil K. Jain is a university distinguished professor in the Department of Computer Science and Engineering at Michigan State University. His research interests include pattern recognition and biometric authentication. He received the 1996 *IEEE Transactions on Neural Networks* Outstanding Paper Award and the Pattern Recognition Society best paper awards in 1987, 1991, and 2005. He served as the editor-in-chief of the *IEEE Transactions on Pattern Analyis and Machine Intelligence* (1991-1994). He is a fellow of the AAAS, ACM, IEEE, IAPR, and SPIE and a member of the IEEE Computer Society. He has received Fulbright, Guggenheim, Alexander von Humboldt, IEEE Computer Society Technical Achievement, IEEE Wallace McDowell, and IAPR King-Sun Fu awards. The holder of six patents in the area of fingerprints, he is the author of a number of books, including *Handbook of Biometrics* (2007), *Handbook of Multibiometrics* (2006), *Handbook of Face Recognition* (2005), *Handbook of Fingerprint Recognition* (2003), *BIOMETRICS: Personal Identification in Networked Society* (1999), and *Algorithms for Clustering Data* (1988). ISI has designated him a highly cited researcher. According to Citeseer, his book *Algorithms for Clustering Data* (Prentice-Hall, 1988) is ranked #93 in most cited articles in computer science. He is currently serving as an associate editor of the *IEEE Transactions on Information Forensics and Security* and the *ACM Transactions on Knowledge Discovery in Data.* He is a member of the Defense Science Board and The National Academies committees on Whither Biometrics and Improvised Explosive Devices.
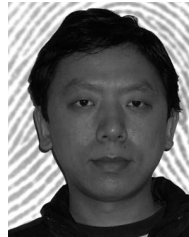
Jianjiang Feng received the BS and the PhD degrees from the School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, China, in 2000 and 2007, respectively. From 2006 to 2007, he was a research associate in the Biometric Research Center at the Hong Kong Polytechnic University. Now he is working as a postdoctoral researcher in the Pattern Recognition & Image Processing Laboratory at Michigan State University. His research interests include fingerprint recognition, palmprint recognition, and structural matching.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.