A Coarse to Fine Minutiae-Based Latent Palmprint Matching

Eryun Liu, Anil K. Jain, *Fellow*, *IEEE*, and Jie Tian, *Fellow*, *IEEE*

Abstract—With the availability of live-scan palmprint technology, high resolution palmprint recognition has started to receive significant attention in forensics and law enforcement. In forensic applications, latent palmprints provide critical evidence as it is estimated that about 30 percent of the latents recovered at crime scenes are those of palms. Most of the available high-resolution palmprint matching algorithms essentially follow the minutiae-based fingerprint matching strategy. Considering the large number of minutiae (about 1,000 minutiae in a full palmprint compared to about 100 minutiae in a rolled fingerprint) and large area of foreground region in full palmprints, novel strategies need to be developed for efficient and robust latent palmprint matching. In this paper, a coarse to fine matching strategy based on minutiae clustering and minutiae match propagation is designed specifically for palmprint matching. To deal with the large number of minutiae, a local feature-based minutiae clustering algorithm is designed to cluster minutiae into several groups such that minutiae belonging to the same group have similar local characteristics. The coarse matching is then performed within each cluster to establish initial minutiae correspondences between two palmprints. Starting with each initial correspondence, a minutiae match propagation algorithm searches for mated minutiae in the full palmprint. The proposed palmprint matching algorithm has been evaluated on a latent-to-full palmprint database consisting of 446 latents and 12,489 background full prints. The matching results show a rank-1 identification accuracy of 79.4 percent, which is significantly higher than the 60.8 percent identification accuracy of a state-of-the-art latent palmprint matching algorithm on the same latent database. The average computation time of our algorithm for a single latent-to-full match is about 141 ms for genuine match and 50 ms for impostor match, on a Windows XP desktop system with 2.2-GHz CPU and 1.00-GB RAM. The computation time of our algorithm is an order of magnitude faster than a previously published state-of-the-art-algorithm.

Index Terms—Palmprint, latent palmprint matching, minutiae clustering, minutia descriptor, match propagation

1 Introduction

Latent palmprints contain critical evidential value in forensic applications because it is estimated that about 30 percent of the latents lifted at crime scenes are those of palmprints [1]. Due to the high cost of live-scan technology for palmprints, most of the existing automatic palmprint matching systems only use low resolution (about 100 ppi) palmprint images (e.g., [2], [3]), and target civilian access control applications. Research on designing automated high-resolution (at least 500 ppi) palmprint matching systems has emerged only in recent years [4], [5], [6], [7], [8]. The features used for high-resolution palmprint matching mainly consist of minutiae points, which have been widely used in fingerprint matching [9], [10], [11], [12]. However, algorithms developed for fingerprint minutiae matching are not appropriate for matching the minutiae in palmprints, due to the following three reasons: 1) the size of palmprint images are much larger than that of fingerprint images, which leads to computationally demanding feature extraction stage; 2) palmprints have a large number of creases (i.e., immutable creases like radial transverse creases, proximal transverse creases and distal transverse creases, and mutable creases [13]) that can lead to many false minutiae (See Fig. 1); and 3) a larger nonlinear distortion exists in palmprint images than in fingerprint images because the skin and bone structure of palms are very flexible.

Based on the source and size of palmprint images, palmprint matching can be divided into three categories: 1) full-to-full palmprint matching; 2) latent-to-full palmprint matching; and 3) live-scan partial-to-full palmprint matching. Due to extremely high level of accuracy of state-of-the-art 10-print fingerprint matchers and lack of a large legacy database of palmprints, full-to-full palmprint matching is generally not of interest in law enforcement and forensics. Accurate and efficient latent-to-full palmprint matching algorithms, on the other hand, are urgently needed in forensic applications because latent palmprints are frequently encountered at crime scenes. Latent-to-full palmprint matching is a challenging problem because of the following reasons:

1. uncontrollable latent image quality;
2. small area of latent palmprints compared to full palmprints;
3. large nonlinear image distortion; and
4. a large number of spurious (noisy) minutiae.

- *E. Liu is with the Department of Computer Science and Engineering, Michigan State University (MSU), 428 S. Shaw Ln., RM 3115, East Lansing, MI 48824, and the School of Life Sciences and Technology, Xidian University, Xi'an, Shaanxi 710126, China. E-mail: liueryun@cse.msu.edu.*
- *A.K. Jain is with the Department of Computer Science and Engineering, Michigan State University, 428 S. Shaw Ln., RM 3115, Engineering Building, East Lansing, MI 48824, and the Department of Brain and Cognitive Engineering, Korea University, Seoul. E-mail: jain@cse.msu.edu.*
- *J. Tian is with the School of Life Sciences and Technology, Xidian University, Xi'an, Shaanxi 710126, China, and the Institute of Automation, Chinese Academy of Sciences, 918 Automation Building, 95 East Zhong Guan Cun Rd., Beijing 100190, China. E-mail: tian@ieee.org.*

Manuscript received 12 July 2012; revised 7 Nov. 2012; accepted 1 Feb. 2013; published online 13 Feb. 2013.
Recommended for acceptance by M. Tistarelli.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2012-07-0528.
Digital Object Identifier no. 10.1109/TPAMI.2013.39.

Published by the IEEE Computer Society

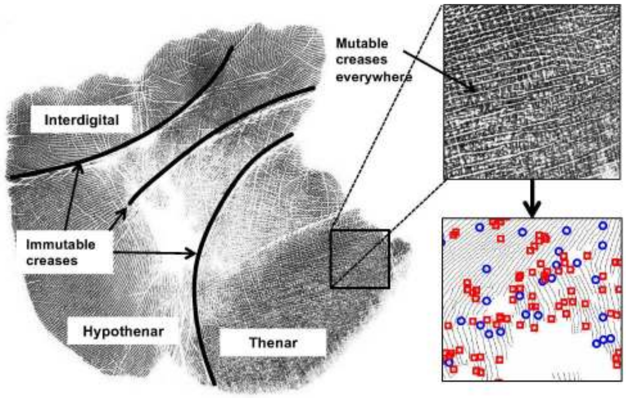


Fig. 1. Immutable creases divide a palmprint into three regions (interdigital, hypothenar, and thenar). Mutable creases break ridge structures and lead to a large number of spurious minutiae (the minutiae shown are extracted by the algorithm in [5], red squares: Unreliable minutiae, blue circles: Reliable minutiae).

Given that no public domain latent-to-full palmprint database is currently available, several studies have resorted to using cropped live-scan prints instead of latent prints to simulate the forensic scenarios. However, the cropped live-scan partial palmprint images have very different characteristics compared to latent palmprints in terms of image quality, distortion and background, making it much easier to match cropped live-scan palmprints compared to latent palmprints.

1.1 Palmprint Matching Algorithms

Table 1 summarizes some of the available high-resolution palmprint matching algorithms as well as the proposed algorithm. Jain and Demirkus [14] proposed a partial-to-full palmprint matching system based on a fusion of SIFT keypoints [15] and minutiae points. Jain and Feng [5] developed a latent-to-full palmprint matching system. They proposed a region growing algorithm to deal with creases

TABLE 1
A Comparison of High-Resolution Palmprint Matching Algorithms Proposed in the Literature

Algorithm	Features used	Alignment	Distortion	Matching strategy	Database	Accuracy & efficiency
Jain and Demirkus [14]	minutiae points and SIFT points [15]	Heart line used for full palmprint alignment	Commercial matcher [16] for sector-wise matching	Sector matching; five sectors detected	Partial-to-full[a] (partials: 500; background: 100); full-to-full (100 palms, with 10 impressions per palm)	Partial-to-full: rank-1 identification rate: 96%; full-to-full: EER≈1%; efficiency: extremly slow
Jain and Feng [5]	Minutiae points and orientation field	Rigid alignment based on most similar minutiae pairs	No explicit consideration to distortion	Global matching	Latent-to-full[a] (latents: 100; background: 10,200); partial-to-full (partials: 150; background: 10,200)	Latent-to-full: rank-1 identification rate: 69% partial-to-full: rank-1 identification rate: 78.7%; efficiency: slow
Dai and Zhou [6]	Minutiae points, ridge density map, principal map and orientation field	Rigid alignment using Hough transform	No explicit consideration to distortion	Global matching	Partial-to-full (partials: 480; background: 13,696); full-to-full (80 palms, with 8 impressions per palm)	Partial-to-full: rank-1 identification rate:91.7%; full-to-full: EER = 4.8%; efficiency: very slow
Dai et al. [7]	Minutiae points, orientation field and ridge density map	Average orientation field for coarse full palmprint alignment and generalized Hough transform for fine segment level alignment; manual alignment for partial (latent) palmprints	Segment level robustness to distortion	Segment matching	Partial-to-full (partials: 2,520; background: 13,736); full-to-full (160 palms, with 8 impressions per palm)	Partial-to-full: TAR=91.9% @FAR=3×10^{-8}; full-to-full: TAR = 97.9% @FAR=2×10^{-7}; efficiency: fast
Cappelli et al. [17]	Minutiae Cylinder Code (MCC)	No alignment needed	No explicit consideration to distortion	Relaxation	Full-to-full (120 palms, with 8 impressions per palm)	Full-to-full: EER $< 0.01\%$; efficiency: fast
Proposed	Minutiae points, orientation descriptor and ridge period descriptor	Local alignment propagation from initial minutiae pairs	Robust to distortion	Local and global matching	Latent-to-full (latents: 446; background: 12,489); full-to-full (160 palms, with 8 impressions per palm)	Latent-to-full: rank-1 identification rate: 79.4%; full-to-full: EER= 0.11%; efficiency: very fast

[a] In this paper, partial palmprint images always denote the cropped live-scan palmprint images used for simulation of latent palmprint images; latent palmprint images denote the images lifted from crime scenes.

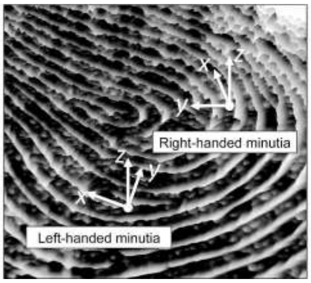


Fig. 2. Three-dimensional Cartesian coordinate system indicating the minutiae handedness property; x-axis: Minutiae direction (tangent vector to ridge); y-axis: Ridge bending direction (curvature vector to ridge); z-axis: Direction pointing out of paper (normal vector to tangent plane of palm surface).

and introduced a novel minutiae descriptor, called the MinutiaCode, to measure the minutiae similarity. A rigid transformation based on the top-5 most similar minutiae pairs was used to establish the minutiae correspondences. In [6], the authors proposed a multifeature palmprint matching system, in which minutiae, ridge density map, principal line map, and orientation field were fused together. These published algorithms basically follow the fingerprint matching strategy, and hence, they suffer from high-computational complexity. To account for the large number of minutiae and nonlinear image distortion, Dai et al. [7] proposed a segment-based palmprint matching system. However, this algorithm is essentially designed for full-to-full palmprint matching. It requires a manual alignment for latent-to-full palmprint matching, which is not feasible in practice.

1.2 Motivation and Contributions of This Paper

Pairwise minutiae similarity computation between query and template palmprint minutiae sets is a computationally demanding step in palmprint matching. Since many different minutiae can be characterized in terms of local properties (e.g., local ridge flow), this property can be used to quickly filter out many false minutiae correspondences. "Handedness" is one of the local ridge flow properties. The term "handedness" here is used to describe the relationship between minutiae direction and ridge bending direction. Let x-axis be the direction of the central minutia (tangent vector to ridge), y-axis be perpendicular to the central minutia direction and point toward the inner curve of local ridge flow (curvature vector to ridge), and let z-axis point outside the paper (normal vector to the tangent plane of palm surface at the minutia). If this is a right-handed Cartesian coordinate system, then we call the central minutia a right-handed minutia, otherwise a left-handed minutia (see Fig. 2). If ridge flow in the local region of central minutiae is not bending in the same direction, then the central minutia is a nonhanded minutia. Our definition of "handedness" is different from the definition in [18]. In [18], "handedness" is defined as a global feature of minutiae in a fingerprint, where singular points are needed to compute this property. This dependency on singular points makes it difficult to apply the definition in [18] to minutiae in palmprints, where it is not easy to identify singular points that can be used as reference points. Fig. 2 shows two minutiae with different handedness property defined in this paper.

For a genuine minutiae pair, its neighboring minutiae should also match with high confidence. This heuristic makes it possible to propagate, among all the minutiae correspondences, from one minutiae pair to another minutiae pair in a genuine pair of palmprints. On the other hand, in an impostor pair of palmprints, corresponding minutiae may appear at arbitrary locations in the palmprint image. So, it is not possible to propagate minutiae correspondences in an impostor pair of palmprints.

The above two observations motivated us to design and develop a fast and accurate coarse to fine latent palmprint matching system. Fig. 3 outlines the proposed system which specifically deals with the challenges associated with large number of minutiae and nonlinear image distortion in palmprints. A minutiae clustering algorithm is first applied to group minutiae points into K_c clusters based on local minutia features. Then, a small number of initial minutiae correspondences is selected within each cluster. Starting with each initial minutiae correspondence, a minutiae match propagation algorithm is invoked to establish global one-to-one minutiae correspondences. The final match score is computed based on individual match propagation results. The contributions of this paper are as follows:

1. A minutiae clustering algorithm is proposed to significantly reduce latent-to-full palmprint matching complexity by avoiding the similarity computation between minutiae with different local characteristics, which are deemed not to be matched.
2. A minutiae match propagation algorithm is proposed to obtain the final feature correspondences between palmprints. The propagation is able to identify most of the true correspondences and quickly reject impostor matchings.
3. Our algorithm has been evaluated on a large real latent palmprint database (a total number of 446 latent images are matched to a background database of 12,489 full palmprint images). Both the complexity and the accuracy of the proposed algorithm are significantly better than state-of-the-art algorithms.
4. A simplified version of the proposed latent-to-full palmprint matching algorithm is developed for full-to-full palmprint matching. This proposed algorithm achieves state-of-the-art performance in terms of verification error rate and computation time on a public domain full palmprint database, THUPALMLAB [7].

The rest of the paper is organized as follows: Section 2 presents a palmprint minutiae clustering algorithm based on minutiae descriptors by using the well known K-means clustering algorithm. Section 3 presents a minutiae match propagation scheme to establish minutiae correspondences on the whole palmprint image from a set of initial minutiae pairs. The final palmprint match score computation is described in Section 4. Experimental results are reported in Section 5. Finally, conclusions and directions for future research are reported in Section 6.

2 Palmprint Minutiae Clustering

Local ridge orientation and ridge period features are used for characterizing the local features of minutiae. The

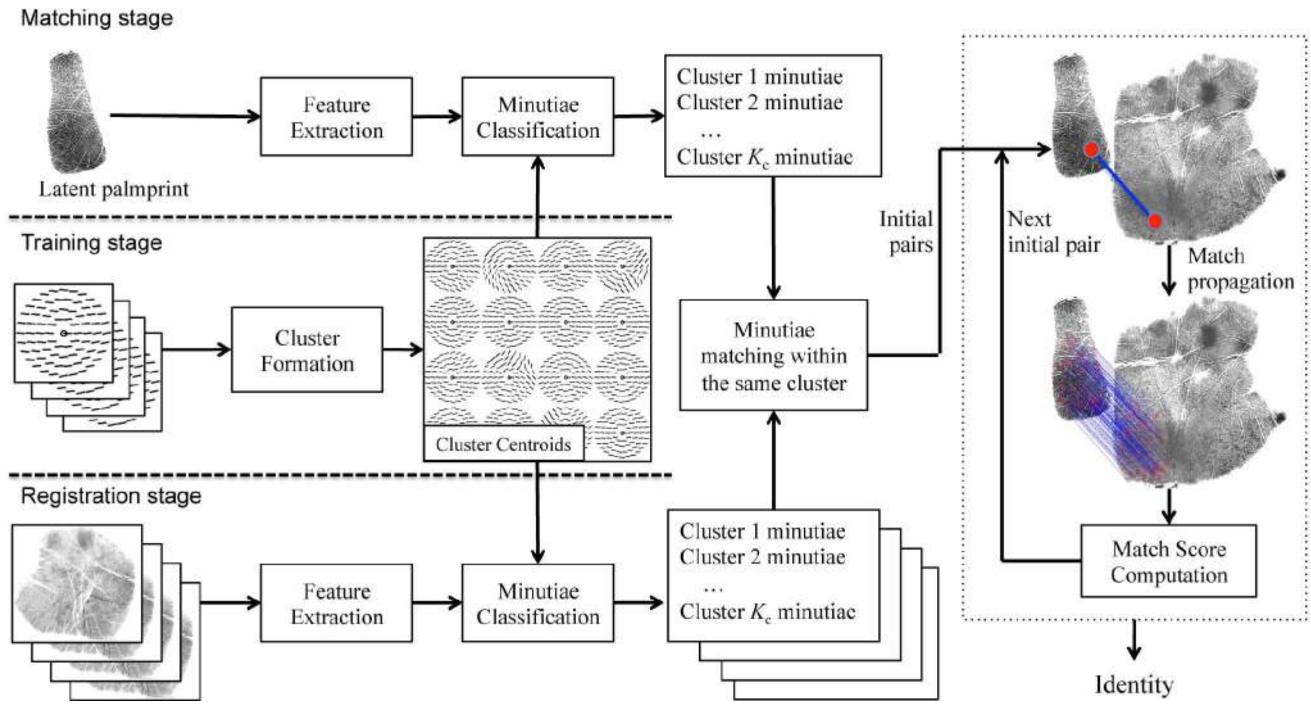


Fig. 3. Flowchart of the proposed latent palmprint matching system. The system consists of three stages: (a) training stage: A K-means clustering algorithm is applied to a set of training descriptors to get a set of cluster centroids; (b) registration stage: For a given template of full palmprint image, minutiae and descriptors are extracted and clustered based on the obtained centroids; (c) matching stage: Given a query latent palmprint image, minutiae and descriptors are first extracted and clustered as the registration stage, then a small number of initial minutiae pairs is selected within the same cluster and fed to a match propagation procedure to get a fine match result.

well-known K-means algorithm [19], [20] is used for clustering minutiae.

2.1 Clustering Minutiae

Tico and Kuasmanen [21] proposed a minutiae local orientation sampling structure to extract local features for minutiae, which has been shown to be useful in fingerprint matching [22], [23]. We adopt this sampling structure to extract local ridge orientation and ridge period features for minutiae in palmprint images. Given a reference minutia m, L circles with radii $r_i, i = 1, \dots, L$, centered at the minutia are considered as its local neighborhood. On each circle, there are $n_i, i = 1, \dots, L$, sample points equally distributed, starting from the projection location of the reference minutia along its direction on the circle. The feature value of orientation descriptor at each sample point is the difference between the local ridge orientation at the sample point and the ridge orientation at the location of reference minutia. The feature value of ridge period descriptor at a sample point is simply set as the ridge period value at that point. The orientation descriptor and ridge period descriptor of minutia m are denoted as $\mathrm{OriDes} = \{\dots, (o_{i,j} - o_m), \dots\}$ and $\mathrm{RidDes} = \{\dots, w_{i,j}, \dots\}$, where $i = 1, \dots, L, j = 1, \dots, n_i$, o_m is the ridge orientation at the center minutia m, and $o_{i,j}$ and $w_{i,j}$ are local ridge orientation and ridge period at the jth sample point on the ith circle, respectively. In this paper, the sampling structure consists of four circles ($L = 4$), whose radii are 27, 45, 63, and 81 pixels, and on which 10, 16, 22, and 28 points are sampled, respectively. The total number of sample points on the four circles is 76. The descriptor vectors denoted as $\mathrm{OriDes} = \{o_i\}_{i=1}^{76}$ and $\mathrm{RidDes} = \{w_i\}_{i=1}^{76}$ are invariant to palmprint translation and rotation.

For the minutiae near the boundary of a palmprint, some of the sample points on the circles may fall outside the foreground region. For these sample points, which we call invalid points, the corresponding feature values cannot be computed directly because there is no ridge orientation or ridge period information available at their locations. To facilitate the use of clustering algorithm for those descriptors containing invalid values, we predict the values at invalid points from their n_v nearest valid (in the foreground region of palmprint) sample points by

$$o = \frac{1}{2}\tan^{-1}\left(\frac{\sum_{i\in V}\sin 2o_i}{\sum_{i\in V}\cos 2o_i}\right), \tag{1}$$

$$w = \frac{1}{n_v}\sum_{i\in V} w_i, \tag{2}$$

where $\tan^{-1}$ is a four-quadrant arctangent function, V is the set of valid nearest sample points with n_v points in the structure, and o and w are the predicted ridge orientation and ridge period values, respectively. Fig. 4 shows two example descriptors with predicted values.

The distance between two orientation descriptors cannot be directly measured in the euclidean space because of the periodic singularity of angle after wrapping. The distance between two orientation descriptors, $\mathrm{OriDes} = \{o_i\}_i^{76}$ and $\mathrm{OriDes}' = \{o_i'\}_i^{76}$, is defined as

$$\mathrm{Dis} = \sqrt{\sum_{i=1}^{76}\left[\left(\cos 2o_i - \cos 2o_i'\right)^2 + \left(\sin 2o_i - \sin 2o_i'\right)^2\right]}. \tag{3}$$

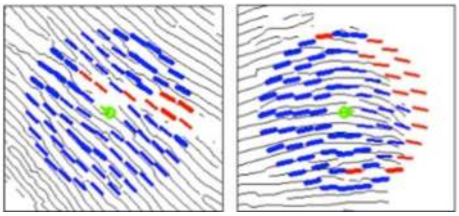


Fig. 4. Examples of descriptors with predicted values; red: Predicted values, blue: Valid values, green: Reference minutiae. Thicker lines indicate larger ridge periods.

Given a set of orientation descriptors belonging to the same cluster, $\mathrm{OriDes}_k = \{o_{k,i}\}$, where $k = 1, \ldots, S$, S is the number of descriptors that belong to the same cluster, and $i = 1, \ldots, 76$, the cluster centroid, $\mathrm{OriCen} = \{c_i\}_{i=1}^{76}$, is computed by

$$c_i = \frac{1}{2}\tan^{-1}\left(\frac{\sum_{k=1}^{S}\sin 2o_{k,i}}{\sum_{k=1}^{S}\cos 2o_{k,i}}\right). \qquad (4)$$

The clustering for ridge orientation descriptors is done by using the K-means algorithm with the distance measure defined in (3) and cluster centroids computed by (4). The clustering for ridge period descriptors is done by using the K-means algorithm with the euclidean distance measure.

2.2 Centroids of Ridge Orientation Descriptors

Fig. 5 shows 16 ($K_c = 16$) ridge orientation descriptor centroids obtained from clustering a total of 642,749 descriptors from a set of 800 full palmprints by K-means algorithm [19], using (3) as the distance measure and (4) for computing the cluster centers. More discussion on the choice of K_c (i.e., the number of clusters) is provided in Section 5. Basically, each orientation descriptor centroid represents a type of ridge flow pattern, such as the left handedness, right handedness, singularity, and so on. These centroids can be regarded as atomic patterns (or prototype patterns) of local ridge flow surrounding a minutia. Six basic types of local ridge flow patterns are revealed from these cluster centroids. They are

1. left-handedness (clusters 1, 3, 7, 8, and 11),
2. right-handedness (clusters 5, 12, 14, 15, and 16),
3. radiation (clusters 9 and 13),
4. contraction (cluster 6),
5. with-core point (cluster 2), and
6. with-delta point (clusters 4 and 10).

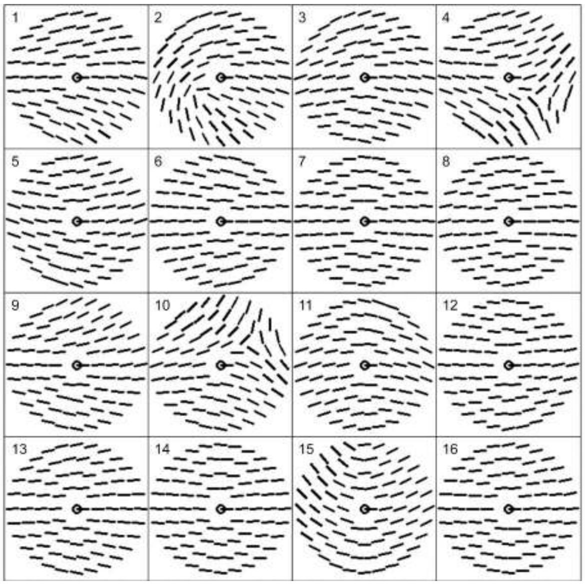


Fig. 5. Orientation descriptor centroids obtained by clustering the descriptors from minutiae in 800 full palmprints with distance measure defined in (3). The total number of descriptors (extracted from 800 full palmprints) that were clustered is 642,749.

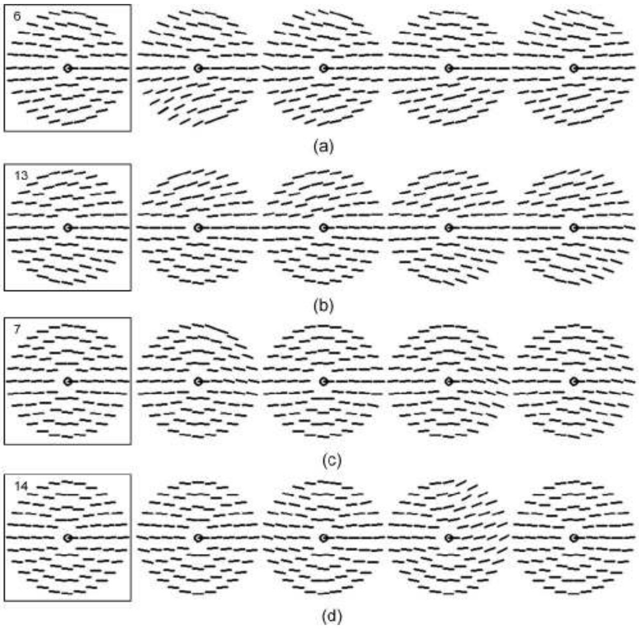


Fig. 6. Examples of minutiae clustering; the orientation descriptors in the first column (in squares) represent the centroids of four different clusters. Columns 2 to 5 show four examples of minutiae orientation descriptors that belong to the corresponding clusters in the first column; (a) a contraction type of cluster; (b) a radiation type of cluster; (c) a left-handedness type of cluster; and (d) a right-handedness type of cluster.

By setting the value of K_c to a larger value, more complex ridge flow patterns can be revealed. Given an orientation descriptor of a minutia (i.e., minutia in latent palmprint or full palmprint), the minutia can be associated to the cluster whose centroid has the smallest distance to its descriptor. Fig. 6 shows some examples of minutiae clustering results. We can see that while the cluster centroids are smoother than individual orientation descriptors of minutiae, the individual orientation descriptors have similar ridge flow patterns as their corresponding cluster centroids.

2.3 Centroids of Ridge Period Descriptors

The K-means clustering algorithm [19] is applied to the ridge period descriptors of the same minutiae set used in training the orientation descriptor cluster centroids. Fig. 7 shows 16 (i.e., $K_c = 16$) centroids of ridge period descriptors, with the width of lines proportional to ridge period at the corresponding locations of sample points. These centroids can be roughly divided into two types. The first type of centroids has a very small variation in the ridge period, such as centroids 1, 2, 3, 8, 11, and 14. Minutiae belonging to these clusters usually appear in flat

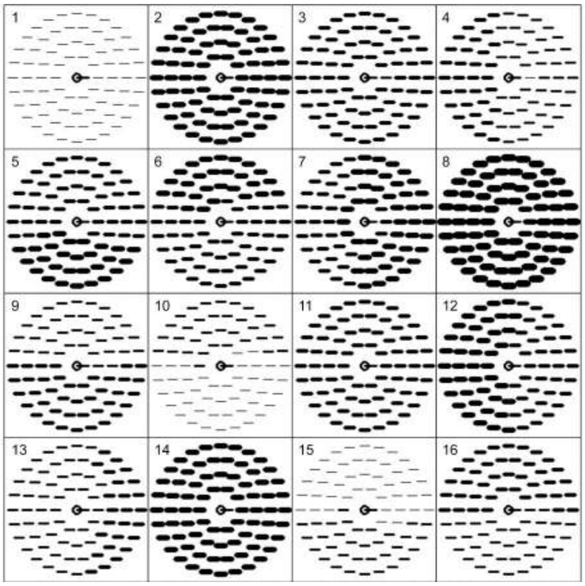


Fig. 7. Ridge period descriptor centroids obtained from 642,749 descriptors of 800 full palmprints. The thickness of the line is proportional to the ridge period value.

regions of palmprints, where the ridge period is almost constant. The remaining centroids have relatively larger variations in ridge period, and the ridge period values change smoothly from one side to the other side of the center minutiae.

3 Minutiae Match Propagation

In [24], the authors proposed a match propagation algorithm, which produced a quasi-dense pixel matching between two images, which was later extended in [25]. This match-growing strategy has wide applications in, e.g., stereo image matching [26], image registration [27], and image rendering [28], [29]. In this paper, we use the match propagation strategy in an attempt to simulate how forensic experts match two minutiae sets [13], [30]. The matching procedure adopted by forensic examiners [13] usually starts with an initial pair of minutiae (or a pair of singular points) in two palmprints to be matched. If this minutiae pair is a true correspondence, then their neighboring minutiae should also match (usually by following the ridge line and counting the minutiae near this ridge); by recording these mated neighboring minutiae, additional neighboring minutiae are matched, recursively, until no more neighboring minutiae can be matched.

The difference between the match propagation of the proposed method and K-plet [31] is that we use a fixed radius-based local minutiae structure descriptor, which makes it possible to tessellate it into sectors and accelerate the match propagation. On the other hand, K-plet is a fixed number of minutiae-based local structure and the matching of two K-plets is based on the string alignment algorithm. The match propagation based on the proposed local structure descriptor is more suitable for palmprint matching because of the large number of minutiae in palmprints.

The proposed minutiae match propagation algorithm consists of three basic steps: 1) minutiae similarity measure; 2) initial mated minutiae pairs selection; and 3) minutiae match propagation.

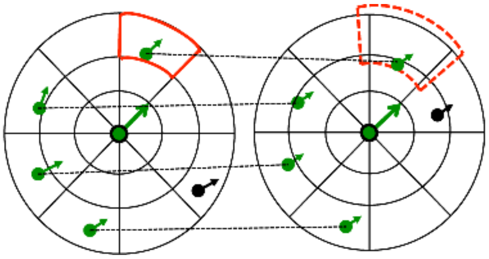


Fig. 8. Matching of local minutiae structures. Left: Query local minutiae structure; Right: Template local minutiae structure.

3.1 Minutiae Similarity Measure

Minutiae similarity measure plays a vital role in minutiae matching. Three types of minutiae similarities are used in this paper. The first two are based on ridge orientation and ridge period descriptors, which are also used for minutiae clustering, and the third one is based on local minutiae structure.

Given a query minutia (i.e., minutia in the latent image) with ridge orientation descriptor $\mathrm{OriDes} = \{o_i\}_{i=1}^{76}$ and ridge period descriptor $\mathrm{RidDes} = \{w_i\}_{i=1}^{76}$, and a template minutia (i.e., minutia in the full palmprint image) with ridge orientation descriptor $\mathrm{OriDes}' = \{o'_i\}_{i=1}^{76}$ and ridge period descriptor $\mathrm{RidDes}' = \{w'_i\}_{i=1}^{76}$, the similarities among orientation and ridge period descriptors are computed, respectively, by

$$S_o = \frac{1}{N_v} \sum_{i \in V} \exp\left(-\frac{\Delta o_i}{\mu_o} \right), \tag{5}$$

$$S_r = \frac{1}{N_v} \sum_{i \in V} \exp\left(-\frac{\Delta w_i}{\mu_r} \right), \tag{6}$$

where V is the set of sample points valid in both descriptors, N_v is the number of valid sample points, $\Delta o_i = \min\{|o_i - o'_i|, 180 - |o_i - o'_i|\}$, $\Delta w_i = |w_i - w'_i|$, and μ_o and μ_r are two predefined parameters. (In this paper, they are set to 4 and 2, respectively). When $N_v < 20$, S_o and S_r are both set to zero. Although the values of invalid sample points are predicted in the minutiae clustering stage, they are not used for the similarity computation here, because the predicted values are not as stable as the valid ones.

The third minutiae similarity measure is based on the local minutiae structure. To ensure a fast and accurate matching, the local region around a minutia is tessellated into several sectors according to the local polar coordinate system with the location and direction of center minutia as pole and polar axis, respectively. The tessellation here is slightly different from [5]. For a query minutia (i.e., in a latent palmprint), the circular region around a central minutia is divided into $R \times K$ sectors using R concentric circles and K radial lines as illustrated by Fig. 8. The lines are equally spaced starting from the direction of the central minutia. The radius of the ith circle (ordered from inner to outer) is $i \times \frac{r_{max}}{R}$, where $i = 1, \ldots, R$ and r_{max} is the radius of the outer most circle. In this paper, R and K are set to 4 and 8, respectively, and r_{max} is set to 180 pixels. For minutiae in template (i.e., full palmprint), the tessellation is different. While the center of each sector is not changed, each sector now has 10 degrees of overlapping angle and 20 pixels of overlapping radius with its neighboring sectors. The overlapped tessellation for template minutiae is to account for distortion in palmprints and displacement of extracted minutiae.

The local minutiae structure is denoted by $\mathrm{LmsDes} = \{S_i\}_{i=1}^{N_b}$, where $N_b = R \times K$ is the number of sectors, $S_i = \{(r_{j,i}, t_{j,i}, \alpha_{j,i})\}_{j=1}^{k_i}$ is the set of local minutiae in the ith sector represented by their local polar coordinates with respect to the center minutiae, k_i is the number of minutiae in the ith sector, r and t are the radius and angle coordinate values in the local polar coordinate system, and α is the normalized minutia direction with respect to the center minutiae direction. All features of local minutiae are represented with respect to a central minutiae, so that the local structures are invariant under translation and rotation.

Once all the sectors are aligned with respect to the reference central minutiae, the matching of local minutiae only needs to be performed within corresponding sectors. That is to say, given a local minutia (r, t, α) in the ith sector of a query structure, it is only necessary to find its mated minutia in the ith sector of a template structure so as to speed up the matching of two local structures.

Two local minutiae, (r, t, α) and (r', t', α'), are considered as being mated if they satisfy the following conditions:

$$\begin{cases} \Delta t < Th_t, \\ \Delta r < \dfrac{r}{r_{max}} \times Th_r, \\ \Delta \alpha < Th_\alpha, \end{cases} \tag{7}$$

where $\Delta t = \min\{|t - t'|, 360 - |t - t'|\}$, $\Delta r = |r - r'|$, $\Delta \alpha = \{|\alpha - \alpha'|, 360 - |\alpha - \alpha'|\}$, Th_α, Th_r, and Th_t are three predefined thresholds. In this paper, the thresholds are set to 8, 20, and 15, respectively. If more than one local minutiae is found in the same sector of template structure that satisfy the above conditions, then only the nearest minutia in terms of euclidean distance will be considered. The similarity of two mated local structures is computed by

$$S_m = \frac{1}{M}\sum_{i=1}^{M} s_i \times \frac{M}{N_t + N_q} \times \frac{M}{M + M_0} \times Q(d_q) \times Q(d_t), \tag{8}$$

where M is the number of matched local minutiae, M_0 is a predefined parameter (in this paper, M_0 is set to 10), N_q and N_t are the numbers of local minutiae in the query and template structures, respectively, and s_i is the similarity of the ith local minutiae pair and is computed by

$$s_i = f_{a_t,b_t}(\Delta t_i) \times f_{a_r,b_r}(\Delta r_i) \times f_{a_\alpha,b_\alpha}(\Delta \alpha_i),$$

with $f_{a,b}$ function defined as

$$f_{a,b}(x) = \begin{cases} 1, & \text{if } x < a, \\ 0, & \text{if } x > b, \\ \dfrac{b - x}{b - a} & \text{otherwise.} \end{cases}$$

$Q(x) = 1 - f_{a_Q,b_Q}(x)$ is a minutiae quality measure function defined in [32], and d_q and d_t are the average euclidean distances of three nearest minutiae to the reference minutia in query and template structures, respectively.

3.2 Initial Mated Minutiae Pair Generation

The initial pairs of mated minutiae serve as seeds for searching minutiae correspondences in the palmprints. They are not only starting points, but also reference points which provide initial alignment for other minutiae. The initial pair generation process is a coarse to fine procedure. Details of initial pair generation are summarized in Algorithm 1. $ComputeWeakSimilarity$ computes the minutiae similarity by multiplying the similarities of orientation and ridge period descriptors, i.e., $S_o \times S_r$, and $ComputeStrongSimilarity$ computes the minutiae similarity by multiplying the similarities of orientation descriptors, ridge period descriptors, and local minutiae structures, i.e., $S_o \times S_r \times S_m$. The computation of S_o and S_r can be made very efficient by using a table look up procedure instead of exponential float computing. On the other hand, the local minutiae structure matching is much slower than orientation or ridge period descriptors. Parameters in Algorithm 1 are defined as $N_1^I = \min\{100, |MQ|\}$ and $N_2^I = \min\{50, N_1^I\}$, where $|MQ|$ is the number of minutiae in the query minutiae set MQ. After two rounds of similarity computation and correspondence selection, the most similar minutiae pairs can be found by sorting. It should be noted that the initial minutiae correspondences need not be one-to-one. We only select the most similar minutiae pairs within the same clusters.

```
Algorithm 1. Initial mated minutiae pair generation.
Input: Query minutiae list $MQ = \{G_1^q, G_2^q, \ldots, G_{K_c}^q\}$
  and template minutiae list $MT = \{G_1^t, G_2^t, \ldots, G_{K_c}^t\}$,
  where $K_c$ is the number of clusters, $G_i^q$ and $G_i^t$ are
  the minutiae subset of cluster $i$ in query and template
  minutiae list, respectively.
Output: A list of initial mated minutiae pairs $I$.
  $I' \leftarrow \{\}$, $I \leftarrow \{\}$
  for $i$ from 1 to $K_c$ do
    for each minutia $m^q$ in $G_i^q$ do
      $Sim_{max} \leftarrow 0$
      $p \leftarrow \{\}$
      for each minutia $m^t$ in $G_i^t$ do
        $Sim_{or} \leftarrow ComputeWeakSimilarity(m^q, m^t)$
        if $Sim_{or} > Sim_{max}$ then
          $Sim_{max} \leftarrow Sim_{or}$
          $p \leftarrow (m^q, m^t, Sim_{or})$
        end if
      end for
      $I' \leftarrow I' \cup p$
    end for
  end for
  //The first round selection.
  $I' \leftarrow$ top $N_1^I$ pairs in $I'$ according to the degree similarity
  for each pair $(m^q, m^t, Sim_{or})$ in $I'$ do
    $Sim_{orm} \leftarrow ComputeStrongSimilarity(m^q, m^t)$
    $I \leftarrow I \cup (m^q, m^t, Sim_{orm})$ // update similarity
 end for
  //The second round selection.
  $I \leftarrow$ top $N_2^I$ pairs in $I$ according to the degree of
  updated similarity
  return Initial mated minutiae pair list $I$.
```

3.3 Match Propagation

Given an initial mated minutiae pair obtained in the previous steps as seeds, the minutiae match propagation procedure is given in Algorithm 2. The function $Pushback$ pushes one minutiae pair into the stack S, and the corresponding $Popup$ function pops a minutiae pair up from the top of stack S. $MatchStruct$ matches the corresponding local structures of minutiae pair p and returns a list of matched local minutiae pairs as described in Section 3.1. Th_{sim} is a predefined threshold value (Th_{sim} is set to 0.0015). Function $GetTransPara$ computes the rigid transformation parameters for a given minutiae pair, and function $WithinLimit$ checks if two rigid transformation parameter sets are sufficiently close to each other. Suppose $T_0 = (\Delta x_0, \Delta y_0, \Delta\alpha_0)$ and $T = (\Delta x, \Delta y, \Delta\alpha)$, where Δx and Δy are the translation parameters along x- and y-axes, and $\Delta\alpha$ is the rotation parameter. $WithinLimit$ function returns true if the following conditions hold:

$$\begin{cases} |\Delta x_0 - \Delta x| < \Delta T_x, \\ |\Delta y_0 - \Delta y| < \Delta T_y, \\ \min(|\Delta\alpha_0 - \Delta\alpha|, 180 - |\Delta\alpha_0 - \Delta\alpha|) < \Delta T_\alpha, \end{cases}$$

where ΔT_x, ΔT_y, and ΔT_α are three predefined thresholds. In this paper, they are set to 40, 40, and 15, respectively.

Algorithm 2. Match propagation from an initial mated minutiae pair.

```
Input: Query minutiae list $MQ$ and template minutiae
  list $MT$, all the minutiae in $MQ$ and $MT$ are initially
  set as available; an initial minutiae pair $p_0 = (m_0^q, m_0^t, s_0)$;
  an empty stack $S$.
Output: Minutiae correspondence list $L$.
  $L \leftarrow \{p_0\}$
  $Pushback(S, p_0)$
  mark $m_0^q$ and $m_0^t$ as not available
  $T_0 \leftarrow GetTransPara(m_0^q, m_0^t)$
  while $S$ is not empty do
    $p \leftarrow Popup(S)$
    $l \leftarrow MatchStruct(p, MQ, MT)$
    for each $(m^q, m^t)$ in $l$ do
      if $m^q$ or $m^t$ is not available then
        continue
      else
        mark $m^q$ and $m^t$ as not available
        $Sim_{orm} \leftarrow ComputeStrongSimilarity(m^q, m^t)$
        $L \leftarrow L \cup \{(m^q, m^t, Sim_{orm})\}$
        $T \leftarrow GetTransPara(m^q, m^t)$
        if $Sim_{orm} > Th_{sim}$ and $WithinLimit(T_0, T)$
        then
          $Pushback(S, (m^q, m^t, Sim_{orm}))$
        end if
      end if
    end for
  end while
  return Minutiae correspondence list $L$.
```

During the minutiae pair propagation, we set two constraints to determine whether the newly found minutiae pairs can be used as new starting points to search for correspondences. The first constraint is $Sim_{orm} > Th_{sim}$. This constraint ensures that the newly found mated minutiae have sufficiently high similarity to avoid false minutiae correspondences. The second constraint is that the difference between the rigid transformation parameters obtained from the newly found pairs and the initial rigid transformation parameters estimated from the initial mated minutiae pairs should be within a certain range. This constraint controls the amount of nonlinear distortion that can be tolerated.

4 Match Score Computation

For different true initial minutiae pairs, the match propagation procedure may output very similar matching results due to the nature of local compatibility. Among these match propagation results, not all of them are needed; we only need the best one or the one with enough correspondences to provide robust matching. Given the kth initial mated minutiae pair, where $k = 1, \ldots, N_2^I$, match propagation is applied to obtain additional minutiae correspondences in the whole palmprint region. If the number of matched minutiae, M, is less than N_1^L, then the match score is set to zero and we proceed to the next initial pair; otherwise, the query palmprint is aligned based on the current initial mated minutiae pair, and the overlapping region of convex hulls determined by matched minutiae lists is obtained. Parameter N_1^L can filter out most of the impostor palmprint matchings. In this paper, N_1^L is set to 5. Three types of similarities are computed and fused together to get the match score associated with the initial minutiae pair if $M \geq N_1^L$. They are 1) orientation field similarity, 2) ridge density map similarity, and 3) average minutiae similarity. Dividing the common regions into 16×16 nonoverlapping blocks, the global orientation field similarity is computed by

$$Score_o = \frac{1}{Num} \sum_{i=1}^{Num} \exp\left(-\frac{\Delta o_i}{\mu_o}\right) \times \frac{N_o}{N_o + 900}, \tag{9}$$

where Num is the total number of blocks in the overlapped region, Δo_i is the orientation difference of the ith block, μ_o is the same parameter as used in (5), and $N_o \leq Num$ is the number of blocks with the average ridge orientation difference less than 22.5 degree. Similarly, the global ridge density map similarity is obtained by

$$Score_r = \frac{1}{Num} \sum_{i=1}^{Num} \exp\left(-\frac{\Delta r_i}{\mu_r}\right) \times \frac{N_r}{N_r + 900}, \tag{10}$$

where Δr_i is the ridge period difference of the ith block, $N_r \leq Num$ is the number of blocks with average ridge period difference less than 2 pixels, and μ_r is the same parameter as used in (6). The minutiae features-based similarity is

$$Score_m = \sum_{i=1}^{M} Sim_{orm}^i, \tag{11}$$

where Sim_{orm}^i is the similarity of the ith minutiae correspondence in L obtained by match propagation. The kth match score is

TABLE 2
Latent Palmprint Databases Used in This Study

Datasets	Noblis	MSP	RS&A
No. of latents	46	54	346
No. of palms	8	22	88
Resolution (ppi)	1000	400	1200
Average no. of minutiae	126	132	139
Standard deviation of no. of minutiae	50	58	105

$$Score^k = Score_o \times Score_r \times Score_m. \qquad (12)$$

If $M > N_2^L$, the matching algorithm outputs $Score^k$ as the final match score, ignoring the rest of initial pairs. Otherwise, the final match score is set to $\max\{Score^k\}$. N_2^L is set to 200 in this paper.

5 Experiments

5.1 Palmprint Databases

To our knowledge, there is no public domain latent and mated full palmprint database that is available. Our latent palmprint images come from Noblis [33], the Forensic Science Division of Michigan State Police (MSP) [34] and Ron Smith and Associates Inc. (RS&A)[35]. The Noblis latent database consists of 46 latent palmprints that come from eight different palms. The MSP latent database consists of 54 latent palmprints that come from 22 different palms, and the RS&A database consists of 346 latent palmprints that come from 88 different palms. These three latent databases are combined together to form a database of 446 latent palmprints. The foreground area of latent palmprints is manually marked in the original images. Table 2 summarizes the latent palmprint databases used in our experiments. We see RS&A database has much higher standard deviation of no. of minutiae than the other two databases. That is because RS&A database contains lots of latent palmprints with small friction ridge region. The background databases of full palmprints consists of eight subsets. Table 3 summarizes each of these full palmprint subsets. The total number of full palmprint images in the background database is 12,489. All latents or full palmprint images used in our experiments that are not at 500 ppi resolution are up or down sampled to 500 ppi using bicubic interpolation.

5.2 Evaluation of Minutiae Clustering

The number of clusters, K_c, is an important parameter in our method. A large number of clusters can divide minutiae set into more groups, resulting in more variety of cluster centroids, but it may also lead to missing some true minutiae correspondences. Furthermore, the clusters trained from different sets of descriptors may be slightly different. To test the number of clusters and measure their stability, we choose eight different values of K_c, namely $K_c = 1, 2, 4, 8, 16, 32, 48$, and 64, where $K_c = 1$ means minutiae clustering was not used. All other parameters used in our algorithm are summarized in Table 4. For each number of clusters, we train 10 sets of cluster centroids independently for matching, with each set of cluster centroids trained from 800 full palmprints (not including mated full palmprints) that were randomly selected from the background database. Our experiments were conducted in the identification mode (i.e., 1:N matching). For each latent, the identification system searches for its mated full palmprint in the background database and returns a list of candidates in order of their similarity.

Two different sets of clusters, i.e., orientation descriptor based (OriDes) and ridge period descriptor based (RidDes), are evaluated in our experiments. Fig. 9a shows the average rank-1 identification rates and standard deviations as a function of the number of clusters. In the case of $K_c = 1$, the latents are matched against the background database only once, so the standard deviation is zero. In general, the average rank-1 identification rate drops and standard deviation increases as the number of clusters K_c increases. A larger number of clusters leads to fewer true mated minutiae pairs being clustered together and, consequently, fewer initial pairs of true correspondences can be selected. The standard deviation for all values of K_c is very small (around 0.5), indicating high stability of both OriDes and RidDes cluster centroids. The average rank-1 identification rate when $K_c = 2$ is slightly higher (by 0.18 percent) than $K_c = 1$. We also observe that the OriDes-based clustering performs better than RidDes-based clustering.

Fig. 9b shows the average genuine and impostor matching time for different number of clusters. The computation time is measured on a Debian linux system with 2.3-GHz CPU. We can see a significant drop in computation time when clustering is used to search for initial minutiae pairs. Without clustering, the genuine and impostor match computation times are about 239 and 141 ms, respectively. With OriDes, the computation time

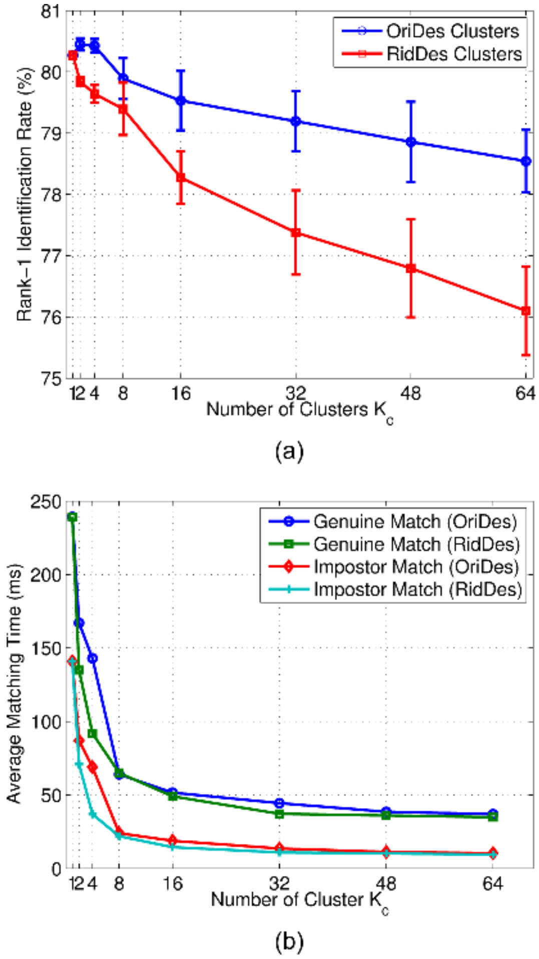
TABLE 3
Background Database of Full Palmprints

Datasets	MSP full palmprint	MSU live-scan	CMC inked [36]	CMC live-scan [36]	THUPALMLAB [7]	MSP mated full	Noblis mated full	RS&A mated full
No. of palms	$9,701$	116	160	106	160	22	8	196 [a]
No. of images	$9,701 \times 1$	$(100 \times 10) + (16 \times 1)$	160×1	106×1	160×8	22×1	8×1	196×1
Resolution (ppi)	500	1000	1000	500	500	400	1000	500
Average no. of minutiae	784	1129	1260	855	879	901	1353	1325
Standard deviation of no. of minutiae	282	207	403	243	213	227	230	410

[a] *Among these 196 images, 88 palmprints are mated full palmprints of the latent RS&A database.*

TABLE 4
Summary of Parameter Values Used in the Proposed Latent-to-Full Palmprint Matching Algorithm

Parameter(s)	Description	Value
1. Parameters for computing minutiae similarity measure		
μ_o, μ_r	Normalization terms for computing orientation and ridge period descriptors similarities in Eqs. (5) and (6).	4, 2
R, K	Parameters for discretizing radius and angle in a local structure.	4, 8
r_{max}	Radius of the outermost circle in query local structure. Parameter value is sensitive to the connectivity of ridge region	180
Th_α, Th_r, Th_t	Parameters in Eq. (7)	8, 20, 15
M_0	Parameter in Eq. (8)	10
a_t, b_t, a_r, b_r, a_α, b_α	Parameters for computing s_i in Eq. (8)	2, 8, 3, 20, 3, 15
a_Q, b_Q	Parameters for computing Q in Eq. (8)	40, 65
2. Parameters for initial minutiae pair generation		
N_1^I	No. of initial minutiae pairs selected in the first round	$\min\{100, \lvert MQ\rvert\}$
N_2^I	No. of initial minutiae pairs selected in the second round	$\min\{50, N_1^I\}$
3. Parameters for match propagation		
Th_{sim}	Minimum minutiae similarity value used for propagation. Parameter value is sensitive to propagation	0.0015
ΔT_x, ΔT_y, ΔT_α	Control the rigid transformation parameters of newly found minutiae pairs within a limited range of those parameters derived from the initial minutiae pair	40, 40, 15
4. Parameters for match score computation		
N_1^L	Minimum no. of minutiae pairs for computing the mach score. Parameter value determines the impostor match time	5
N_2^L	Parameter to control further match score computation. If the no. of currently matched minutiae pairs exceeds N_2^L, then the remaining initial pairs will be ignored. Parameter value determines the genuine match time	200



(a)

(b)

Fig. 9. Matching performance of the proposed algorithm as a function of number of clusters, where $K_c = 1$ means no clustering of minutiae descriptors was performed. (a) Average rank-1 identification rates along with standard deviations. (b) Average genuine and impostor matching time.

decrease rapidly when K_c increases from 2 to 16. With the number of clusters, K_c, further increasing to 64, the computation time decreases to 37 and 10 ms, respectively. So, the proposed minutiae clustering method makes the latent palmprint matching algorithm up to 14 times faster depending on the number of clusters. The computation time stabilizes as K_c increases beyond a certain value (around 16) because the time saved by clustering becomes negligible with respect to the time spent in minutiae match propagation. RidDes clusters are slightly better than OriDes clusters in terms of matching time. As a tradeoff between matching accuracy and computation efficiency, we suggest using OriDes cluster centroids with $K_c = 16$ for latent-to-full palmprint matching.

5.3 Latent-to-Full Palmprint Matching

Fig. 10 shows the Cumulative Match Characteristic (CMC) curves for matching latent images in MSP, Noblis, and RS&A databases, separately, against the background database of 12,489 full palmprints by the proposed algorithm with 16 OriDes clusters. The reported performance is the average identification rate based on 10 sets of different cluster centroids. The average identification rates on MSP and Noblis databases are much higher than that on the RS&A database. This is because MSP and Noblis databases have higher image quality than palmprint images in the RS&A database.

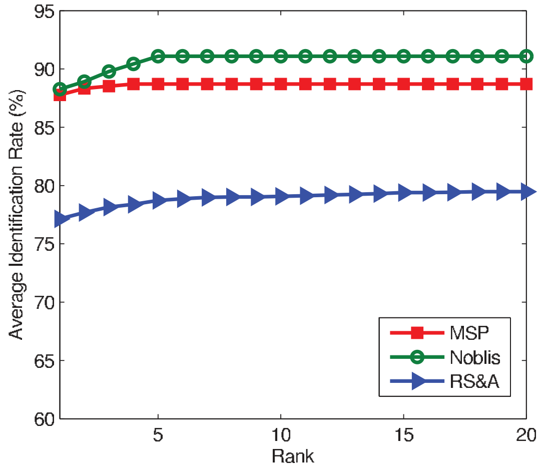


Fig. 10. CMC curves for matching latent palmprints in MSP, Noblis, and RS&A databases against a background database of 12,489 full palmprints by the proposed algorithm with 16 OriDes clusters.

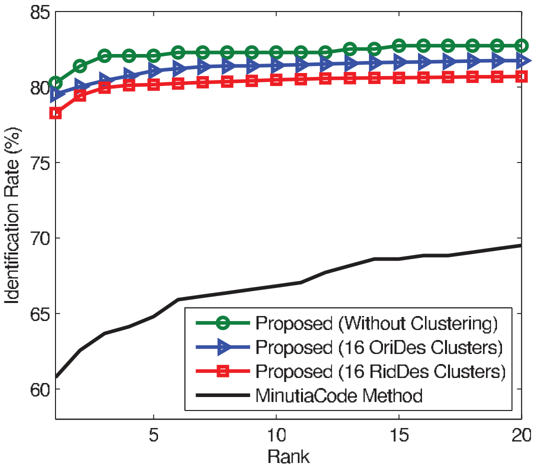


Fig. 11. CMC curves of the proposed method and the MinutiaCode method [5] in searching 446 latents against a background database of 12,489 full palmprints.

The proposed latent-to-full palmprint matching algorithm is compared with the MinutiaCode method [5] on the combined latent database. The same feature extraction algorithm as reported in [5] is used to extract features for the two methods. Fig. 11 shows the CMC curves, which demonstrate that a significant performance improvement is achieved by the proposed algorithm over the MinutiaCode method [5]. With 16 OriDes clusters, the rank-1 identification rate of the proposed algorithm is about 18.6 percent higher than that of the MinutiaCode method. Recall that the purpose of minutiae clustering is to make the latent palmprint matching very efficient. Our algorithm not only meets this requirement, but it also improves the matching performance over state-of-the-art algorithms.

The accuracy improvement achieved by the proposed algorithm is mainly because of the following two reasons:

1. In palmprint matching, false correspondences can be easily established if their local compatibility is not considered because of the large number of minutiae in palmprints. Our algorithm reduces these false correspondences by minutiae match propagation.
2. The minutiae correspondences are established starting from an initial pair and gradually expanding to the whole palmprints without applying any global rigid alignment. As a result, our algorithm is robust to nonlinear distortion.

Fig. 12 shows three examples where latent-to-full palmprint matching is successful. Fig. 13 shows two examples where the latent is matched to its true mate at ranks higher than 1,000. We see from the directions of match propagation denoted by blue arrows in these figures that each minutia is matched several times in the local structures of its neighboring minutiae. So, the propagation sometimes moves backward. This increases the chance of mated minutiae pairs being matched. We can also see that the depth of propagation tree starting from a genuine minutiae pair is much deeper than that from impostor minutiae pairs. Since impostor minutiae pairs lack high local compatibility, the propagation stops quickly. Most of the identification failures are due to low-quality palmprints and small overlap between latent and full palmprints.

5.4 Full-to-Full Palmprint Matching

Due to the large number of minutiae (average number of minutiae is about 1,000) in full palmprint images, it is highly probable to find more correspondences between two full palmprint images in the initial pair generation stage. To further reduce the computation time of full-to-full palmprint matching, we simplify the latent-to-full version of palmprint matching algorithm to make it more suitable for full-to-full palmprint matching. First, the minutiae clusters for full palmprints are "compound" clusters obtained by combining the OriDes and RidDes clusters using Cartesian product.[1] Two minutiae are regarded as belonging to the same cluster only if they are clustered to the same OriDes centroid and RidDes centroid. Second, during the initial pair generation, only the orientation descriptor similarity is used in the $ComputeWeakSimilarity$ function, and the second stage selection of initial pairs using the $ComputeStrongSimilarity$ function is not applied. Third, only the top 5 most similar pairs of minutiae correspondences are selected as initial pairs. Fourth, the global orientation field similarity and ridge density map similarity are not used. The final match score is simply the sum of minutiae similarity $Score_m$ defined in (11).

The proposed algorithms, both simplified and nonsimplified full-to-full palmprint matcher, were evaluated on a public domain full palmprint database, THUPALMLAB [7], under the verification protocol, and compared with the segment-based method [7], MinutiaCode method [5], and MCC-based method [17]. The total number of palms in THUPALMLAB is 160, with eight impressions for each palm. The genuine matches are formed by matching every pair among the eight impressions of each palm, and the impostor matches are simulated by matching the first impression of each palm against the first impression of all the other palms. So, the total number of genuine and impostor matches are 4,480 and 12,720, respectively.[2]

Our experimental results show that the proposed algorithms, both nonsimplified and simplified versions, achieve much higher accuracy than the segment-based method and the MinutiaCode method. The equal error rates (EERs) of the segment based and MinutiaCode methods are 1.46 and 1.47 percent, respectively. The nonsimplified version of the proposed algorithm with 16 OriDes clusters achieves an EER of 0.11 percent, and the simplified version of the proposed algorithm with 64 OriDes clusters and 16 RidDes clusters achieves an EER of 0.13 percent.

The EER of the MCC-based algorithm [17] is estimated to be about 0.01 percent, but its ZeroFMR is 0.48 percent compared to our performance of 0.29 percent using the simplified version. Given the small number of genuine matches (4,480 in our experiments and 3,360 in [17] for MCC), the difference between the two algorithms is just three false nonmatches when FMR $= 0$ percent. However, the matching speed of the proposed algorithm is significantly faster than the MCC-based matching algorithm. The average genuine and impostor matching times of the

[1] Each minutia is clustered into two types of cluster centroids, one for OriDes and another one for RidDes.

[2] The experimental results of MCC-based algorithm are from [17].

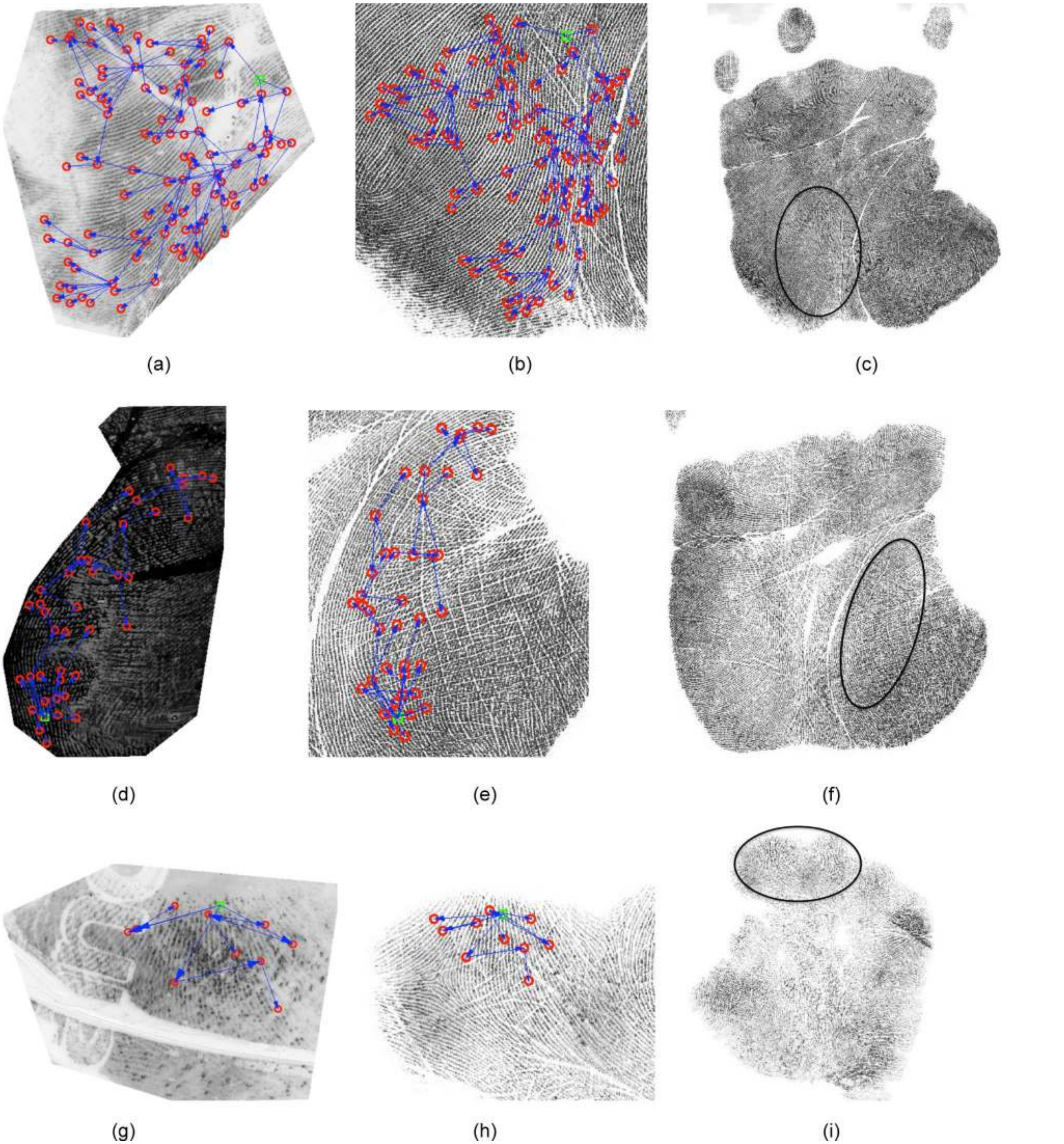


Fig. 12. Three examples of successful latent-to-full palmprint matching with OriDes clusters and $K_c = 16$. (a), (d), and (g) are three latent palmprint images from hypothenar, thenar, and interdigital regions, respectively. The second column contains three zoomed in images of mated full palmprints, and the third column contains three mated full palmprint images with target areas (where the latent print comes from) marked by ellipses. The red circles denote matched minutiae, green squares denote the initial minutiae pairs, and blue arrows denote the direction of minutiae match propagation. Mated full palmprints of latents (a) and (d) were successfully identified at rank-1 and mated print of latent (g) was identified at rank-2.

nonsimplified algorithm are 133.2 and 102.2 ms, respectively, and the average genuine and impostor matching times of our simplified algorithm are 14.7 and 2.0 ms, respectively. The simplified algorithm significantly reduces the computation time of full-to-full palmprint matching without any significant loss in the matching accuracy.

Fig. 14 shows an example of the proposed algorithm's performance in matching full palmprint images with large nonlinear distortion.

5.5 Evaluation Results by FVC-onGoing

The proposed algorithm was also tested on the public palmprint verification benchmarks, PV-Full-1.0, and PV-Partial-1.0, hosted on FVC-onGoing [37]. The number of genuine and impostor attempts in PV-Full-1.0 benchmarks are 2,800 and 4,950, respectively, and the number of genuine and impostor attempts in PV-Partial-1.0 benchmarks are 400 and 10,000, respectively. For PV-Partial-1.0, we used the latent-to-full matcher with $K_c = 16$ for OriDes,

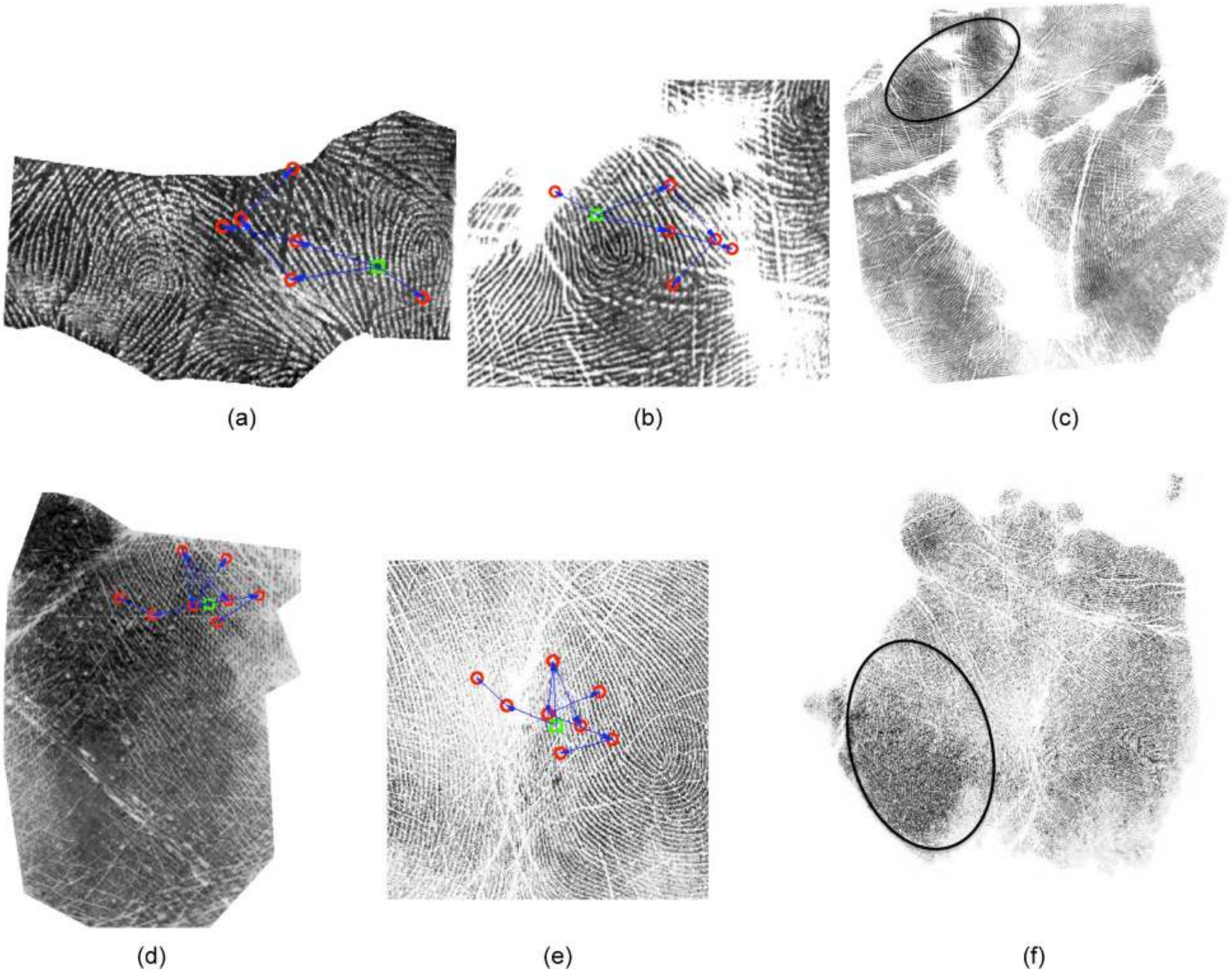


Fig. 13. Two failure cases of latent palmprint matching with OriDes clusters and $K_c = 16$. (a) a latent palmprint image with small overlap with its mated full print; (b) zoomed in area of matched region of (a) with incorrectly paired minutiae; (c) target region of (a) in the mated full palmprint; (d) a latent palmprint image with poor quality; (e) zoomed in area of matched region of (d) with incorrectly paired minutiae; (f) target region of (d) in the mated full palmprint. The red circles denote matched minutiae, green squares denote the initial minutiae pairs, and blue arrows denote the direction of minutiae match propagation. The retrieval ranks of (a) and (d) with the true mates are 1,244 and 8,427, respectively.

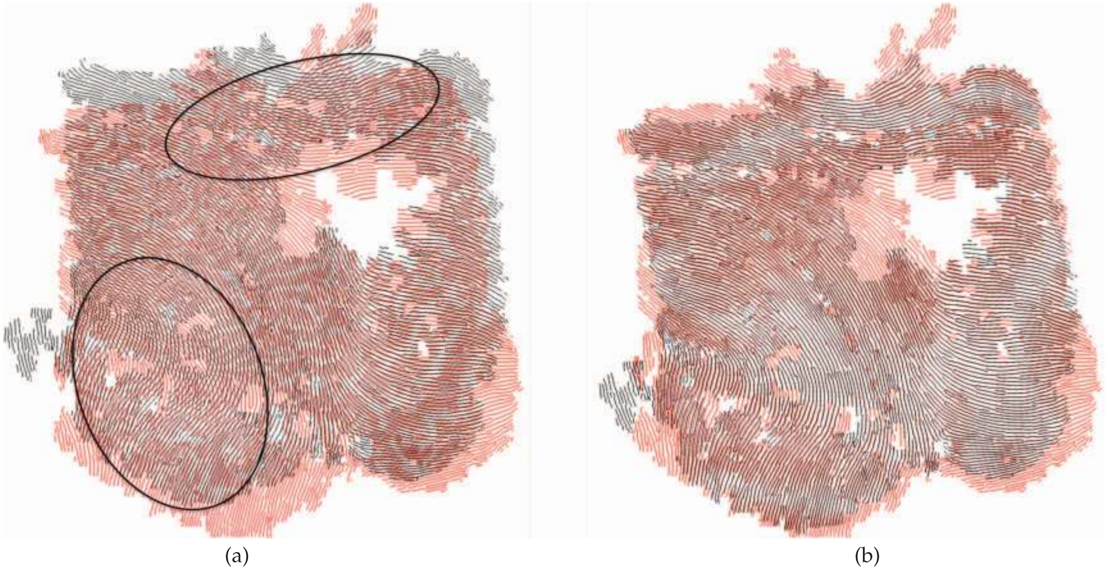


Fig. 14. An example of nonlinearly distorted full palmprint matching by the proposed method. (a) Rigid alignment based on initial mated minutiae pair, and (b) affine alignment based on mated minutiae pairs generated by match propagation.

TABLE 5
Comparison of Matching Performance on PV-Full-1.0 by FVC-onGoing [37]

Algorithm	Avg. match time	Avg. model size	EER (%)	FMR1000 (%)	ZeroFMR (%)
Mnt	428ms	93K	4.07%	7.14%	100%
MCC [17]	228ms	237K	2.57%	4.04%	19.68%
MinutiaCode [5]	2148ms	617K	10.73%	44.25%	57.14%
Proposed	86ms	3344K	5.60%	6.86%	10.21%

TABLE 6
Comparison of Matching Performance on PV-Partial-1.0 by FVC-onGoing [37]

Algorithm	Avg. match time	Avg. model size	EER (%)	FMR1000 (%)	ZeroFMR (%)
MCC [17]	165ms	143K	3.50%	4.50%	6.25%
MinutiaCode [5]	1159ms	356K	6.26%	28.50%	40.50%
Proposed	115ms	1948K	5.00%	8.75%	9.25%

and for PV-Full-1.0, we used the full-to-full matcher (simplified) with $K_c = 16$ for both OriDes and RidDes.

Tables 5 and 6 are the performance comparisons with the published results on FVC-onGoing. We see that while the MCC-based matching algorithm achieves the best accuracy, the proposed "coarse to fine" algorithm has the lowest average matching time.

In the proposed algorithm, the cluster centers need to be learned from the training data. The cluster centers used in our algorithm submitted to FVC-onGoing were trained on 800 full palmprints that were randomly chosen from the background database (not including the mated full palmprints), consisting of MSP full, MSU live, CMC inked or live scan databases. We did not use the PV-TEST-FULL or PV-TEST-PARTIAL for training because the number of palmprints in it is too small (only 80 full palmprints and 40 partial palmprints).

The image characteristics of palmprints in PV-TEST-FULL (or PV-TEST-PARTIAL) are quite different from the palmprints in databases that we used (MSP full, MSU live, CMC inked or live scan). Each impression of a palm in PV-TEST-FULL was collected under different controlled conditions, such as torsion, traction, shrinking or expanding fingers, humidity, and so on, whereas the palmprints in our database were collected under normal conditions.

This difference in characteristics of the palmprint databases used by us and PV-TEST-FULL (or PV-TEST-PARTIAL) partly explains our somewhat lower matching performance (PV-TEST-FULL and PV-TEST-PARTIAL were collected using the same protocol as PV-Full-1.0 and PV-Partial-1.0).

5.6 Computational Complexity

To evaluate and compare the computation time, we tested the proposed algorithm, the segment-based algorithm and the MinutiaCode algorithm[3] on the same computer with Dual-Core AMD Opteron (tm) 2.20-GHz CPU, 1.00-GB RAM, and Windows XP operating system. The segment-based algorithm is not tested on the latent-to-full database, because this algorithm is designed to work only for full-to-full palmprint matching. To apply it to latent palmprint matching, manual alignment is needed, which is not practical in operational scenarios because most of the latent palmprint images only contain a small portion of the palm. In some instances, we are not even able to determine which part (e.g., interdigital, hypothenar or thenar) of the palm the latent comes from. The computation time of the proposed full-to-full palmprint matching algorithm is reported for the simplified algorithm. We did not implement the MCC-based matching algorithm [17]. To compare the matching efficiency with MCC, the match time on PV-Full-1.0 and PV-Partial-1.0 are used for comparison (evaluated by FVC-onGoing). As shown in Table 7, the proposed algorithm is extremely efficient in full-to-full matching compared with the segment-based matching algorithms [7], MinutiaCode [5], and MCC-based algorithm [17]. For full-to-full palmprint matching, the genuine and impostor matching are, respectively, 17 and 104 times, faster than the segment-based algorithm [7], and are, respectively, 85 and 437 times faster than the MinutiaCode algorithm [5]. For latent-to-full palmprint matching, the proposed algorithm for genuine and impostor matching times are, respectively, about 12 and 22 times faster than the MinutiaCode-based method. Another advantage of our algorithm is that the impostor matching is significantly faster than the genuine matching, which makes our algorithm very suitable for operating in the identification mode. Compared with MCC in FVC-onGoing, the proposed algorithm is 2-3 times faster in full-to-full matching. For latent-to-full matching, the genuine match time is very close, but impostor match time of the proposed algorithm is faster than MCC.

6 Conclusions

We have proposed an efficient and robust latent-to-full palmprint matching algorithm, which can also be applied to full-to-full palmprint matching. To deal with the large number of minutiae in palmprint images, a minutiae clustering algorithm is proposed to group minutiae into several clusters based on the local features (i.e., ridge orientation and ridge period) in minutiae neighborhood. These clusters can greatly narrow down the search space of initial minutiae correspondences so as to drastically cut down the computation time. The initially selected minutiae correspondences are fed to a fine matching procedure, called minutiae match propagation, to establish the minutiae correspondences in the whole palmprint. The proposed

TABLE 7
Comparison of Match Time (ms) with Published Palmprint Matching Algorithms

Algorithm	Full-to-full (ms)		Latent-to-full (ms)	
	Genuine match	Impostor match	Genuine match	Impostor match
MinutiaCode [5]	4,703	3,496	1,753	1,095
Segment-based[a][7]	964	834	-	-
MCC-based [17]	(237)	(224)	(162)	(166)
Proposed	55(96)	8(80)	141(167)	50(113)

(The numbers in brackets are match time (ms) evaluated on PV-Full-1.0 and PV-Partial-1.0 by FVC-onGoing [37]). [a] The above speed was measured using an earlier released version of this algorithm. A faster version has since been released on the authors' website. Because the segment-based method cannot be directly applied to latent palmprint images, we only report its full-to-full matching time here.

[3] The implementations of these two algorithms are available from the authors' website: http://ivg.au.tsinghua.edu.cn/.

minutiae match propagation algorithm ensures that local compatibility among the minutiae correspondences is satisfied to quickly reject false minutiae correspondences. Our algorithm does not require global rigid alignment between the two palmprints. This matching strategy is especially suitable for palmprint matching since palmprints contain a large number of minutiae and suffer from nonlinear distortion.

Experimental results demonstrate the high efficiency and robustness of the proposed algorithm for both latent-to-full and full-to-full palmprint matching. With 16 OriDes clusters, the proposed algorithm has a 79.4 percent average rank-1 accuracy in searching 446 latent palmprints against a background database of 12,489 full palmprints, which is 18.6 percent higher than the performance of the MinutiaCode method [5]. To evaluate the computational efficiency, the proposed algorithm was tested on three different computing systems: 1) a linux server system for evaluating the computation time under different numbers of clusters, 2) a Windows XP system for comparison with other palmprint matching algorithms which were implemented on the same platform, and 3) a public platform for palmprint matching algorithm evaluation hosted by FVC-onGoing [17]. The results show that the matching can be significantly speeded up (5 to 14 times faster, on average) by introducing minutiae clustering without adversely affecting the matching accuracy. Compared with the MinutiaCode method [5] in latent-to-full palmprint matching, our method is 12 and 22 times faster for genuine and impostor matchings, respectively. Our algorithm is also the fastest palmprint matching algorithm published in FVC-onGoing.

The proposed algorithm (with/without simplification) can also be applied to full-to-full palmprint matching. Compared with two state-of-the-art full-to-full palmprint matching algorithms, the proposed algorithm is extremely fast and accurate. The EER of our algorithm is 10 times lower than that of the segment based [7] and MinutiaCode [5] algorithms. For genuine and impostor matchings, the proposed algorithm is, respectively, 17 and 104 times, faster than the segment-based method, and is, respectively, 85 and 437 times, faster than the MinutiaCode method.

Spurious minutiae and small overlap between palmprints remain two of the major challenges in latent-to-full palmprint matching. Due to the presence of immutable creases in palmprints and uncontrollable image acquisition conditions, a large number of spurious minutiae might be extracted especially in palmprint images. These spurious minutiae could lead to many false initial minutiae pairs which, in turn, leads to field matches. Certain types of latent palmprints, e.g., writer's palmprints might have very small overlap with the full palmprints. This makes it difficult to find a sufficient number of true minutiae correspondences. In our future work, we will further improve the proposed algorithm keeping the above two aspects in mind.

ACKNOWLEDGMENTS

Anil Jain's research was partially supported by the World Class University (WCU) program funded by the Ministry of Education, Science and Technology through the National Research Foundation of Korea (R31-10008). Eryun Liu's research was partially supported by the National Natural Science Foundation of China under Grant No. 61100234. All correspondence regarding this paper should be addressed to A.K. Jain.

REFERENCES

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

[2] D. Zhang, W. 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.

[3] A. Kong, D. Zhang, and M. Kamel, "A Survey of Palmprint Recognition," *Pattern Recognition,* vol. 42, no. 7, pp. 1408-1418, 2009.

[4] 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,* vol. 2, pp. 1849-1854, 1998.

[5] A. Jain and J. Feng, "Latent Palmprint Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 6, pp. 1032-1047, June 2009.

[6] J. Dai and J. Zhou, "Multifeature-Based High-Resolution Palmprint Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 33, no. 5, pp. 945-957, May 2011.

[7] J. Dai, J. Feng, and J. Zhou, "Robust and Efficient Ridge-Based Palmprint Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 34, no. 8, pp. 1618-1632, Aug. 2012, doi:10.1109/TPAMI.2011.237.

[8] R. Wang, D. Ramos, and J. Fierrez, "Latent-to-Full Palmprint Comparison Based on Radial Triangulation under Forensic Conditions," *Proc. Int'l Joint Conf. Biometrics,* pp. 1-6, 2011.

[9] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition,* second ed. Springer-Verlag, 2009.

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

[11] 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, Mar. 2006.

[12] Y. He, J. Tian, L. Li, H. Chen, and X. Yang, "Fingerprint Matching Based on Global Comprehensive Similarity," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 6, pp. 850-862, June 2006.

[13] D. Ashbaugh, *Quantitative-Qualitative Friction Ridge Analysis: An Introduction to Basic and Advanced Ridgeology.* CRC, 1999.

[14] A. Jain and M. Demirkus, "On Latent Palmprint Matching," technical report, Michigan State Univ., http://biometrics.cse.msu.edu/Publications/Palmprints/JainDemirkusOnLatentPalmprintMatching08.pdf, 2008.

[15] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[16] Neurotechnology, "VeriFinger 4.2 SDK," http://www.neurotechnology.com/verifinger.html, 2012.

[17] R. Cappelli, M. Ferrara, and D. Maio, "A Fast and Accurate Palmprint Recognition System Based on Minutiae," *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics,* vol. 42, no. 3, pp. 956-962, June 2012.

[18] K. Cao, X. Yang, X. Chen, X. Tao, Y. Zang, J. Liang, and J. Tian, "Minutia Handedness: A Novel Global Feature for Minutiae-Based Fingerprint Matching," *Pattern Recognition Letters,* vol. 33, no. 10, pp. 1411-1421, 2012.

[19] S. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans. Information Theory,* vol. 28, no. 2, pp. 129-137, Mar. 1982.

[20] A. Jain and R. Dubes, *Algorithms for Clustering Data.* Prentice-Hall, 1988.

[21] 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.

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

[23] E. Liu, H. Zhao, J. Liang, L. Pang, H. Chen, and J. Tian, "Random Local Region Descriptor (RLRD): A New Method for Fixed-Length Feature Representation of Fingerprint Image and Its Application to Template Protection," *Future Generation Computer Systems,* vol. 28, no. 1, pp. 236-243, 2012.

[24] M. Lhuillier and L. Quan, "Match Propagation for Image-Based Modeling and Rendering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 8, pp. 1140-1146, Aug. 2002.

[25] J. Kannala and S. Brandt, "Quasi-Dense Wide Baseline Matching Using Match Propagation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

[26] Q. Zhu, B. Wu, and Y. Tian, "Propagation Strategies for Stereo Image Matching Based on the Dynamic Triangle Constraint," *ISPRS J. Photogrammetry and Remote Sensing,* vol. 62, no. 4, pp. 295-308, 2007.

[27] H. Wang, S. Yan, J. Liu, X. Tang, and T. Huang, "Correspondence Propagation with Weak Priors," *IEEE Trans. Image Processing,* vol. 18, no. 1, pp. 140-150, Jan. 2009.

[28] J. Yao and W. Cham, "Clustering-Based Match Propagation for Image-Based Rendering from Multiple Images," *Proc. Int'l Conf. Image Processing,* vol. 5, pp. 2969-2972, 2004.

[29] J. Yao and W. Cham, "3D Modeling and Rendering from Multiple Wide-Baseline Images by Match Propagation," *Signal Processing: Image Comm.,* vol. 21, no. 6, pp. 506-518, 2006.

[30] A. Hale, "Morphogenesis of Volar Skin in the Human Fetus," *Am. J. Anatomy,* vol. 91, no. 1, pp. 147-181, 1952.

[31] S. Chikkerur, A. Cartwright, and V. Govindaraju, "K-plet and Coupled BFS: A Graph Based Fingerprint Representation and Matching Algorithm," *Proc. Int'l Conf. Biometrics,* pp. 309-315, 2005.

[32] K. Cao, E. Liu, L. Pang, J. Liang, and J. Tian, "Fingerprint matching by Incorporating Minutiae Discriminability," *Proc. Int'l Joint Conf. Biometrics,* pp. 1-6, 2011.

[33] Noblis, http://www.noblis.org, 2012.

[34] Forensic Science Division of Michigan State Police, http://www.michigan.gov/msp/0,4643,7-123-1593_3800-15901-,00.html, 2012.

[35] Ron Smith and Assoc., http://www.ronsmithandassociates.com, 2012.

[36] CMC Limited, http://www.cmcltd.com, 2012.

[37] Univ. of Bologna, "FVC-Ongoing: On-Line Evaluation of Fingerprint Recognition Algorithms," https://biolab.csr.unibo.it/fvcongoing/UI/Form/Home.aspx, 2012.

Eryun Liu received the bachelor's degree in electronic information science and technology and the PhD degree in pattern recognition and intelligence system from Xidian University, Xi'an, Shaanxi, China, in 2006 and 2011, respectively. He is currently a postdoctoral fellow in the Department of Computer Science and Engineering, Michigan State University, East Lansing. He is also with Xidian University as an assistant professor. His research interests include biometric recognition, point pattern matching, and information retrieval, with a focus on fingerprint recognition and palmprint recognition.

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 served as the editor-in-chief of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1991-1994). The holder of six patents in the area of fingerprints, he is the author of a number of books, including *Handbook of Fingerprint Recognition* (2009), *Handbook of Biometrics* (2011), *Handbook of Multibiometrics* (2006), *Handbook of Face Recognition* (2005), *BIOMETRICS: Personal Identification in Networked Society* (1999), and *Algorithms for Clustering Data* (1988). He served as a member of the Defense Science Board and The National Academies Committees on Whither Biometrics and Improvised Explosive Devices. 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 has received Fulbright, Guggenheim, Alexander von Humboldt, IEEE Computer Society Technical Achievement, IEEE Wallace McDowell, ICDM Research Contributions, and IAPR King-Sun Fu Awards. He is a fellow of the IEEE, AAAS, ACM, IAPR, and SPIE.

Jie Tian received the PhD degree (with honor) in artificial intelligence from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 1992. From 1995 to 1996, he was a postdoctoral fellow with the Medical Image Processing Group, University of Pennsylvania, Philadelphia. He is also with the Life Science Center, Xidian University, Xian, China. Since 1997, he has been a professor in the Intelligent Medical Research Center, Institute of Automation, China. His current research interests include medical image processing and analysis and pattern recognition. He has authored or coauthored more than 100 research papers in international journals and conferences. He is the Beijing Chapter chair of the Engineering in Medicine and Biology Society of the IEEE. He is a fellow of the IEEE.