Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 40 (2007) 1793–1803
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

Efficient fingerprint search based on database clustering

Manhua Liu, Xudong Jiang*, Alex Chichung Kot

EEE, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore

Received 30 June 2006; received in revised form 10 November 2006; accepted 16 November 2006

Abstract

Fingerprint identification has been a great challenge due to its complex search of database. This paper proposes an efficient fingerprint search algorithm based on database clustering, which narrows down the search space of fine matching. Fingerprint is non-uniformly partitioned by a circular tessellation to compute a multi-scale orientation field as the main search feature. The average ridge distance is employed as an auxiliary feature. A modified K-means clustering technique is proposed to partition the orientation feature space into clusters. Based on the database clustering, a hierarchical query processing is proposed to facilitate an efficient fingerprint search, which not only greatly speeds up the search process but also improves the retrieval accuracy. The experimental results show the effectiveness and superiority of the proposed fingerprint search algorithm.

Keywords: Fingerprint identification; Fingerprint search; Biometrics; Fingerprint classification; Orientation field

1. Introduction

Fingerprint as a kind of human biometric feature has been widely used for personal recognition in the commercial and forensic areas because of its uniqueness, immutability and low cost. In general, fingerprint based recognition systems work in two modes: authentication and identification [1]. In the authentication mode, the user inputs his fingerprint and claims an identity information, then the system verifies whether the input fingerprint is consistent with the claimed identity. In the identification mode, the user input his fingerprint and the system identifies the potential corresponding ones in the database without a claimed identity. Therefore, fingerprint identification requires searching the database for a match, which is more complex than the authentication. Although satisfactory performances have been reported for fingerprint authentication, both the efficiency and accuracy of identification deteriorate seriously by simple extension of a 1:1 authentication procedure to a 1:N identification system [1]. How to efficiently search the fingerprint database is a great challenge. Multi-level matching approaches are proposed to facilitate the database search by incorporating the global and local information of fingerprint [2,3]. The coarse level matching (search) is often used to reduce the search space of the time-consuming fine matching and alleviate the accuracy deterioration of identification [1]. Exclusive classification, fingerprint indexing and continuous classification have been proposed for the coarse level search of database.

Exclusive fingerprint classification is a traditional approach that has been widely investigated in the literature [4–11]. It classifies each fingerprint exclusively into one of the predefined classes such as Henry classes. Although it has some advantages such as human-interpretability, fast retrieval and rigid database partitioning, most automated classification algorithms are able to classify fingerprints into only four or five classes. Moreover, fingerprints are not evenly distributed in these classes. The natural fingerprint distribution of the Henry five classes is 3.7% plain arch, 2.9% tented arch, 33.8% left loop, 31.7% right loop and 27.9% whorl. On average, a query fingerprint still needs to be compared with about 29.48% of database templates in the fine matching of identification. Thus, the exclusive classification cannot sufficiently narrow down the search of database.

* Corresponding author. Tel.: +65 6790 5018; fax: +65 6793 3318.
E-mail address: exdjiang@ntu.edu.sg (X. Jiang).

0031-3203/$30.00 © 2006 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2006.11.007

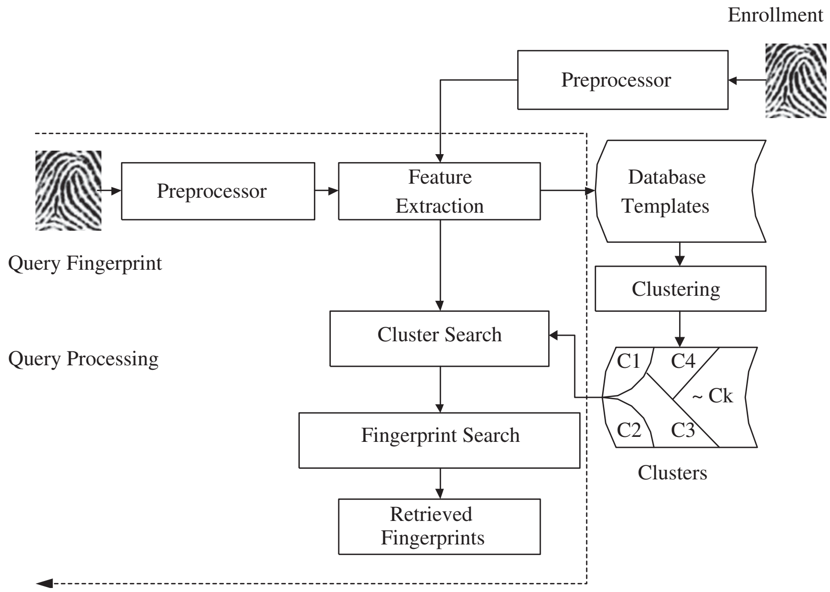

Fig. 1. The overview of the clustering based fingerprint search algorithm.

In fact, it is not necessary to classify fingerprints into human-interpretable classes for an automated identification system. Fingerprint indexing, which divides fingerprint database into a number of bins based on the minutia triplets, was proposed in Refs. [3,12]. This approach classifies fingerprints into more classes (or bins) than the exclusive classification as it exploits the more discriminating features, minutiae. However, the minutia points are the most important local features and widely used in fingerprint fine matching algorithms [2,13–15]. Although this approach can speed up the database search, it should take care to avoid a redundant representation of fingerprint in an identification system. This is because the accuracy deterioration of the identification system is hardly alleviated if the features used in the coarse search and fine matching are strongly correlated.

Continuous classification is proposed to overcome the problems of exclusive classification by representing fingerprint with numerical feature vectors [16–18]. The fingerprint search is performed by comparing the query fingerprint with all database templates and retrieving the closest ones. The tradeoff between retrieval efficiency and accuracy can be easily adapted by adjusting the size of retrieval neighborhood. Although the comparison between the query fingerprint and template is much faster than the fine matching, this full fingerprint search is still prohibitive for large database. Moreover, the continuous classification only ranks the database templates according to their similarities to the query fingerprint while neglecting the similarities among the database templates. This limits the search performance. Although some combined techniques were proposed to improve the performance of fingerprint classification [19,20], further work to facilitate an efficient search of database is still of great interest to the researchers in the area of fingerprint identification.

Data clustering is a crucial technique used in discovering the underlying structure in a data set by unsupervised grouping of the similar patterns. It accelerates the content based image retrieval by comparing the query image with a few cluster representatives instead of all database templates [21,22]. This work proposes an efficient fingerprint search algorithm based on database clustering. The data-flow chart of this algorithm is shown in Fig. 1. It differs from the continuous classification (full search) in that clustering is employed to exploit the similarities among the database templates. Fingerprint is non-uniformly partitioned by a circular tessellation to compute a multi-scale orientation field as the main feature for the search. The average ridge distance (ARD) is extracted as an auxiliary search feature. Our proposed fingerprint search algorithm consists of two phases: offline database clustering and online query processing. During the offline database clustering, a modified form of K-means clustering is proposed to partition the orientation feature space into clusters and fingerprints of each cluster are further divided into bins according to their ARDs. Based on the offline database clustering, a hierarchical online query processing is proposed to facilitate an efficient fingerprint search. In cluster search, each query fingerprint is compared with the cluster prototypes to retrieve the close clusters followed by searching the bins of the retrieved clusters. Fingerprint search is finally performed on the retrieved bins to find the templates close to the query fingerprint.

The next section presents the feature extraction, including the computation of multi-scale orientation field and ARD. In Section 3, we present the proposed fingerprint search algorithm based on database clustering. The experimental results and comparisons are presented in Section 4. Finally, the conclusions are arrived in Section 5.

2. Feature extraction

Fingerprint is composed of parallel ridge and valley flows. There are two kinds of features for its representation: global features that describe the flow structure and local features that describe the minute details of ridges. The global features such as singular points and orientation field (Fig. 2) are often used in the coarse level search (classification) algorithms [4,6,7,16,17], while the local features such as minutiae are employed in most fine matching algorithms [2,13,15]. To avoid redundant representation, the coarse level search features should not be strongly correlated with those for fine matching. The orientation field describes the global ridge flow pattern and the ARD is an intrinsic property of the ridge. These two global features have little correlation and hence are employed for our fingerprint search.

Most fingerprint images consist of the foreground pixels originated from the contact of fingertip with the sensor and the background pixels, i.e., the blank or heavy noisy area. To avoid inclusion of the corrupted feature elements in the background, segmentation is employed to separate the fingerprint foreground pixels from the background pixels for feature selection.

2.1. Orientation vector construction

To construct a compact orientation vector for efficient fingerprint search, the orientation field computation is based on blocks. The local block orientation is estimated by an orientation operator [23] and a two-step orientation smoothing framework [24]. Translation and rotation are needed to bring two different imprints into alignment. A reference point defined in Ref. [25] is detected for the translational alignment while a reference direction [26] is used for the rotational alignment. Let $\hat{\theta}_{i,j}$ be the orientation of block (i, j) and θ_r be the reference direction. Due to the periodicity and discontinuity of $\hat{\theta}_{i,j}$ at $\pm\pi/2$ and θ_r at $\pm\pi$, the aligned local orientation is computed as

$$\theta_{i,j} = \begin{cases} \Delta\theta & \text{if } -\pi/2 < \Delta\theta \leqslant \pi/2, \\ \Delta\theta - \pi & \text{if } \Delta\theta > \pi/2, \\ \Delta\theta + \pi & \text{if } \Delta\theta \leqslant -\pi/2, \end{cases} \tag{1}$$

where $\Delta\theta = \hat{\theta}_{i,j} - \theta_r$. Obviously, $-\pi/2 < \theta_{i,j} \leqslant \pi/2$.

The orientation field computed by uniformly dividing fingerprint into blocks has been widely used as the feature for classification [6,16,17]. This uniform spacing orientation field may obscure the discriminatory power of the orientations in the important singular regions. Larger scale is often required for noise attenuation and dimensionality reduction. Non-uniform spacing is proposed to concentrate orientation measurements more densely in the areas more likely containing the singular points [5,27]. This strengthens the feature elements with large discriminatory power without compromising the performance of noise attenuation and the compactness of feature vector.

To evaluate the discriminatory power of each orientation element, we compute the inconsistency of its orientation values over the aligned fingerprints as in Ref. [26]. The first fingerprint instances of the NIST special database 4 (NIST DB4) are used to test the orientation inconsistency. Fig. 3a shows the orientation inconsistency field. We find that the elements of the region below the reference point have more variant orientation patterns than those above. To improve the discriminability of orientation vector, the elements with large discriminatory power can be estimated in finer scale. In addition, the ridge curvature of the inner region around the reference point is usually larger than that of outer region. The orientations of the high curvature area can be estimated in finer scale than those of low curvature area.

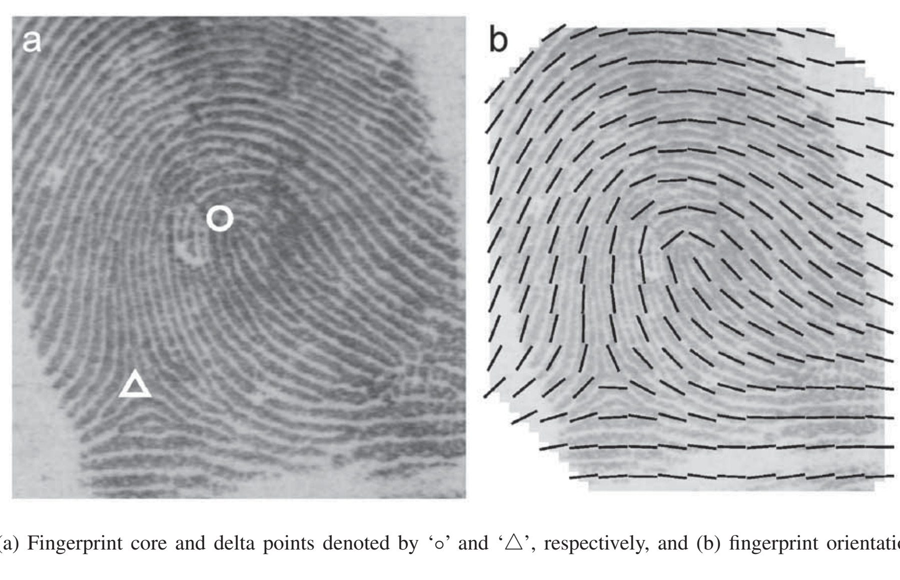

Fig. 2. (a) Fingerprint core and delta points denoted by '∘' and '△', respectively, and (b) fingerprint orientation field.

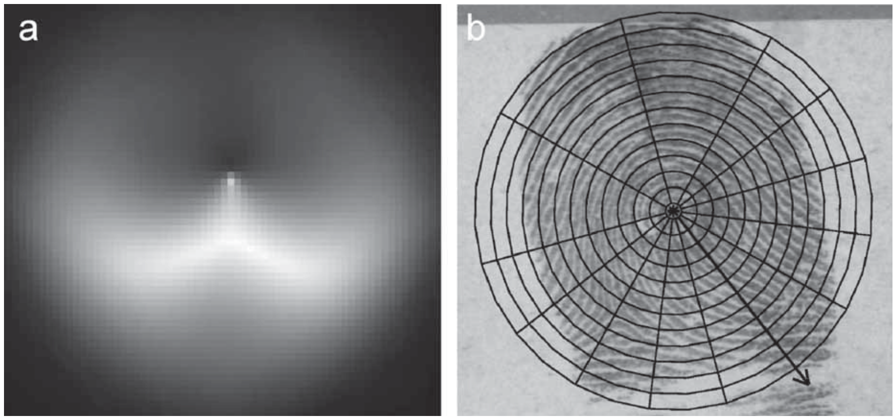

Fig. 3. (a) Fingerprint orientation inconsistency field (large value in white area) and (b) the circular tessellation of fingerprint aligned by the reference point (∗) and direction (the line with arrow).

To compute the multi-scale orientation field, we construct a circular tessellation to non-uniformly divide fingerprint based on above observations. Let $I(x, y)$ be the gray value at pixel (x, y) of a fingerprint of size $X \times Y$ and (x_r, y_r) be the reference point. The circular tessellation is composed of sectors determined by the radius r from (x_r, y_r) and the rotation angle φ from θ_r. The jth sector of the ith band $S_{i,j}$ $(1 \leqslant i \leqslant E, 1 \leqslant j \leqslant F)$ is computed as

$$S_{i,j} = \{(x, y) | (i-1)b + b_0 \leqslant r < i \cdot b + b_0, \varphi_{j-1} \leqslant \varphi < \varphi_j, 1 \leqslant x \leqslant X, 1 \leqslant y \leqslant Y\}, \tag{2}$$

$$r = \sqrt{(x - x_r)^2 + (y - y_r)^2},$$

$$\varphi = \tan^{-1}\left(\frac{y - y_r}{x - x_r}\right) - \theta_r \bmod 2\pi, \tag{3}$$

where b is the band width and b_0 is the width of the innermost band which is not used for orientation extraction due to its large inconsistency. The parameters φ_j, b, E, F are determined empirically to obtain the best performance of fingerprint search. Each band is segmented into 13 non-uniform sectors ($F = 13$) that put finer scale estimation in the region with φ close to θ_r than that far from θ_r. $\varphi_j = j\pi/8$, $(2j-5)\pi/8$ and $(j+3)\pi/8$ for $1 \leqslant j \leqslant 5$, $6 \leqslant j \leqslant 8$ and $9 \leqslant j \leqslant 13$, respectively. Parameter b depends on the dpi resolution of the sensor. It is set to 18 pixels for fingerprints scanned at 500 dpi. Parameter E depends on the size of the contact area of fingertip with sensor. The region around the reference point is partitioned into 12 bands ($E = 12$) for the fingerprints of 512×480 pixels in the NIST DB4. The circular tessellation is shown in Fig. 3b.

An orientation vector $\Theta_q = [\theta_{q,1}, \theta_{q,2}, \ldots, \theta_{q,M}]$ ($M = E \times F$) is constructed for a fingerprint q by concatenating the aligned local orientations of all sectors. The orientation in each sector captures the local ridge flow pattern, while the ordered enumeration of the tessellation describes the global relationships among the local patterns. This orientation vector consists of 156 elements. It covers most important areas of fingerprint if the reference point is located at the center of image. However, only a part of fingerprint is included in the tessellation due to the unfavorable position of reference point so that a substantial number of feature elements are from the noisy background or the outside of image. The segmentation is therefore used to label the valid and invalid feature elements. The segmentation result of fingerprint q is denoted by a vector $S_q = [s_{q,1}, s_{q,2}, \ldots, s_{q,M}]$ where $s_{q,k} \in \{0, 1\}$ and $s_{q,k} = 1$ indicates that sector k is segmented as foreground and valid for feature selection.

2.2. ARD computation

Fingerprint local ridge distance is defined as the distance between the center points of two adjacent ridges. It is another important intrinsic property of fingerprint. For the fingerprints scanned at 500 dpi, the local ridge distance varies from 3 to 25 pixels [28]. Obviously, it is discriminating and has little correlation with the orientation field and minutia. Thus, the ridge distance can be used as an auxiliary feature to bring more information for the fingerprint search. The local ridge distance map $\{\lambda_{x,y}\}$ can be estimated by one of the available approaches [28,29]. However, the local ridge distance of the same finger varies due to the different manners that an elastic finger presses on a plane sensor. Noise and image deterioration may result in large estimation error of ridge distance. The estimated local ridge distance is less stable than the local orientation. It seems unfeasible to construct a high-dimensional feature vector using local ridge distances for the efficient fingerprint search. Nevertheless, the ARD over the fingerprint foreground shows a stable yet discriminating feature. It is a scalar feature invariant of the translation and rotation of fingerprint. Therefore, the 1-D ARD serves as an auxiliary feature for our fingerprint search.

3. Fingerprint search based on database clustering

Fingerprint search on exclusive classification is a fast process. But it cannot sufficiently narrow down the search of database due to the small number of classes. Although the continuous classification can alleviate this problem, the full (exhaustive) fingerprint search is time consuming in the online query process. We take advantages of both approaches by performing the fingerprint search after the cluster retrieval. The 156-D orientation vector and 1-D ARD are used for the fingerprint search. Instead of exclusively classifying fingerprints into a small number of human predefined classes, we employ the clustering technique to partition the database into a number of non-overlapping groups with more flexibility. The clustering based fingerprint search consists of two phases: offline database clustering and online query processing.

3.1. The offline database clustering

The orientation feature space is high-dimensional and unevenly distributed. For example, the orientation fields of whorl fingerprints are more variant than those of plain arch fingerprints as a whorl fingerprint contains more singularities with sharp orientation changes. The clustering technique is applied to partition the orientation feature space into non-overlapping clusters for an efficient fingerprint search. The K-means clustering algorithm [30] is the most widely used partitional clustering algorithm because of its high computational efficiency and low memory space requirement. This clustering algorithm represents each cluster with its mean vector and assign each pattern to the closest cluster iteratively. It terminates when the cluster labels do not change. A modified form of the K-means clustering is developed to group the close orientation vectors into clusters for efficient fingerprint search of database.

It is well known that the K-means clustering needs to specify the number of clusters previously. Instead of understanding the inherent data structure correctly, the main purpose of clustering for fingerprint search is to exploit the similarities among the database templates and facilitate a fast and effective query process. Thus, the initial number of clusters is approximately determined to balance the time efficiency and accuracy of fingerprint search. After grouping N patterns into K clusters, the average number of comparisons is approximately computed as $K + N/K$ for retrieving the nearest cluster followed by fingerprint search in the retrieved cluster. It is minimized when $K = \sqrt{N}$. However, multiple clusters close to the query fingerprint are often retrieved to improve the accuracy of fingerprint search. To balance these effects, the initial number of clusters is set to about $\tau\sqrt{N}$ $(1 < \tau < 3)$.

Euclidean distance measure is often used in the traditional K-means clustering to assign each pattern to the closest cluster. This makes it only effective to discover the hyperspherical clusters. Since our feature vector for clustering is composed of the orientation angles, Euclidean distance by averaging the squared differences cannot be directly applied due to the orientation's periodicity and discontinuity. In addition, if all elements between two orientation vectors consistently have a constant difference, the orientation fields of such two vectors are very similar just with a rotation in human perception. The distance between them is zero but their Euclidean distance can achieve a large deviation. To overcome this problem, we introduce a distance measure by averaging the unit vectors of the doubled difference instead of the squared differences over all valid orientations. The distance between the orientation vectors of fingerprint p and q (i.e., Θ_p and Θ_q) is computed as

$$d_C(\Theta_p, \Theta_q) = 1 - \frac{\left|\sum_{k=1}^{M} v_k e^{j2(\theta_{p,k}-\theta_{q,k})}\right|}{\sum_{k=1}^{M} v_k}, \tag{4}$$

where $v_k = s_{p,k}s_{q,k}$ ($v_k \in \{0, 1\}$ and $v_k = 1$ means that element k is valid for both fingerprint p and q) , $j = \sqrt{-1}$ and $|z|$ computes the magnitude of the complex variable z. This distance measure $d_C(\Theta_p, \Theta_q)$ ($\in [0\,1]$) quantifies the dissimilarity of Θ_p and Θ_q based on the inconsistency of the orientation differences $\theta_{p,k} - \theta_{q,k}$ among all valid elements. It reaches the minimum of zero when all orientation differences are same and increases with the increase of the variation of the orientation differences. Let a and c be two scalars and I be a unit vector of the same size as Θ. It is easy to verify that the distance measure (4) satisfies

$$d_C(\Theta_p + aI, \Theta_q + cI) = d_C(\Theta_p, \Theta_q), \tag{5}$$

because

$$\left|\sum_{k=1}^{M} v_k e^{j2(\theta_{p,k}+a-\theta_{q,k}-c)}\right| = |e^{j2(a-c)}| \left|\sum_{k=1}^{M} v_k e^{j2(\theta_{p,k}-\theta_{q,k})}\right| = \left|\sum_{k=1}^{M} v_k e^{j2(\theta_{p,k}-\theta_{q,k})}\right|.$$

This indicates that the distance measure (4) is invariant of constant amount of orientation differences caused by a slight rotation between two aligned fingerprints. It is employed to compute the distance between two orientation vectors in this work. The modified K-means clustering employs this distance measure to assign each orientation vector to the cluster with the closest prototype.

After all patterns are assigned to their closest clusters in each iteration, the mean vector of each cluster is computed as its new prototype in the traditional K-means clustering. However, the mean vector by directly averaging the orientation vectors is not applicable due to the periodicity. For example, the average orientation of 0 and π is 0 instead of the arithmetic mean value $\pi/2$. To avoid this problem, the orientation averaging is often performed by separately averaging two components of the unit vector of doubled angles

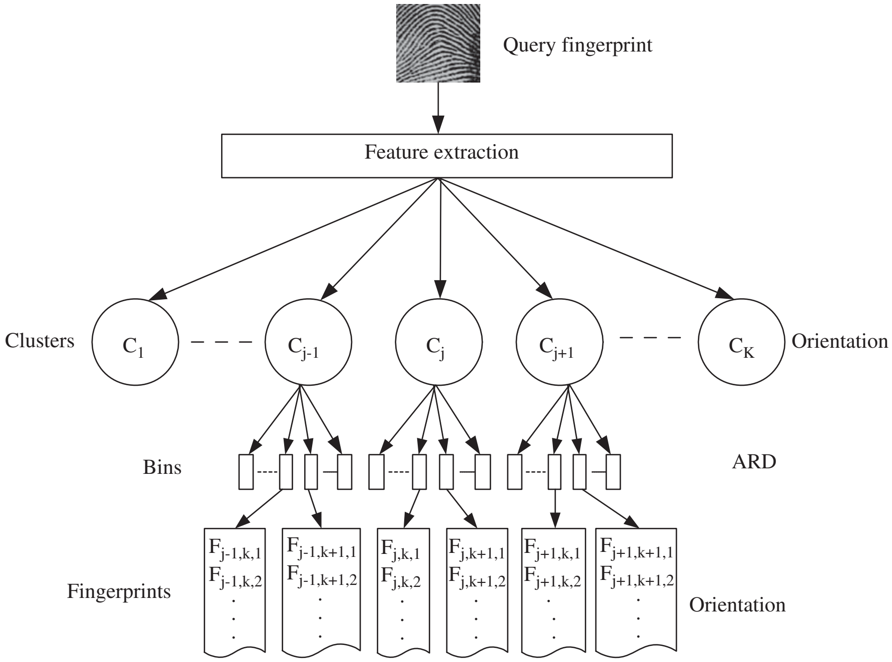

Fig. 4. The hierarchical online query processing by incorporating two features.

$[\cos 2\theta, \sin 2\theta]$. This method is used to compute the mean vector of each cluster. Let $\{Z_1, Z_2, \ldots, Z_K\}$ denote the K cluster prototypes. In the modified K-means clustering, the prototype Z_l of cluster C_l is updated as

$$Z_l = \frac{1}{2} \arctan \frac{\sum_{p \in C_l} S_p \sin 2\Theta_p}{\sum_{p \in C_l} S_p \cos 2\Theta_p}. \tag{6}$$

The K-means clustering cannot guarantee the global minimum of the cluster criterion. In this application, the skewed fingerprint distribution on the clusters may deteriorate the effectiveness of fingerprint search. Let p_i $(1 \leqslant i \leqslant K)$ be the portion of fingerprints assigned to cluster C_i. The average portion of fingerprints retrieved is about $\sum_{i=1}^{K} p_i^2$ if the nearest cluster is retrieved for a query fingerprint. It achieves the minimum on even distribution $(p_1 = p_2, \ldots, = p_K)$ and increases on skewed distribution. To alleviate the above problems, the modified K-means clustering eliminates the small clusters and splits the large clusters into two clusters. To split the large cluster, a new prototype is added by randomly choosing one feature vector from the cluster.

The traditional K-means clustering is modified by replacing the Euclidean distance measure with the distance measure (4), computing the cluster prototype with Eq. (6), eliminating the small clusters and splitting the large clusters. The modified K-means clustering algorithm repeats above procedures and outputs the clusters when the cluster prototypes do not change. The processing steps of this algorithm are summarized as:

(1) initialize the number of clusters K and cluster prototypes;
(2) compute distances between each orientation vector and K prototypes with Eq. (4) and assign the fingerprint to the closest cluster;
(3) compute the new cluster prototypes with Eq. (6);
(4) compute distances between the new and old prototypes with Eq. (4). If the maximum one is larger than ε, go to step (2);
(5) if there are no small or large clusters, output the clusters. Otherwise, eliminate the small clusters, split the large clusters and go to step (2).

After partitioning the orientation feature space into clusters, the 1-D ARD is employed as an auxiliary feature to further divide the fingerprints of each cluster into bins. Over the whole range of ARD, B bins of equal width are predefined in each cluster. The center of the kth bin is computed as $ARD_{\min} + k \times (ARD_{\max} - ARD_{\min})/B$. Since more than three bins close to the query fingerprint are usually retrieved in the fingerprint search, B is set to larger than 20. The fingerprints in each clusters are assigned to the bin with the closest center to its ARD.

3.2. The online query processing

The online query processing is to retrieve a subset of database templates close to the query fingerprint. We propose a hierarchical query process that consists of three levels of search (see Fig. 4). In the first level, we search the clusters by comparing the query orientation vector with the cluster prototypes. Some ambiguous fingerprints are located near the cluster boundary no matter how well the database is partitioned. To alleviate this problem, we retrieve multiple nearest clusters instead of only the nearest one. Similarly, we search the bins of the retrieved clusters by comparing the query ARD with the bin centers and retrieve multiple nearest bins in the second level. These coarse level searches can efficiently narrow down the search of database because the number of groups is much smaller than the number of templates. In the finest level, the fingerprint search is performed on the retrieved bins using the orientation feature to further narrow down the search space. Therefore, the online query processing of fingerprint search is accelerated by database clustering without compromising the effectiveness of fingerprint search.

For the cluster search, we compute the orientation distances (4) $d_C(\Theta_q, Z_l)$ $(1 \leqslant l \leqslant K)$ between the query fingerprint and cluster prototypes and retrieve the clusters with the distances smaller than a threshold. Since the clusters may be unevenly distributed in the orientation feature space, the threshold is adaptively determined as $\min_{l=1}^{K} d_C(\Theta_q, Z_l) + \sigma_1$ where σ_1 is tuned to adjust the retrieval neighborhood. In the retrieved clusters, the distances between the query ARD and bin centers are computed to retrieve the bins with the distances smaller than a threshold. This threshold is set to produce small retrieval error in this coarse level search. It is constantly specified as 1 pixels in our experiments on the NIST DB4.

In the fingerprint search, we compute the orientation distances $d_C(\Theta_q, \Theta_j)$ between query fingerprint and all fingerprints in the retrieved bins. The fingerprints with the distances smaller than a threshold are finally retrieved for the fine matching. Similarly, the retrieval threshold is adaptively determined as $\min_{j=1}^{N_q} d_C(\Theta_q, \Theta_j) + \sigma_2$ where N_q is the number of fingerprints in the retrieved bins and σ_2 is tuned to adjust the portion of retrieved fingerprints.

4. Experimental results and comparisons

Most published results of fingerprint search are based on the NIST special database 4 (NIST DB4). To have a comprehensive comparison, we also test our algorithm on this well-known database, which contains 2000 pairs of fingerprints of size 480×512 pixels. NIST DB4 is collected for testing the exclusive classification so that the five common classes (arch, tented arch, left loop, right loop and whorl) are evenly distributed in the database. However, the natural fingerprint distribution in these five classes is significantly different. To resemble the natural distribution, we reduce the number of fingerprints of less frequent classes and obtain 1204 pairs of fingerprints. The reduced database, called data set 2, and the original NIST DB4 are both applied in our experiments. The first fingerprint instances are used as the database templates while the second instances serve as query fingerprints.

The performance of fingerprint search is evaluated by the penetration rate, retrieval accuracy and search complexity. The penetration rate is the average portion of database re-

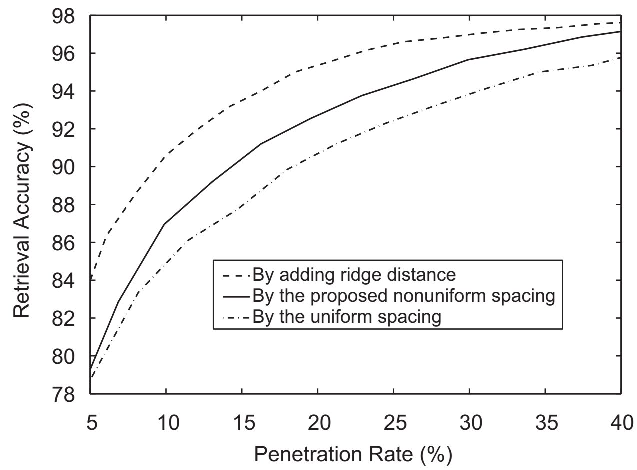

Fig. 5. Results of fingerprint search by the orientation feature computed on the uniform and proposed non-uniform spacing and by adding the average ridge distance.

trieved over all query fingerprints. It indicates how much the fingerprint search can narrow down the database and is controlled by the parameter σ_1 and σ_2 in our proposed approach. For a query fingerprint, the search is successful if one of the retrieved candidates is from the same finger as the query. It is more likely to retrieve the correct one if more templates are retrieved from the database. The retrieval accuracy is thus computed at various different penetration rates. The orientation comparisons of Eq. (4) cost most of the computation in the online query processing. The search complexity is thus evaluated by the average number of such comparisons required over all query fingerprints.

4.1. Experiments on feature extraction

This experiment tests the effectiveness of two features for fingerprint search: a 156-D orientation vector and a 1-D ARD. Full database search based on the orientation vectors constructed by the uniform and proposed non-uniform spacing is applied on the NIST DB4. The uniform spacing orientation field is computed by dividing fingerprint into blocks of 27×27 pixels. An orientation vector consisting of $361(19 \times 19)$ elements is constructed by concatenating the phase angles after the same translational and rotational alignments. To show the improvement of search performance by adding the ARD, the fingerprints whose ARDs are close to that of query fingerprint (their distances are smaller than 1 pixel) are retrieved for the fingerprint search on the non-uniform spacing orientation vectors. Their results of fingerprint search are shown in Fig. 5 where the penetration rate is adapted by varying the parameter σ_2. We can see that the orientation extraction by our proposed non-uniform spacing not only produces more compact feature vector but also achieves better retrieval accuracy than that by the uniform spacing. The 1-D ARD as an auxiliary feature consistently improves the retrieval accuracy. It also reduces about 38%

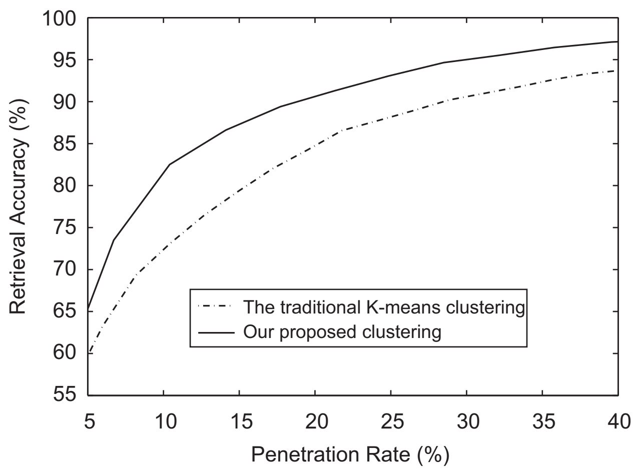

Fig. 6. The results of fingerprint search on the traditional and our proposed modified K-means clustering algorithms.

orientation comparisons in the query process since the average portion of the retrieved fingerprints by ARD is about 62% of the fingerprints in the database.

4.2. Experiments on clustering based fingerprint search

All experiments in this section are performed on the NIST DB4.

4.2.1. Comparison with the traditional K-means clustering

To apply the traditional K-means clustering algorithm, the orientation vector is constructed on the unit vectors $[\cos(2\Theta), \sin(2\Theta)]$ instead of the phase angles and Euclidean distance measure is used to assign each fingerprint to the closest cluster. The number of clusters and the cluster prototypes are initialized same for both clustering techniques. The final numbers of clusters are 90. To better reflect the effectiveness of clustering techniques on the fingerprint search, we present the search results just by retrieving the close clusters produced by clustering the orientation feature space. The 1-D ARD and the following fingerprint search are not used in this experiment. Fig. 6 shows the experimental results where the penetration rate is adapted by varying the parameter σ_1. Our proposed modified K-means clustering outperforms the traditional K-means clustering for the fingerprint search of database.

4.2.2. Comparison with full fingerprint search

In this experiment, the same feature set is used in the fingerprint search procedures with and without clustering. Using the 1-D ARD as an auxiliary feature for full fingerprint search narrows down the search space to about 62% of database. In the query process of our clustering based fingerprint search, the number of orientation comparisons is the number of clusters (90 in our experiments) plus the number of fingerprints in the retrieved bins. It varies at different penetration rates. Fig. 7 shows the results of the full fingerprint search (without clustering) and our proposed fingerprint search (with clustering). From Fig. 7a, we can see the search complexity is greatly reduced by using the clustering, especially at the low penetration rates. For example, to retrieve 5% of database, we require 300 ($N \times 15\%$) orientation comparisons in our clustering based fingerprint search that is much smaller than 1240 ($N \times 62\%$) comparisons in the full fingerprint search. Moreover, the retrieval accuracy is slightly yet consistently improved by the modified K-means clustering (see Fig. 7b). This may be resulted by exploiting the similarities among the database templates through the clustering. The results demonstrate that the proposed clustering based approach not only speeds up the search process but also improves the retrieval accuracy.

4.2.3. Effects of the number of clusters

This experiment is to test the effects of different number of clusters on the performance of fingerprint search. Although it is not necessary to specify the optimal number of clusters for our fingerprint search, the final number of clusters may have some effects on the search results. The number of clusters varies from 20 to 135 in our experiments and the search results are shown in Fig. 8. The retrieval accuracy is improved when increasing the number of cluster from 20 to 90 and cannot be further improved with the number of clusters increased to 135 (see Fig. 8a). From Fig. 8b, we can see that the search complexity is reduced by increasing the number of clusters from 20 to 60 and deteriorates by further increasing it to 90 and 135.

4.3. Comparisons with other approaches

The continuous classification approach [18] was tested on the data set 2 in Ref. [18]. It performs better than the approach [16]. We also implement our search algorithm on data set 2. Fig. 9 shows our results and the results reported in Ref. [18] on the same data set. We can see that consistent performance improvement of our approach is visible at all penetration rates in Fig. 9 and significant performance improvement is achieved at low penetration rates.

A fingerprint search approach based on indexing the triplets of minutia points is proposed in Ref. [12]. The result of fingerprint search is further improved in Ref. [3] by adding two new features. The best performed approach was tested in Ref. [3] on the second 1000 pairs of fingerprints of the NIST DB4. As stated in Ref. [3], the retrieval accuracies on the penetration rates of 5%, 10%, 15% and 20% are 83.3%, 88.1%, 91.1% and 92.6%, respectively. Fig. 10 shows the results reported in Ref. [3] and the experimental results of our proposed approach on the same data set. We can see that our proposed fingerprint search approach outperforms the approach [3] on indexing the minutia triplets.

We also compare our proposed fingerprint search approach with some state-of-the-art approaches of exclusive

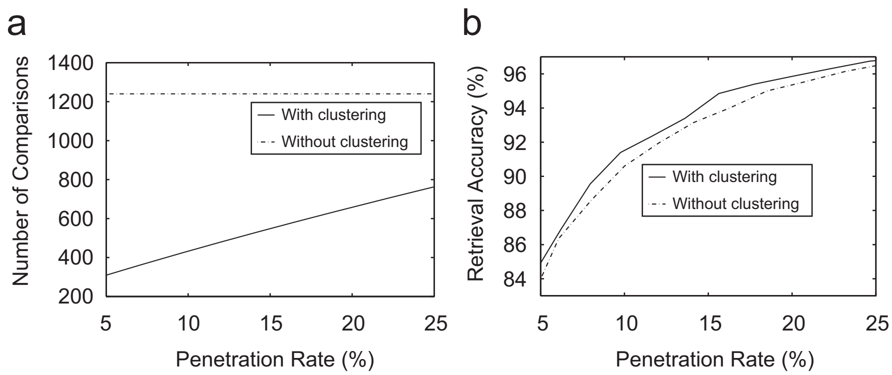

Fig. 7. Results of full fingerprint search (without clustering) and our proposed fingerprint search (with clustering): (a) search complexity and (b) retrieval accuracy.

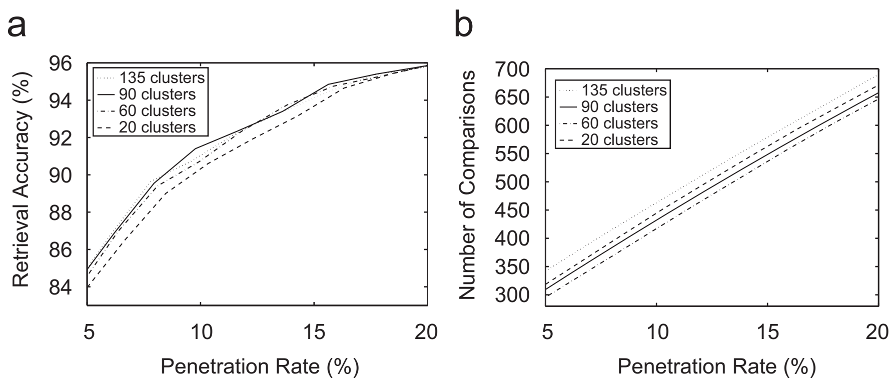

Fig. 8. Results of fingerprint search on the different number of clusters: (a) retrieval accuracy and (b) search complexity.

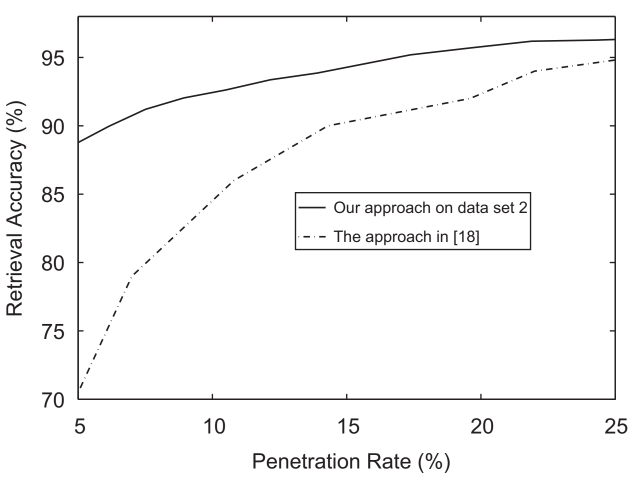

Fig. 9. Results of fingerprint search in our approach and [18] on data set 2.

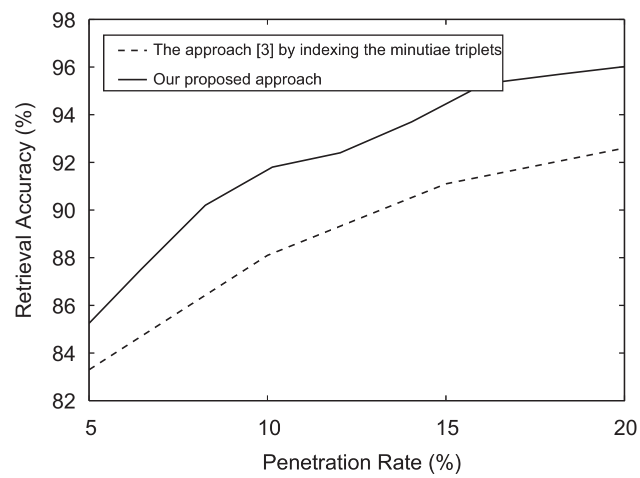

Fig. 10. Results of fingerprint search reported in Ref. [3] and our proposed approach on the same data set.

classification according to the error rate at about the same penetration rate. Most published exclusive classification approaches classify fingerprints into four or five Henry classes. The penetration rate is 20% if a perfect classifier is applied to partition NIST DB4 into five Henry classes which are evenly distributed in the database. It will increase up to 28% if two classes (plain and tented arches) are merged into one. As the fingerprint frequency in each class does not reflect the real distribution, many researchers weight the classification results or construct a subset

Table 1
Error rates of some exclusive classification approaches and our clustering based approach on NIST DB4

Method, year and source	$PR = 20\%$, five classes	$PR = 29.5\%$, five weighted classes	$PR = 28\%$, four classes	$PR = 29.7\%$, four weighted classes	Test set
Candela et al. 95 [6]	—	—	11.4	6.1	Second half
Karu and Jain 96 [4]	14.6	11.9	8.6	9.4	Whole
Jain et al. 99 [5]	10	7.0	5.2	—	Second half
Jain and Minut 02 [31]	—	—	8.8	9.3	Whole
Cappelli et al. 99 [18]	—	12.9	—	—	Data set 2
Cappelli et al. 99 [7]	7.9	6.5	5.5	—	Second half
Senior 01 [10]	—	—	—	5.1	Second half
Yao et al. 01 [9]	10.7	9.0	6.9	—	Second half
Marcialis et al. 01 [11]	12.1	9.6	—	—	Second half
Zhang and Yan 04 [32]	15.7	—	7.3	—	Whole
Park and Park 05 [8]	9.3	—	6.0	—	Whole
Our approach	4.2	2.9	3.1	2.9	Whole

(data set 2) according to the natural distribution for testing. For the weighted classification results or on data set 2, the penetration rates are 29.48% and 29.69% for the Henry four and five classes, respectively. For a fair comparison, our approach retrieves fingerprints at the penetration rates of or slightly smaller than 20%, 28%, 29.48% and 29.69% in the experiments. Table 1 shows the error rates of 11 published exclusive classification approaches and our proposed approach on NIST DB4 (‘whole’) or its second half. All error rates in the same column are at the same penetration rate. The number of classes labelled by the exclusive classification approaches and whether weighting is used in the error calculation are indicated in the second row of Table 1.

It should be noted that the error of exclusive classification is not fully equivalent to the error of fingerprint search for the application to identification. The exclusive classification for fingerprint identification is successful only if the query fingerprint and the corresponding one in the database are consistently classified in the same class. There are about 17% of fingerprints in the NIST DB4 labelled as two classes by human experts. The error rates of approaches [4–7,9,31,32] are calculated by assuming a correct classification if the classifier output is any one of two class hypotheses. This assumption gives lower error rate than that obtained using only one class label. In addition, the error rates of approaches [5,9,11] are obtained at 1.8% rejection rate, which slightly increases their penetration rates. Nevertheless, experimental results in Table 1 demonstrate that the proposed approach achieves lower error rate than various exclusive classification approaches.

5. Conclusions

Techniques that facilitate an efficient and effective search of database in fingerprint identification have been extensively studied in the past decades. The exclusive classification cannot sufficiently narrow down the search of database. The continuous classification by full fingerprint search neglects the similarities among the database templates so that its search performance is limited. This paper proposes an efficient fingerprint search algorithm based on database clustering which performs cluster search before the fingerprint search of retrieved clusters. In orientation extraction, we propose a non-uniform spacing of fingerprint by a circular tessellation to compute a multi-scale orientation field as the main feature for the fingerprint search. The orientation extraction by the non-uniform spacing not only produces more compact feature vector but also achieves better performance of fingerprint search than that by the uniform spacing. The 1-D ARD is employed as an auxiliary search feature which not only reduces the orientation comparisons in the query process but also consistently improves the retrieval accuracy. A modified K-means clustering was proposed to partition the orientation feature space into clusters. It outperforms the traditional K-means clustering for the fingerprint search. Based on the offline database clustering, a hierarchical query processing is proposed to facilitate an efficient fingerprint search. It not only reduces the search complexity but also improves the retrieval accuracy. The extensive experimental studies and comparisons consistently demonstrate the effectiveness and superiority of the proposed fingerprint search framework.

References

[1] D. Maltoni, D. Maio, A.K. Jain, A. Prabhakar, Handbook of Fingerprint Recognition, Springer, New York, 2003.

[2] N.K. Ratha, S. Chen, K. Karu, A. Jain, A real-time matching system for large fingerprint databases, IEEE Trans. Pattern Anal. Mach. Intell. 18 (8) (1996) 799–813.

[3] X. Tan, B. Bhanu, Y. Lin, Fingerprint identification: classification vs. indexing, in: Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS’03), 2003, pp. 151–156.

[4] K. Karu, A.K. Jain, Fingerprint classification, Pattern Recognition 29 (3) (1996) 389–404.
[5] A.K. Jain, S. Prabhakar, L. Hong, A multichannel approach to fingerprint classification, IEEE Trans. Pattern Anal. Mach. Intell. 21 (4) (1999) 348–359.
[6] G.T. Candela, P.J. Grother, C.I. Watson, R.A. Wilkinson, C.L. Wilson, PCASYS—a pattern-level classification automation system for fingerprints, Technique Report: NIST TR 5647, 1995.
[7] R. Cappelli, D. Maio, D. Maltoni, Fingerprint classification based on multi-space kl, in: Proceedings of the Workshop on Automatic Identification Advanced Technologies, 1999, pp. 117–120.
[8] C.H. Park, H. Park, Fingerprint classification using fast Fourier transform and nonlinear discriminant analysis, Pattern Recognition 38 (4) (2005) 495–503.
[9] Y. Yao, P. Frasconi, M. Pontil, Fingerprint classification with combination of support vector machine, in: Proceedings of the Third International Conference on Audio- and Video-based Biometric Person Authentication, 2001, pp. 253–258.
[10] A. Senior, A combination fingerprint classification, IEEE Trans. Pattern Anal. Mach. Intell. 23 (10) (2001) 1165–1174.
[11] G.L. Marcialis, F. Roli, P. Frasconi, Fingerprint classification by combination of flat and structural approaches, in: Proceedings of the Third International Conference on Audio- and Video-based Biometric Person Authentication, 2001, pp. 241–246.
[12] B. Bhanu, X. Tan, Fingerprint indexing based on novel features of minutiae triplets, IEEE Trans. Pattern Anal. Mach. Intell. 25 (5) (2003) 616–622.
[13] X.D. Jiang, W.Y. Yau, W. Ser, Detecting the fingerprint minutiae by adaptive tracing the gray level ridge, Pattern Recognition 34 (5) (2001) 999–1013.
[14] A.K. Jain, L. Hong, S. Pankanti, R. Bolle, An identity-authentication system using fingerprints, Proc. IEEE 85 (9) (1997) 1365–1388.
[15] X.D. Jiang, W. Ser, On-line fingerprint template improvement, IEEE Trans. Pattern Anal. Mach. Intell. 24 (8) (2002) 1121–1126.
[16] A. Lumini, D. Maio, D. Maltoni, Continuous versus exclusive classification for fingerprint retrieval, Pattern Recognition Lett. 18 (10) (1997) 1027–1034.
[17] T. Kamei, M. Mizoguchi, Fingerprint preselection using eigenfeatures, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1998, pp. 918–923.
[18] R. Cappelli, A. Lumini, D. Maio, D. Maltoni, Fingerprint classification by directional image partitioning, IEEE Trans. Pattern Anal. Mach. Intell. 21 (5) (1999) 402–421.
[19] R. Cappelli, D. Maio, D. Maltoni, Combining fingerprint classifiers, in: Proceedings of the First International Workshop on Multiple Classifier Systems, 2000, pp. 351–361.
[20] J.D. Boer, A.M. Bazen, S.H. Gerez, Indexing fingerprint database based on multiple features, in: Proceedings of the ProRISC, 12th Annual Workshop on Circuits, Systems and Signal Processing, 2001.
[21] R. Zhang, Z. Zhang, A clustering based approach to efficient image retrieval, in: Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence, 2002.
[22] K.-M. Lee, W.N. Street, Cluster-driven refinement for content-based digital image retrieval, IEEE Trans. Multimedia 6 (6) (2004) 817–827.
[23] X.D. Jiang, Extracting image orientation feature by using integration operator, Pattern Recognition 40 (2) (2007) 705–717.
[24] X.D. Jiang, On orientation and anisotropy estimation for online fingerprint authentication, IEEE Trans. Signal Process. 53 (10) (2005) 4038–4049.
[25] X.D. Jiang, M. Liu, A.C. Kot, Reference point detection for fingerprint recognition, in: 17th International Conference on Pattern Recognition, 2004.
[26] M. Liu, X.D. Jiang, A.C. Kot, Fingerprint reference point detection, EURASIP J. Appl. Signal Process. 2005 (4) (2005) 498–509.
[27] C.L. Wilson, G.T. Candela, C.I. Watson, Neural network fingerprint classification, J. Artif. Neural Networks 1 (2) (1994) 203–228.
[28] L. Hong, Y. Wan, A.K. Jain, Fingerprint image enhancement: algorithm and performance evaluation, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 777–789.
[29] X.D. Jiang, Fingerprint image ridge frequency estimation by higher order spectrum, in: Proceedings of the International Conference on Image Processing, vol. 1, 2000, pp. 462–465.
[30] J. McQueen, Some methods for classification and analysis of multivariate observations, in: Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, 1967, pp. 281–297.
[31] A.K. Jain, S. Minut, Hierarchical kernel fitting for fingerprint classification and alignment, in: Proceedings of the 16th International Conference on Pattern Recognition, vol. 2, 2002, pp. 469–473.
[32] Q. Zhang, H. Yan, Fingerprint classification based on extraction and analysis of singularities and pseudo ridges, Pattern Recognition 37 (11) (2004) 2233–2243.

About the Author—MANHUA LIU received B.Eng. degree in 1997 and M.Eng. degree in 2002 in automatic control from North China Institute of Technology and Shanghai JiaoTong University, China, respectively. Since 2002, she has been a Ph.D. candidate at Nanyang Technological University (NTU), Singapore. Her research interests include biometrics, pattern Recognition, image processing and machine learning.

About the Author—XUDONG JIANG received the B.Eng. and M.Eng. degrees from the University of Electronic Science and Technology of China (UESTC), in 1983 and 1986, respectively, and the Ph.D. degree from the University of German Federal Armed Forces, Hamburg, Germany, in 1997, all in electrical and electronic engineering. From 1986 to 1993, he was a Lecturer at UESTC, where he received two Science and Technology Awards from the Ministry for Electronic Industry of China. From 1993 to 1997, he was with the University of German Federal Armed Forces as scientific assistant. From 1998 to 2002, he was with Nanyang Technological University, Singapore, as a Senior Research Fellow, where he developed a fingerprint verification algorithm that achieved the most efficient and the second most accurate fingerprint verification in the International Fingerprint Verification Competition (FVC2000). From 2002 to 2004, he was a Lead Scientist and Head of the Biometrics Laboratory at the Institute for Infocomm Research, Singapore. Currently, he is a faculty member and serves as the Director of the Centre for Information Security, the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interest includes pattern recognition, signal and image processing, computer vision and biometrics.

About the Author—ALEX KOT was educated at the University of Rochester, New York, and at the University of Rhode Island, USA, where he received the Ph.D. degree in electrical engineering in 1989. He was with the AT&T Bell Company, New York, USA. Since 1991, he has been with the Nanyang Technological University (NTU), Singapore, where he is the Vice Chair of the School of EEE. His research and teaching interests are in the areas of signal processing, watermarking, and information security. Dr. Kot served as the General Co-Chair for the Second International Conference on Information, Communications and Signal Processing (ICICS) in December 1999, the Advisor for ICICS'01 and ICONIP'02. He received the NTU Best Teacher of the Year Award in 1996 and has served as the Chairman of the IEEE Signal Processing Chapter in Singapore. He is the General Co-Chair for the IEEE ICIP 2004 and served as Associate Editor for the IEEE Transactions on Signal Processing and the IEEE Transactions on Circuits and Systems for Video Technology.