Pattern Recognition 43 (2010) 1062 – 1070

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Fingerprint classification based on Adaboost learning from singularity features

Manhua Liu*

Department of Instrument Science and Engineering, School of EIEE, Shanghai Jiao Tong University, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 2 June 2009
Received in revised form 5 August 2009
Accepted 15 August 2009

Keywords:
Biometrics
Fingerprint classification
Fingerprint singularities
Decision tree classifier
Adaboost learning

ABSTRACT

Fingerprint classification is an important indexing scheme to narrow down the search of fingerprint database for efficient large-scale identification. It is still a challenging problem due to the intrinsic class ambiguity and the difficulty for poor quality fingerprints. In this paper, we presents a fingerprint classification algorithm that uses Adaboost learning method to model multiple types of singularity features. Firstly, complex filters are used to detect the singularities. For powerful representation, we compute the complex filter responses of the detected singularities at multiple scales and a feature vector is constructed for each scale that consists of the relative position and direction and the certainties of the singularities. Adaboost learning method is then applied on decision trees to design a classifier for fingerprint classification. Finally, fingerprint class is determined by the ensemble of the classification results at multiple scales. The experimental results and comparisons on NIST-4 database have shown the effectiveness and superiority of the fingerprint classification algorithm.

1. Introduction

Fingerprint as a kind of human biometrics on the finger tips has been widely used for personal recognition in forensic and civilian applications because of its uniqueness, immutability and low cost. Generally, fingerprint based recognition systems work in two modes: verification and identification [1]. In the verification mode, the user inputs his fingerprint and claims an identity, then the system verifies whether the input fingerprint is consistent with the claimed identity. In the identification mode, however, the user inputs his fingerprint and the system identifies the potential corresponding ones in the database without a claimed identity. Therefore, automatic fingerprint identification requires the input fingerprint be compared with all fingerprints in the database for a match, which is more complex than the verification mode especially for large database. Although satisfactory performances have been reported for fingerprint verification, both the time efficiency and accuracy deteriorate seriously by simple extension of a 1:1 verification procedure to a 1:N identification system [1]. A common strategy is to organize the fingerprints in a database into a number of groups that have similar properties before the fine matching [2,3]. An input fingerprint is first compared with the groups at a coarse level and then matched to a subset of database corresponding to one or more groups similar to that fingerprint at a fine level. Fingerprint classification classifies fingerprints into one of the pre-specified classes in an accurate and consistent manner. It is an important indexing scheme to speed up the search of fingerprint database in large-scale identification system. Fingerprint classification has been studied for more than one century. Most of the classification algorithms are based on the Galton–Henry classification scheme [4] to classify fingerprints into five common classes: Arch, Tented Arch, Left Loop, Right Loop and Whorl as shown in Fig. 1. The five common classes are human-interpretable. Consistent and reliable fingerprint classification is still a challenging problem due to the intrinsic class ambiguity and the difficulty for poor quality fingerprints. In this work, we also classify fingerprints into the five common classes.

Fingerprint classification has generated great interests due to its importance and intrinsic difficulty and there are a number of approaches proposed in the literature [5–13]. A typical fingerprint classification algorithm usually extracts a representative feature set to capture the individuality of each fingerprint and then does some strategies to determine the fingerprint class. Fingerprint is composed of interleaved ridge and valley flows. In general, there are two main types of features for fingerprint representation: coarse level features which globally describe the ridge-valley flow patterns (e.g., singularities) and fine level features with the minute details of ridge flow anomalies (e.g., minutia points) as shown in Fig. 2. Singularities can also be considered as the anomalies of ridge flow structure. There are two types of singularities i.e., core and delta points. To facilitate the database search, a fingerprint is often classified based on the coarse level features while the fine level features are used in the

* Tel.: +86-21-548 10988.
E-mail addresses: lmanhua@msn.com, mhliu@sjtu.edu.cn (M. Liu).

0031-3203/$ - see front matter
doi:10.1016/j.patcog.2009.08.011

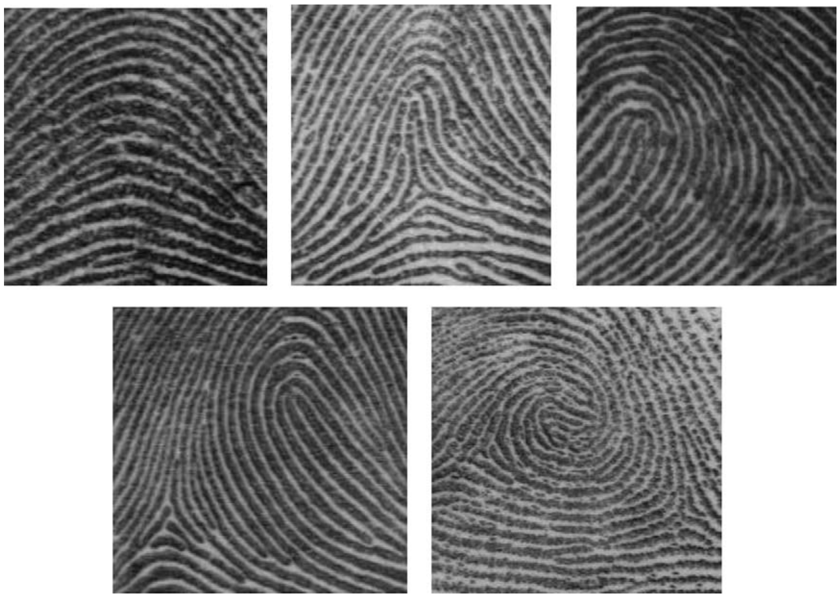

Fig. 1. Five common fingerprint classes: Arch, Tented Arch, Left Loop, Right Loop and Whorl from left to right and top to down.

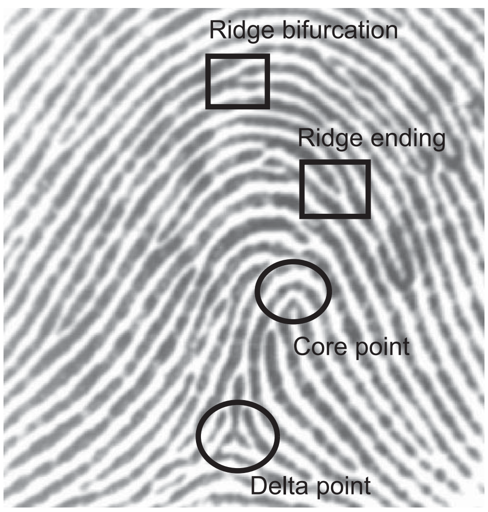

Fig. 2. Singularities (black circles) and minutia points (black boxes).

fine matching. Most of published fingerprint classification approaches use one or more of the following features: singularities, orientation field, ridge flow and Gabor filter responses. The strategies such as rule-based, syntactic, structural, learning-based and hybrid methods are often employed to make fingerprint classification.

Since the Galton–Henry five classes can be distinguished according to the number and relative position of singularities, several researchers have proposed to extract this information and generate some heuristic rules for fingerprint classification [5,14]. However, the singularities detection is sensitive to noise which results in the difficulty of consistent and reliable classification. The traced pseudo-ridges and orientation field are used as a supplement to compensate for the missing singularities and improve the classification performance [14,15]. A rule-based approach is proposed to define a fingerprint kernel to model the general geometrical shape of the fingerprints in each class and fingerprint classification is conducted by hierarchical kernel fitting [16]. Since the ridge flows contain more information than the singularities, a syntactic classification approach is proposed based on the global distribution of 10 basic ridge patterns, the analysis of the ridge shapes and a sequence of ridge distributions [17]. The orientation field composed of the local ridge orientations represents well the ridge flow structure of fingerprint. The structural classification approaches have been proposed to partition the fingerprint orientation field into "homogeneous" orientation regions and the relational graphs of these regions are used to make fingerprint classification [18,19]. The classification approaches based on global structure can work well on partial and poor quality fingerprints.

In addition, many approaches have been proposed to use the coarse-level features via some learning techniques for fingerprint classification [7,8,20,21]. The Karhunen–Loeve (KL) and multi-space generalization of KL (MKL) transforms are used to reduce the dimensionality of orientation vectors for fingerprint classification [7,8]. The neural network classifier as a powerful learning technique is widely used to model some feature sets and determine the fingerprint class. The probabilistic and self-organizing neural networks are used to model and discriminate the fingerprint classes [7,20,21]. A multi-layer artificial neural network composed of six independent sub-networks is constructed to discriminate six fingerprint classes (with each subnetwork for one class) [22]. Support vector machines are also used to perform fingerprint classification [10,15]. Furthermore, different learning techniques are integrated to use various feature sets and improve the performance of classification by a single classifier [6,11,23,24]. A hybrid multi-channel classifier is designed to take the FingerCode feature and apply the K-nearest neighbor classifier followed by a set of neural networks classifier to perform fingerprint classification [6]. A hybrid classifier is proposed to combine hidden Markov model and decision tree for fingerprint classification [11]. Support vector machines and recursive neural networks are combined to discriminate fingerprint classes [23]. Recently, a hybrid fingerprint classification approach is proposed to integrate naive Bayes classifier and support vector machines that use different feature sets [24]. When several types of feature sets and leaning methods are integrated, fingerprint classification tends to be more reliable than that using single feature set and single classifier especially for poor quality fingerprints.

In this paper, we propose a fingerprint classification approach based on Adaboost learning from the feature sets of singularities. Two complex filters are used to detect the singularities firstly. We compute the complex filter responses of the detected singularities at multiple scales and a feature vector is constructed for each scale that includes not only the relative positions and directions of the singularities but also their certainties, i.e., the magnitude of filter response. To model the multiple types of singularity features, the classifier is designed by using Adaboosted decision trees to make fingerprint classification. The classification results on multi-scale singularities are combined to determine fingerprint class. Our fingerprint classification approach consists of the following major steps:

1. Compute the orientation field.
2. Extract the singularities information and construct feature vectors at multiple scales.
3. Design classifiers using Adaboosted decision trees and perform fingerprint classification.

The remainder of this paper is organized as follows. Section 2 presents the feature extraction. In Section 3, we present the fingerprint classifier based on Adaboost learning method. The experimental results and comparisons are presented in Section 4. Finally, conclusions are given in Section 5.

2. Feature extraction

The class of a fingerprint is often determined by its global ridge and furrow structure. A valid feature set for fingerprint classification should be able to capture this global information effectively. The singularities, i.e., core and delta points, are two landmarks of fingerprint with sharp changes of ridge orientations (see Fig. 3). The numbers and positions of core and delta points are closely related to the five fingerprint classes. For example, whorl fingerprint often contains two pairs of core and delta points, loop fingerprint usually has one pair of core and delta points while no singularities exist in plain arch fingerprint. In addition, the features of singularities are invariant to the translation, rotation, and moderate amounts of scale changes in fingerprint image. In this work, we construct numerical feature vectors of fixed size from the singularities for fingerprint classification. Firstly, we estimate the local ridge orientation and segment the foreground from background for each fingerprint. Secondly, two complex filters are applied to the orientation field of foreground for detection of singularities. Finally, instead of using the rigid number and positions of singularities, we construct numerical feature vectors of fixed size based on multi-scale singularities information for representation.

2.1. Orientation estimation and segmentation

Estimation of local ridge orientation is important for the singularities detection. To reduce the computation cost, fingerprint is divided into blocks of the same size and the orientation of each block is estimated. There are many approaches proposed for fingerprint orientation estimation which can be broadly classified as the gradient-based method [25–27] and the pixel-alignment-based method [5,21]. We use the gradient-based method to estimate the local ridge orientation. The orientation field may still contain some corrupted elements resulted by heavy noise such as scars, ridge breaks, and low gray value contrast. Orientation smoothing is expected to further correct the corrupted orientations. The smoothing method proposed in [28], which can attenuate the noise while avoiding the orientation blurring, is applied for smoothing of the orientation field.

Most fingerprint images consist of not only the foreground pixels originated from the contact of fingertip with the sensor but also the background, i.e., the blank or heavy noisy area. A reliable feature extraction should exclude the background and be performed only on the foreground of fingerprint. The fingerprint segmentation approach [29] is used to separate fingerprint foreground from the background for reliable feature extraction in this work.

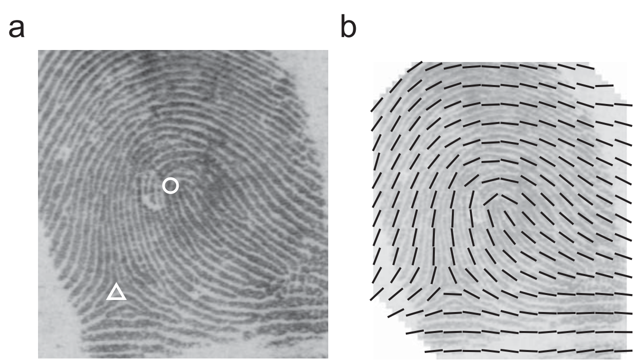

Fig. 3. (a) Core and delta points denoted by ∘ and Δ, respectively, and (b) orientation field of fingerprint.

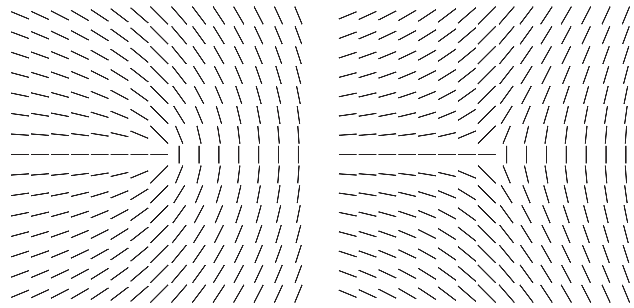

Fig. 4. The local orientation patterns of $k\varphi/2$ with the first order symmetry: $k=1$ (left) and $k=-1$ (right).

2.2. Singularities detection by complex filters

One popular method for detection of fingerprint singularities (i.e., core and delta points) is the Poincare Index method. However, this method can only provide the type and position of singularities. A set of complex filters have been proposed for detection of the patterns with radial symmetries [30]. Complex filters of the first order is used to detect core and delta points of fingerprint [31]. This method not only provides the type and positions of singularities, but also computes singularity directions and certainties (i.e., a continuous measurement on how similar the detected singularity is to the referred symmetrical pattern). In [32], complex filters of the first order are applied on the complex-valued orientation field and a feature vector of 338 dimensions is constructed by concatenating the magnitudes of two filter responses for fingerprint retrieval based on Euclidean distance measure. Similarly, we apply the complex filters on the complex-valued orientation field in this work. However, instead of constructing a high-dimensional feature vector, we just detect the position, direction and certainties of singularities based on the filter responses from which a 16-dimensional feature vector is constructed for fingerprint classification.

The complex filter of order k was modelled as $e^{jk\varphi}$ where k is an integer, j is the imaginary unit and φ is the phase angle of the patterns. A polynomial approximation of these filters is computed as $(x+jy)^k$ where (x,y) are the coordinates. The filters of first order symmetry are computed as $h_{\pm 1}=(x\pm jy)=re^{j\varphi_{x,y}}$ where $\varphi_{x,y}=\angle x\pm jy$. The orientation patterns of $k\varphi/2$ with $k=\pm 1$ are shown in Fig. 4. We can see that the orientation patterns are very close to those of core and delta points in a fingerprint. To detect the singularities, the complex filters of the first order symmetry ($k=\pm 1$) are applied on the complex-valued orientation field of fingerprint.

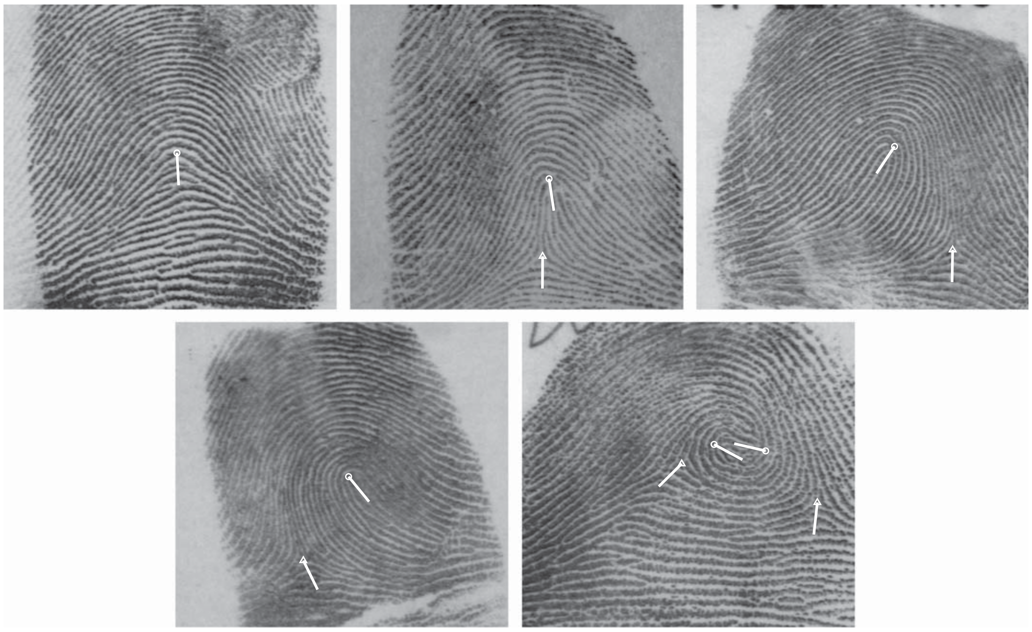

Fig. 5. The position and direction of the detected singularities for sample Arch, Tented Arch, Left Loop, Right Loop and Whorl fingerprints from left to right and top to down (core and delta points are denoted by ∘ and Δ and direction is denoted by a white line, respectively).

Let $\theta_{m,n}$ be the local ridge orientation of block (m,n). To facilitate the complex filtering, the local ridge orientation is represented by a complex value, i.e., $z(m,n)=\cos(2\theta_{m,n})+j\sin(2\theta_{m,n})$. The 2D scalar product $\langle h,z\rangle$, which is the 2D complex convolution between the orientation field z and the complex filter $h_{\pm1}$, is calculated for decomposing fingerprint orientation field into two special symmetrical patterns. For the filtering window of size $(2s+1)\times(2s+1)$, the complex filter responses of block (m,n) are computed as

$$r(m,n)=\frac{\sum_{x=-s}^{s}\sum_{y=-s}^{s}e^{j2\theta_{m+x,n+y}}\cdot e^{-jk\varphi_{x,y}}}{(2s+1)\times(2s+1)} \tag{1}$$

where s is also called as the filtering scale. The complex filter response is $r=ue^{j\alpha}$, where the magnitude $u\in[0,1]$ is a certainty measurement of the symmetrical pattern in fingerprint orientation field, and the phase angle α evaluates the direction of singularity with respect to the referred symmetrical pattern. If the orientation pattern of singularity is exactly a rotated version of the symmetrical pattern described by the complex filter, the magnitude u achieves its maximum value 1 and the phase angle α indicates the corresponding rotation with respect to the reference symmetry. Therefore, the core and delta points of fingerprint produce the local maximum magnitudes of the complex filters with $k=1$ and -1, respectively. Thus, we locate the points with local maximum magnitudes of filter responses as the candidate singularities. To reduce the spurious singularities, the local maximum magnitudes should be larger than a threshold and the multi-scale approach is used to detect the singularities. Initially, the positions with local maximum magnitudes are located as the singularities at the coarse scale, which can eliminate some spurious singularities. To get more precise position of singularities, a new search is done for the local maximum magnitudes in the finer scales. The fine scale search is done in the window computed in the previous coarser scale. However, for plain arch fingerprint, the magnitudes of filter responses are always smaller than the threshold so that no singularities are detected. In this case, instead of no singularities detection, we also search the position with the maximum response magnitudes of the complex filter with $k=1$ as a singularity at multiple scales. Fig. 5 shows the detected singularities of sample images for five fingerprint classes. The maximum number of core and delta points is 2. Finally, we can get three types of information about the singularities for a fingerprint, i.e., position, direction with respect to the reference symmetry and the continuous certainty measurement of the symmetrical pattern (i.e., the magnitude of filter responses). All these singularities information will be used for construction of feature vector.

2.3. Feature vector construction

It is well known that a fingerprint can be classified according to the singularities information. In this work, we try to extract all information from the detected singularities that might be useful for fingerprint classification and allow our classifier to automatically decide which to use and how to use them. In addition to the number and positions of singularities, their directions can provide the relative position of singularities and have discriminative power for fingerprint classes to some extent. For example, the direction of core point is often left for Left Loop fingerprint, is right for Right Loop fingerprint, and is almost down to delta point for Tented Arch fingerprint. Furthermore, the magnitude of filter responses provides a continuous certainty measurement on the symmetrical pattern and also have some discriminative power to discriminate fingerprint classes. For example, plain arch fingerprint has no core and delta points in strict sense so that the certainty of detected singularities is usually smaller than those of other fingerprint classes. Let (x_c^i,y_c^i,α_c^i), $i=1,2$ denote the coordinates and direction of the detected core points and let (x_d^i,y_d^i,α_d^i), $i=1,2$ denote the coordinates and direction of the detected delta points (see Fig. 6).

To facilitate fingerprint classification, a feature vector of fixed size is constructed from the detected singularities. The constructed feature vector includes all above singularities information: position, direction and certainty measurement. Since the numbers of core and delta points are between 0 and 2, we denote the detected core points as $core_1$ and $core_2$ and delta points as $delta_1$ and $delta_2$. The feature vector is defined as $F=[core_1,core_2,delta_1,delta_2]$, where:

- $core_1=(x_c^1,y_c^1,\alpha_c^1,h_c^1)$ includes the information of primary core point. If two core points are detected for a fingerprint, the upper core is selected as primary core. (x_c^1,y_c^1), α_c^1 and h_c^1 are the position, direction and certainty of primary core, respectively.

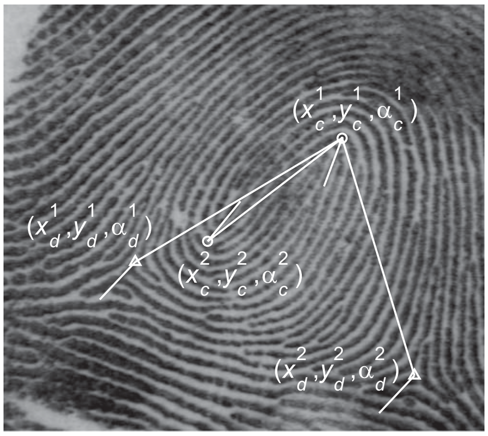

Fig. 6. The coordinates and direction of the detected core and delta points for a whorl fingerprint.

- $core_2 = (r_c, \Delta\beta_c, \Delta\alpha_c, h_c^2)$ is concerned with the second core point. $r_c = \sqrt{(x_c^1 - x_c^2)^2 + (y_c^1 - y_c^2)^2}$, i.e., the distance between two core points. $\Delta\beta_c$ is the difference between the direction of line segment from $core_1$ to $core_2$ and the direction of primary core α_c^1. $\Delta\alpha_c = \alpha_c^2 - \alpha_c^1$ is the direction difference between $core_1$ and $core_2$. h_c^2 is the certainty of the second core.
- $delta_1 = (r_d^1, \Delta\beta_d^1, \Delta\alpha_d^1, h_d^1)$ is related to the primary delta point. If two delta points are detected for a fingerprint, the one closest to the primary core is selected as the primary delta point. $r_d^1 = \sqrt{(x_c^1 - x_d^1)^2 + (y_c^1 - y_d^1)^2}$, i.e., the distance between the primary core and the primary delta points. $\Delta\beta_d^1$ is the difference between the direction of line segment from $core_1$ to $delta_1$ and the direction of primary core α_c^1. $\Delta\alpha_d^1 = \alpha_d^1 - \alpha_c^1$ is the direction difference between the primary core and primary delta. h_d^1 is the certainty of primary delta point.
- $delta_2 = (r_d^2, \Delta\beta_d^2, \Delta\alpha_d^2, h_d^2)$ is concerned with the second delta point. $r_d^2 = \sqrt{(x_c^1 - x_d^2)^2 + (y_c^1 - y_d^2)^2}$, i.e., the distance between the primary core and the second delta point. $\Delta\beta_d^2$ is the difference between the direction of line segment from $core_1$ to $delta_2$ and the direction of primary core α_c^1. $\Delta\alpha_d^1 = \alpha_d^2 - \alpha_c^1$ is the direction difference between primary core and the second delta point. h_d^2 is the certainty of second delta point.

To avoid ambiguity, the computation of direction difference $\Delta\alpha$ is modified as $\Delta\alpha = \Delta\alpha - 2\pi$ if $\Delta\alpha > 2\pi$ or $\Delta\alpha = \Delta\alpha + 2\pi$ if $\Delta\alpha < -2\pi$. The dimensionality of this feature vector is 16 in total. The missing singularity is denoted by $\mathbf{0} = [0, 0, 0, 0]$. If there are one core and one delta point detected, $F = [core_1, \mathbf{0}, delta_1, \mathbf{0}]$. If only one core point is detected, $F = [core_1, \mathbf{0}, \mathbf{0}, \mathbf{0}]$. The feature vector can not only indicate the type and number of detected singularities, but also evaluate their relative positions, relative directions and certainty measurements.

We have used multi-scale method to search the local maximum magnitude of filter responses for robust and reliable detection of singularities. The direction and certainty of the detected singularities can be easily obtained in the finest scale. The complex filter responses of the fine scale can capture the detailed information but they may be sensitive to noise. The complex filter responses of the coarse scale are robust to noise but they may blur the detailed information. To balance these effects, we compute the complex filter responses at multiple scales for each detected singularity and a feature vector is constructed for each scale. In our implementation, we use four scales (i.e., set $s = 1, 2, 3, 4$ in Eq. (1)) to compute the filter responses of the singularities and then construct four feature vectors of 16 dimensions for each fingerprint.

3. Classification of fingerprints

For fingerprint classification, we apply a probabilistic binary classifier based on Adaboost learning method which combines three types of singularities information to compute the class probability. To give a powerful representation, we have extracted as many useful information of the detected singularities as possible such as relative position, relative direction and certainty in feature extraction. But it is difficult to determine which features are discriminative and important for fingerprint classification. A decision tree is a sequence of binary splits of data. Several popular criteria such as *Gini* diversity index are proposed to determine the best variable and best place on which to split a node in learning decision tree. Decision tree classifier is powerful for automatic feature selection. But one decision tree is unstable and only provides limited modelling of the joint statistics of data. This problem can be remedied by the use of boosting, which is a powerful learning technique to combine many weak classifiers to build a strong classifier [33]. Boosting is a greedy procedure that cannot go back to adjust weights of weak learner but produce multiple copies of the weak learner with different weights. For boosting, the training events misclassified will be given a larger weight to build a new weak learner and this procedure is repeated. At the end of training, all weak learners are combined with different weights to make decision. Adaboost is one of the popular boosting methods with good performance. Decision tree is a good weak learner. In this work, we use Adaboosted decision trees for classification of fingerprints, since they work well for automatic feature selection and are efficient to train and apply.

Assume that $T_1, T_2, \ldots, T_M$ are the real-valued weak hypotheses of M decision trees. A set of training data and the corresponding class labels are denoted as $x_1, x_2, \ldots, x_N$ and $y_1, y_2, \ldots, y_N \in \{-1, 1\}$, respectively. The logistic regression version of Adaboost [34] is applied to design the classifiers for fingerprint classification. Let $w_{1,1}, w_{1,2}, \ldots, w_{1,N}$ be the initial weights of N training data and K be the number of nodes for each decision tree. The training steps of the Adaboosted decision trees are given as follows:

1. Initialize $t = 1$.
2. Learn decision tree T_t of K nodes based on the weighted distribution $\mathbf{w_t} = [w_{t,1}, w_{t,2}, \ldots, w_{t,N}]$ using *Gini* split criterion [33].
3. Assign the weighted log-likelihood ratio to each node $T_{t,k}$ for $k = 1, 2, \ldots, K$:

$$f_{t,k} = \frac{1}{2} \log \frac{\sum_{i:y_i=1, x_i \in T_{t,k}} w_{t,i}}{\sum_{i:y_i=-1, x_i \in T_{t,k}} w_{t,i}} \tag{2}$$

4. Update the weights of training data as

$$w_{t+1,i} = w_{t,i} \frac{1}{1 + \exp\left(y_i \sum_{j=1}^{t} f_{j,k_j}\right)}$$

   with k_j such that $x_i \in T_{j,k_j}$ (i.e., x_i is in the node k_j of tree T_j).
5. Normalize the weights such that $\sum_{i=1}^{N} w_{t+1,i} = 1$.
6. If $t < M$, set $t = t + 1$ and go to step, 2.

The output of classifier consists of the structures of M decision trees and the weighted log-ratio for each node of decision tree, i.e., $f_{t,k}$ ($t = ,\ldots, M$, $k = 1, 2, \ldots, K$). To avoid overfitting and stabilize the learning process, a small constant is added to the numerator and denominator when computing the log-likelihood ratio for each node. When training the classifier of Adaboosted decision trees, two important parameters are required to set: the number of weak learner trees and the number of nodes per tree. For choosing these parameters, we perform experiments and compare the classification accuracies as they vary in certain range.

In feature extraction, we construct multiple feature vectors from the multi-scale complex filter responses of detected singularities. Based on the feature vectors of training set at each scale, we train separate classifiers of Adaboosted decision trees to distinguish among the fingerprint classes. Each class is learned in a one-vs-all fashion. For example, to distinguish the five common fingerprint classes, we train five classifiers of Adaboosted decision trees to estimate the probabilities of a fingerprint being Arch, Tented Arch, Left Loop, Right Loop and Whorl. The output of Adaboost classifier for a test fingerprint is a number which represents the confidence that the fingerprint belongs to the corresponding class. We re-scale these confidence values to the range of [0 1] using logistic regression. Thus, the resulting values can be interpreted as the class probabilities of fingerprint being Arch, Tented Arch, Left Loop, Right Loop and Whorl, which is denoted as $\psi(y|x_s)$ at the s scale. This makes it easier to combine the classification results of multiple scales. To get a class probability for a test fingerprint, we simply sum the class probabilities over all scales, i.e., $\sum_{s=1}^{4}\psi(y|x_s)$. Finally, the class with the maximum probability is assigned to the test fingerprint.

4. Experimental results

4.1. Data set

NIST special fingerprint database 4 (NIST-4) is one of the most important benchmarks for fingerprint classification. Most published results on fingerprint classification are based on this database. For comparison with other approaches, we also perform our fingerprint classification algorithm on this database. NIST-4 contains 4000 fingerprints of size 480×512 pixels, taken from 2000 fingers with two instances per finger. The first fingerprint instances are numbered from f0001 to f2000 and the second fingerprint instances are numbered from s0001 to s2000. All fingerprints in this database are used in our experiment. This fingerprint database is divided into training set and test set. The training set contains the 2000 fingerprints from the first 1000 fingers (f0001 to f1000 and s0001 to s1000) and the remaining 2000 fingerprints (f1001 to f2000 and s1001 to s2000) are used for the test set. The five classes: Arch, Tented Arch, Left Loop, Right Loop and Whorl are evenly distributed in this database.

The performance of a fingerprint classification algorithm is often measured in terms of classification error rate or accuracy. The error rate is computed as the ratio between the number of misclassified fingerprint and the total number of fingerprints in the test set. Each fingerprint in NIST-4 is assigned to one or two of the five classes by human experts. There exist about 17% ambiguous fingerprints that have two classes assigned to them. In our experiments, we make use of only the first class labels of the training set to train the fingerprint classifier which can simplify the training procedure. For testing, however, the classifier output is considered as correct classification if it matches any one of the pre-specified fingerprint classes. This consideration is also adopted by other researchers in comparing the classification results on the NIST-4 database [5–7,16,10,14,15,24,35]. We will report the error rate of our fingerprint classification algorithm on the NIST-4 database for the five-class problem.

4.2. Analysis of multi-scale feature extraction

We first evaluate the effect of the multi-scale feature extraction on the classification accuracy for the five fingerprint classes problem. In feature extraction, we compute the complex filter responses at four scales for the detected singularities and a feature vector is constructed for each scale. We train and test the individual Adaboosted decision trees based on the feature vectors extracted at each scale and get a classification result. The final fingerprint classification result is based on the ensemble of the results estimated at all scales. In this experiment, we set the number of weak leaner trees to 3, 4, and 5, and the number of nodes for each tree is set to 20. In addition, we also give the classification error rate by ensemble of the multi-scale classification results for comparison. Table 1 shows the classification results based on different scales on NIST-4 fingerprint database. We can see that the classification result based on the ensemble of the classification results in multiple scales is better than the results of individual scales.

Table 1
The classification error rates (%) at different scales on NIST-4.

Error rate (%)	Scale 1	Scale 2	Scale 3	Scale 4	Multiple scales
3 trees	8.55	8.40	8.35	8.05	6.55
4 trees	8.10	8.40	8.45	8.00	5.90
5 trees	8.80	8.95	8.70	8.65	6.30

4.3. Analysis of classifier parameters

The classifier for fingerprint classification is designed by using the Adaboosted decision trees. In fact, Adaboosted decision trees sequentially carve the feature space into a set of hypercubes and estimate the conditional likelihood ratio for each class. The granularity of hypercubes can be made arbitrarily fine when increasing the number of weak learner trees. But the likelihood ratios for data points falling into each cube are affected by the size of tree, i.e., the number of leaf nodes in each decision tree. If the decision tree has only two leaves, the strong classifier is based on sum of functions that takes only one attribute as input. Thus, when using Adaboosted decision trees, there are two important parameters to choose: the number of weak learner trees and the number of leaves per tree. In general, increasing the number of decision trees will not degrade the classification accuracy, as the Adaboost learning method is robust to overfitting to some extent.

We have performed experiments for analysis of these two classifier parameters. The classification performance has been examined by tuning the number of weak learner trees and tuning the number of leaves per tree. When changing the number of leaves per decision tree, we keep the number of trees fixed at 4. Similarly, when changing the number of decision trees, we keep the number of leaf nodes per tree fixed at 20. Figs. 7(a) and (b) show the classification error rates while tuning these two classifier parameters. From Fig. 7(a), we can see that increasing the number of weak learner trees over 4 tends to slightly degrade the classification performance. This is probably because larger weight will be given to the noisy training data caused by poor quality fingerprints (which is difficult to classify) in the later weak learner trees and result in overfitting. From Fig. 7(b), we can see that the classification performance tends to be stable when the number of leaf nodes per tree is larger than 15. The best classification result is obtained for Adaboosted decision trees with about 4 decision trees and 20 leave nodes per tree. Thus, we perform the classifier on the multi-scale singularities and make fingerprint classification by the classification results at multiple scales. The confusion matrix for the five classes in our proposed approach is shown in Table 2 where W, L, R, T and A represent the Whorl, Left Loop, Right Loop, Tented Arch and Arch fingerprint classes, respectively.

To test the computational complexity, the proposed algorithm is implemented by using MATLAB programming language and is executed under Windows XP Professional O.S. on a PC with Intel Pentium (R) Dual-Core CPU E5200 at 2.50 GHz. The computation time of Adaboost Learning from singularity features is 4.8 s for each classifier. But it is offline training process and not very important for

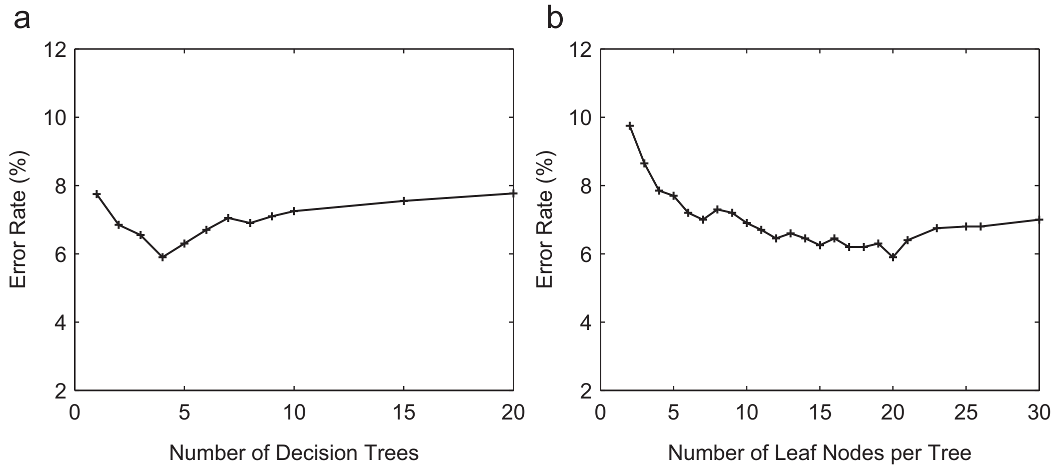

Fig. 7. Analysis of classifier parameters: (a) classification error rate vs. the number of decision trees with 20 nodes per tree and (b) classification error rate vs. the number of leaf nodes per decision tree when Adaboosting 4 decision trees. The classification results are robust to the classifier parameters to some extent.

Table 2
Confusion matrix for five fingerprint classes.

True class	Hypothesized class				
	W	L	R	T	A
W	365	10	15	0	8
L	1	390	0	7	9
R	0	0	375	11	12
T	0	3	3	320	36
A	0	1	0	2	432

classification system. For a test fingerprint, the on-line classification in our proposed approach takes 1.6 s on average. Most of the computation time is spent on the detection of singularities.

4.4. Performance comparison

We have compared the proposed approach to other approaches that have previously been published in the literature. Since Arch and Tented Arch fingerprint classes have a substantial overlap, it is very difficult to separate these two classes. Some researchers combine these two classes into one Arch class. In Table 3, we show the classification results reported by other researchers as well as the results of the proposed approach on both five- and four-class (i.e., 5-c and 4-c in the table) problems for comparison. All these classification results are based on the NIST-4 fingerprint database. The rule-based classification approaches [5,14,6] were tested on the whole NIST-4 database while the other learning-based approaches were tested on the 2nd half of NIST-4. The features and classifiers used in these approaches are also listed in Table 3.

From Table 3, we can see that the proposed approach performs better on fingerprint classification than most of the published approaches. In other reported works on fingerprint classification, the number and relative position of singularities are used to generate heuristic classification rules with the ridge information extracted as supplement [5,14]. In addition, orientation field and Gabor filters responses which can capture more detailed ridge flow information than singularities are used for fingerprint classification via some learning techniques [7,35,6,10]. These features are also combined with singularities by using some learning techniques for fingerprint classification [15,24]. Comparing to these related approaches, the proposed approach has the following three features. Firstly, a compact feature vector of 16 dimensions is constructed that not only indicates the number and relative position of detected singularities but also includes their certainties and relative directions. Thus, it can provide rich information on the singularities without tracing the ridges and using high-dimensional feature vector, e.g., 192 dimensions in [6,10,24] and 1680 dimensions in [7,35]. Secondly, the classifier of Adaboosted decision trees is employed to effectively model the multiple types of singularities features (i.e., certainties, relative position and relative direction) and automatically select the discriminatory features for each class. Finally, the classification results using multi-scale singularities are combined to make fingerprint classification which is more powerful than that using single scale. In summary, our proposed approach is able to make use of the rich information on singularities and the results show that this approach has good performance of fingerprint classification. Most of misclassifications in the proposed approach are caused by heavy noise in the poor quality fingerprints where it is very difficult to correctly detect the singularities (see Fig. 8a). Our approach also fails to correctly classify some fingerprint images where a pair of core and delta points are too close to detect (see Fig. 8b).

5. Conclusions

In this paper, a fingerprint classification algorithm is presented that uses the Adaboosted decision trees to model the feature set of singularities. The complex filters of the first order are used to detect the singularities. There are three main contributions in this paper. Firstly, instead of using the number and relative positions of singularities, we propose to construct a 16-dimensional feature vector as classifier input that can not only reflect the number and relative positions of singularities but also evaluate their relative directions and certainty measurements. This representation of fingerprint can capture the rich information on singularities with compact feature. Secondly, to effectively model the multiple types of singularities features, we propose to design the classifier by using Adaboosted decision trees to make fingerprint classification, which can automatically select discriminatory features for fingerprint classes. Finally, the fingerprint class is determined by ensemble of classification results using multi-scale singularities which performs better than that using single scale. The experimental results and comparisons on the NIST-4 database demonstrate the effectiveness of the proposed fingerprint classification approach.

The selectivity of classification-based techniques for the latent fingerprint searching strongly depends on the number of classes and the natural distribution of fingerprints in these classes [1].

Table 3
Classification error rates of various fingerprint classification approaches for the five-class (5-c) and four-class (4-c) problems on NIST-4 (where NN: neural network; *k*-NN: *k*-nearest neighbor; SVM: support vector machines; MKL: multi-space generalization of the KL transform; SPD: subspace-based pattern discrimination; NDA: nonlinear discriminant analysis; NB: naive Bayes; AbDT: Adaboosted decision trees).

Algorithms	Features	Classifier	5-c (%)	4-c (%)	Test set
Candela et al. [7]	Orientation field	NN	–	11.4	2nd half
Karu and Jain [5]	Singularities	Rule based	14.6	8.6	Whole
Jain et al. [6]	Gabor filters	*k*-NN and NN	10	5.2	2nd half 1.8% rejects
Jain and Minut [16]	Ridge kernel	Rule based	–	8.8	Whole
Yao et al. [10]	Gabor filters	SVM	10.7	6.9	2nd half
Zhang and Yan [14]	Singularities and ridge	Rule based	15.7	7.3	Whole
Park and Park [9]	Fourier transform	NDA	9.3	6.0	2nd half
Hong et al. [24]	Singularities and Gabor filters	SVM and NB	9.2	5.1	2nd half
Cappelli et al. [35]	Orientation field	MKL and SPD	4.8	3.7	2nd half
Tan et al. [13]	Learned feature	Bayesian	8.4	6.7	2nd half
Li et al. [15]	Singularities and orientation field	SVM	6.5	5.0	2nd half
Our approach	Singularities	AbDT	5.9	4.3	2nd half

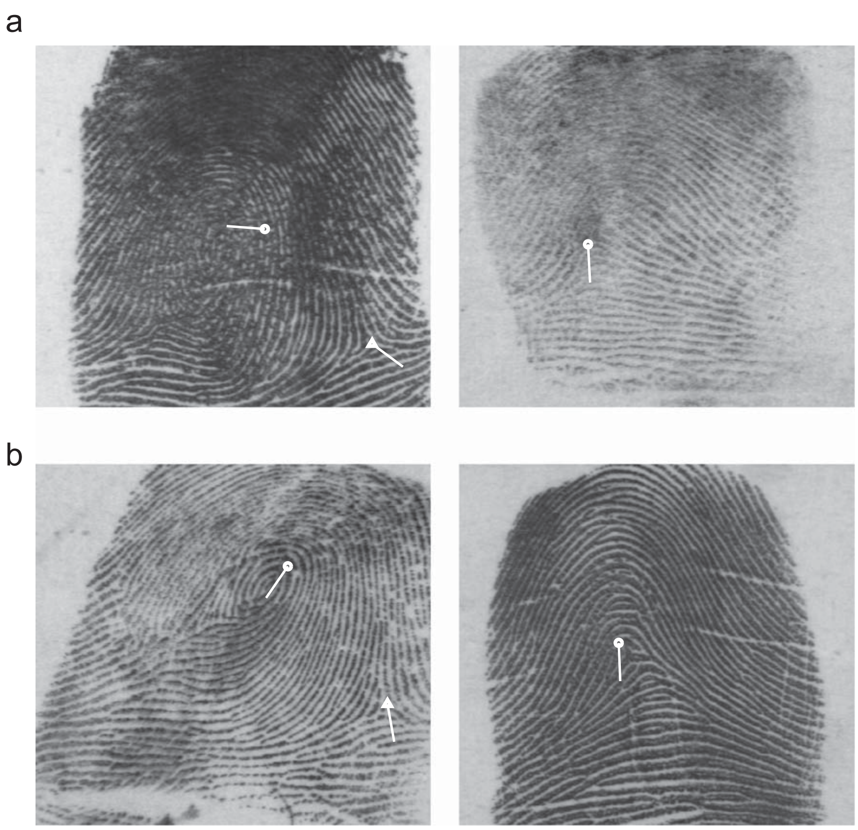

Fig. 8. Misclassifications of fingerprint images due to (a) heavy noise (b) one pair of core and delta points which are too close to detect (two whorl fingerprints on left are misclassified as Left Loop and two Tented Arch fingerprints on right are misclassified as plain arch).

However, the real fingerprints are non-uniformly distributed in the five classes. To alleviate the effect of this problem, we can perform the proposed classification approach followed by the continuous classification method [36], which is not constrained by the distribution, to improve the selectivity.

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 Transactions on Pattern Analysis and Machine Intelligence 18 (8) (1996) 799–813.

[3] M. Liu, X.D. Jiang, A.C. Kot, Efficient fingerprint search based on database clustering, Pattern Recognition 40 (6) (2007) 1793–1803.

[4] E.R. Henry, Classification and Uses of Fingerprints, Routledge, London, 1900.

[5] K. Karu, A.K. Jain, Fingerprint classification, Pattern Recognition 29 (3) (1996) 389–404.

[6] A.K. Jain, S. Prabhakar, L. Hong, A multichannel approach to fingerprint classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (4) (1999) 348–359.

[7] 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, August 1995.

[8] R. Cappelli, D. Maio, D. Maltoni, Fingeprint classification based on multi-space KL, in: Proceedings of Workshop on Automatic Identification Advanced Technologies, 1999, pp. 117–120.

[9] C.H. Park, H. Park, Fingerprint classification using fast Fourier transform and nonlinear discriminant analysis, Pattern Recognition 38 (4) (2005) 495–503.

[10] Y. Yao, P. Frasconi, M. Pontil, Fingerprint classification with combination of support vector machine, in: Proceedings of 3rd International Conference on Audio- and Video-based Biometric Person Authentication, 2001, pp. 253–258.

[11] A. Senior, A combination fingerprint classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (10) (2001) 1165–1174.

[12] G.L. Marcialis, F. Roli, P. Frasconi, Fingerprint classification by combination of flat and structural approaches, in: Proceedings of International Conference on Audio- and Video-based Biometric Person Authentication (3rd), 2001, pp. 241–246.
[13] X. Tan, B. Bhanu, Y. Lin, Fingerprint classification based on learned features, IEEE Transactions on System, Man and Cybernetics, Part C 35 (3) (2005) 287–300 (special issue on Biometrics).
[14] Q. Zhang, H. Yan, Fingerprint classification based on extraction and analysis of singularities and pseudo ridges, Pattern Recognition 37 (11) (2004) 2233–2243.
[15] J. Li, W.-Y. Yau, H. Wang, Combining singular points and orientation image information for fingerprint classification, Pattern Recognition 41 (1) (2008) 353–366.
[16] A.K. Jain, S. Minut, Hierarchical kernel fitting for fingerprint classification and alignment, in: Proceedings of International Conference on Pattern Recognition (16th), vol. 2, 2002, pp. 469–473.
[17] J.-H. Chang, K.-C. Fan, A new model for fingerprint classification by ridge distribution sequences, Pattern Recognition 35 (2002) 1209–1223.
[18] D. Maio, D. Maltoni, A structural approach to fingerprint classification, in: Proceedings of 13th International Conference on Pattern recognition, vol. 3, Vienna, 1996, pp. 578–585.
[19] R. Cappelli, A. Lumini, D. Maio, D. Maltoni, Fingerprint classification by directional image partitioning, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (5) (1999) 402–421.
[20] M.S. Pattichis, G. Panayi, A.C. Bovik, S.-P. Hsu, Fingerprint classification using an AM–FM model, IEEE Transactions on Image Processing 10 (6) (2001) 951–954.
[21] U. Hallici, G. Ongun, Fingerprint classification through self-organizing feature maps modified to treat uncertainties, Proceedings of the IEEE 84 (10) (1996) 1497–1512.
[22] K.A. Nagaty, Fingerprints classification using artificial neural networks: a combined structural and statistical approach, Neural Networks 14 (2001) 1293–1305.
[23] Y. Yao, G.L. Marcialis, M. Pontil, P. Frasconi, F. Roli, Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines, Pattern Recognition 36 (2) (2003) 397–406.
[24] J.-H. Hong, J.-K. Min, U.-K. Cho, S.-B. Cho, Fingerprint classification using one-vs-all support vector machines dynamically ordered with naïve Bayes classifiers, Pattern Recognition 41 (2) (2008) 662–671.
[25] L. Hong, Y. Wan, A.K. Jain, Fingerprint image enhancement: algorithm and performance evaluation, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (8) (1998) 777–789.
[26] A.M. Bazen, S.H. Gerez, Systematic methods for the computation of the directional fields and singular points of fingerprints, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 905–919.
[27] X.D. Jiang, On orientation and anisotropy estimation for online fingerprint authentication, IEEE Transactions on Signal Processing 53 (10) (2005) 4038–4049.
[28] M. Liu, X.D. Jiang, A.C. Kot, Fingerprint reference point detection, EURASIP Journal on Applied Signal Processing 2005 (4) (2005) 498–509.
[29] A.M. Bazen, S.H. Gerez, Segmentation of fingerprint images, in: Proceedings of ProRISC2001, 12th Annual Workshop on Circuits, Systems and Signal Processing, 2001, pp. 276–280.
[30] J. Bigun, Pattern recognition in images by symmetries and coordinate transformations, Computer Vision and Image Understanding 68 (3) (1997) 290–307.
[31] K. Nilsson, J. Bigun, Localization of corresponding points in fingerprints by complex filtering, Pattern Recognition Letters 24 (2003) 2135–2144.
[32] M. Liu, X.D. Jiang, A.C. Kot, Fingerprint retrieval by complex filter responses, in: Proceedings of the 18th International Conference on Pattern Recognition, Hong Kong, China, 2006, pp. 1042–1045.
[33] B. Roe, H. Yang, J. Zhu, Boosted decision trees, a powerful event classifier, in: Proceedings of PHYSTAT05 (Statistical Problems in Particle Physics, Astrophysics and Cosmology), Oxford, UK, 2005.
[34] M. Collins, R. Schapire, Y. Singer, Logistic regression, Adaboost and Bregman distances, Machine Learning 48 (1–3) (2002) 253–285.
[35] R. Cappelli, D. Maio, D. Maltoni, L. Nanni, A two-stage fingerprint classification system, in: Proceedings of the 2003 ACM SIGMM Workshop on Biometrics Methods and Applications, 2003, pp. 95–99.
[36] L. Sha, X. Tang, Combining exclusive and continuous fingerprint classification, in: Proceedings of 2004 International Conference on Image Processing, vol. 2, 2004, pp. 1245–1248.

About the Author—MANHUA LIU received Ph.D. degree in 2008 from EEE, Nanyang Technological University, Singapore. Before that, she got B.S. degree in 1997 and M.Eng. degree in 2002 from North China Institute of Technology and Shanghai Jiao Tong University, China, respectively. Currently, she is a lecturer in Department of Instrument Science and Engineering, School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University, China. Her research interests include biometrics, pattern recognition, image processing and machine learning and so forth.