



Fingerprint classification by a hierarchical classifier[☆]

Kai Cao^a, Liaojun Pang^a, Jimin Liang^a, Jie Tian^{a,b,*}

^a Life Sciences Research Center, School of Life Sciences and Technology, Xidian University, Xi'an 710071, China

^b Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China



ARTICLE INFO

Article history:

Received 20 March 2012

Received in revised form

25 March 2013

Accepted 9 May 2013

Available online 23 May 2013

Keywords:

Fingerprint classification

Complex filter response

Support vector machine

Ridge line flow

Hierarchical classifier

ABSTRACT

Fingerprint classification is still a challenging problem due to large intra-class variability, small inter-class variability and the presence of noise. To deal with these difficulties, we propose a regularized orientation diffusion model for fingerprint orientation extraction and a hierarchical classifier for fingerprint classification in this paper. The proposed classification algorithm is composed of five cascading stages. The first stage rapidly distinguishes a majority of Arch by using complex filter responses. The second stage distinguishes a majority of Whorl by using core points and ridge line flow classifier. In the third stage, K-NN classifier finds the top two categories by using orientation field and complex filter responses. In the fourth stage, ridge line flow classifier is used to distinguish Loop from other classes except Whorl. SVM is adopted to make the final classification in the last stage. The regularized orientation diffusion model has been evaluated on a web-based automated evaluation system FVC-onGoing, and a promising result is obtained. The classification method has been evaluated on the NIST SD 4. It achieved a classification accuracy of 95.9% for five-class classification and 97.2% for four-class classification without rejection.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

The identification of a person with an AFIS (automated fingerprint identification system) requires a comparison of his/her fingerprint with all the fingerprints in a database [1]. This database may be very large in many forensic and civilian applications, which leads to long processing time and deteriorated accuracy, hence it is unsuitable in real time applications. A common strategy to speed up the search is to divide the fingerprint database into a number of groups that have similar properties. Fingerprint classification that assigns a fingerprint to a class (based on predefined classes) is an effective way.

Most of the classification algorithms are based on the Galton–Henry classification scheme [2]. Five most common classes of the Galton–Henry classification scheme are Arch (A), Tented Arch (T), Left Loop (L), Right Loop (R) and Whorl (W), as shown in Fig. 1. Even five classes are involved, fingerprint classification is still a challenging problem due to small inter-class variability, large intra-class variability and poor quality fingerprints [3,1]. Many fingerprint classification algorithms have been proposed to deal

with these problems. Among these algorithms, almost all these methods are based on one or more of the following features [1]: singular points, ridge line flows, orientation image and Gabor filter responses [3].

Singular points are one of the characteristic features of fingerprints. There are usually two kinds of singular points in fingerprints, namely core point and delta point. Singular points can be extracted by Poincaré index [4], complex filters [5], zero-pole model [6] and so on. A fingerprint can be simply classified according to the quantity, types and relative positions of fingerprint singular points. This is also the approach commonly used by human experts for manual classification [1]. Due to its simplicity and distinguishability, a number of singular points based approaches have been proposed in the literature [4,7–9]. Recently, an Adaboost learning method was proposed to model multiple types of singular feature [10], in which a feature vector was constructed by detecting singular points at multiple scales then Adaboost learning method was applied on decision trees to design a classifier. However, singular point detection is sensitive to noise and extracting the singular points reliably itself is a very challenging problem. The noise in fingerprint image makes the singular point extraction unreliable, including missed or wrong detection.

The ridge line flows can be traced by drawing curves according to the orientation images [11]. Chang et al. [12] defined 10 basic ridge patterns, and used the distribution of such ridge patterns as features. Classification was conducted by analyzing the ridge shape and a sequence of ridge distribution. Dass and Jain [13] analyzed

[☆]This paper is supported by the National Natural Science Foundation of China under Grant nos. 60902083, 60803151, 60875018, 61100234, 61101247, the Fundamental Research Funds for the Central Universities.

* Corresponding author at: Life Sciences Research Center, School of Life Sciences and Technology, Xidian University, Xi'an 710071, China.

Tel.: +86 10 82618465, +86 29 88201842. fax: +86 10 62527995.

E-mail address: tian@ieee.org (J. Tian).

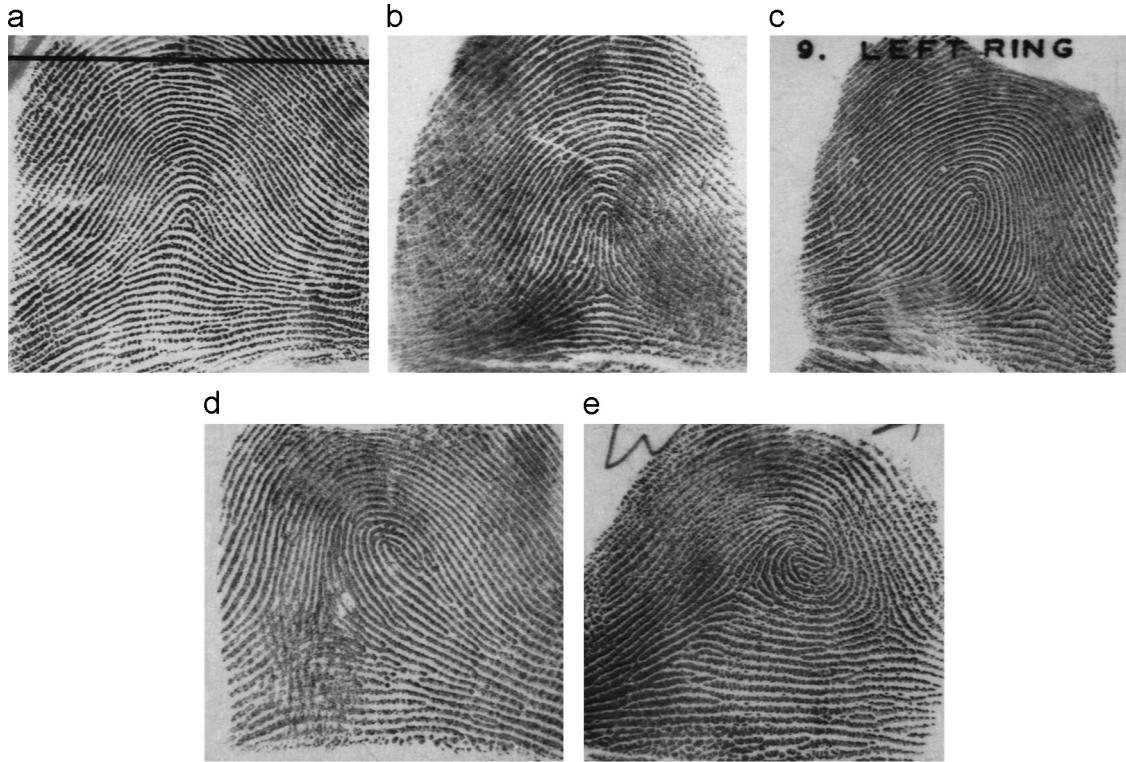


Fig. 1. Examples of common five fingerprint classes: (a) Arch, (b) Tented Arch, (c) Left Loop, (d) Right Loop and (e) Whorl.

the isometric maps for tangent planes as a point traversed from one end to the other end. The main drawback of ridge line flow based methods is that they are not able to distinguish Tented Arch from Arch. When delta point is close to core point, it is also difficult for these methods to distinguish Right Loop and Left Loop from Tented Arch. Therefore, the ridge line flows were usually used to make up the singular point based fingerprint classification [14,15] or used with other features to train the classifiers [16].

Most of the existing fingerprint classification algorithms made use of the orientation image [1]. In fact, singular points and ridge line flow can also be extracted from orientation image. Many approaches directly used the orientation image as a feature vector, by simply nesting its rows. Each element of the orientation image was encoded with two component $[r \cdot \cos 2\theta, r \cdot \sin 2\theta]$ [17], where r gave a significance value of the orientation. This approach resulted in high dimensional feature vectors. To reduce memory consumption and computation complexity, some dimension reduction techniques have been adopted to reduce the feature dimension [11,17–20]. The Karhunen–Loeve (KL) transform, which guarantees a good preservation of Euclidean distances between vectors, was usually adopted for this purpose [11,18,20]. A generalization of KL transform called MKL was used not only for dimension reduction but also for classification [21]. However, the orientation image based methods are challenged by small inter-class variabilities, as shown in Fig. 2.

Both singular points and orientation image are effective for fingerprint classification. Meanwhile, they both have their shortcomings. Researchers tried to combine them to improve the classification accuracy. In [22], the locations of the singular points were used together with an orientation image for the training of two disjoint neural networks, whose outputs were fed to a third one, which produced the final classification. Recently, coefficients of the orientation model were combined with the information of the singularities (the number of singular points and their relative positions) to represent fingerprints [23].

Many learning approaches have been proposed for fixed-size fingerprint features classification. Neural networks were first widely used in fingerprint classification. These neural networks include multilayer artificial neural network [22,24], probabilistic neural network [11] and self-organizing neural network [25,26]. Support vector machine (SVM), which is founded in statistical learning theory, is a relatively powerful tool for pattern classification [27]. It also has been used in fingerprint classification [23,16]. Furthermore, different classifiers were integrated to improve classification accuracy. Support vector machine and recursive neural network were combined in [28]. PCASYS combined two other classifiers: the hidden Markov model classifier and an approach based on ridge shape feature, to achieve fingerprint classification. Two-stage classification strategy was another integrating approach. Jain et al. [3] proposed to use K-NN classifier to find two most probable classes from a FingerCode feature vector, then a specific neural network was trained to distinguish the two classes. Cappelli et al. [21] adopted similar strategy by using MKL-based classifier to find two most probable classes and using SPD (subspace-based pattern discrimination) classifier to make final decision. Shah and Sastry [29] proposed another two-stage strategy by separating Arch and Tented Arch classes from Loop and Whorl at the first stage.

To facilitate high-performance classification, algorithms for accurate orientation estimation and classification algorithm are needed. In this paper, we propose a regularized orientation diffusion model for fingerprint orientation extraction, which has been submitted to a web-based automated evaluation system FVC-onGoing [30] and a promising result is obtained. Orientation image, complex filter responses as well as ridge line flows are used to represent a fingerprint. A novel hierachic classifier is proposed for fingerprint classification. The proposed algorithm is evaluated on the NIST SD4 [31]. The experiments demonstrate its effectiveness and an accuracy of 95.9% for five-class problem is obtained. The rest of the paper is organized as follows: Section 2

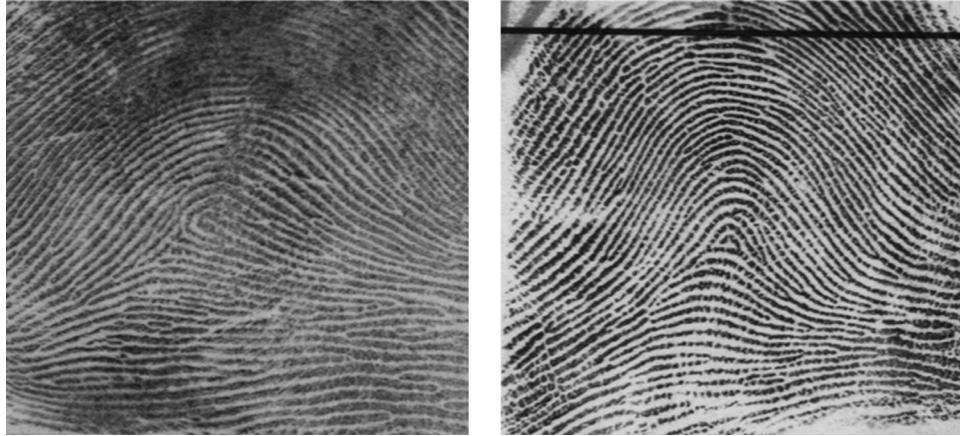


Fig. 2. Examples of two different type fingerprints with small inter-class variabilities.

describes the fingerprint orientation extraction. Section 3 discusses the classification method. The experimental results are reported in Section 4 and conclusions are drawn in Section 5.

2. Fingerprint orientation extraction

An accurate estimation of fingerprint orientation image is essential in fingerprint classification. The following steps are performed to obtain the orientation image for a given fingerprint.

2.1. Coarse orientation extraction

Root filtering [32] is a fingerprint enhancement technique in the Fourier domain, which has been demonstrated to be an effective approach for low-quality fingerprints. The fingerprint is firstly divided into overlapping block with block size $blkSize$ and each block is enhanced by the root filtering

$$I_{enh}(x, y) = FFT^{-1}\{F(u, v) \cdot |F(u, v)|^k\} \quad (1)$$

where k is an experimentally determined constant ($k=0.5$ in this work) and

$$F(u, v) = FFT(I(x, y)) \quad (2)$$

Then gradient-based method is adopted to estimate the local ridge orientation and coherence [33]. The orientation of the block centered at (x, y) is given by

$$\theta(x, y) = \frac{1}{2} \tan^{-1} \left(\frac{2G_{xy}(x, y)}{G_{xx}(x, y) - G_{yy}(x, y)} \right) + \frac{\pi}{2} \quad (3)$$

where

$$G_{xx}(x, y) = \sum_{(u, v) \in W} G_x^2(u, v) \quad (4)$$

$$G_{yy}(x, y) = \sum_{(u, v) \in W} G_y^2(u, v) \quad (5)$$

$$G_{xy}(x, y) = \sum_{(u, v) \in W} G_x(u, v)G_y(u, v) \quad (6)$$

W is the neighborhood window, G_x and G_y are the gradients in horizontal and vertical directions, respectively. For small block size, a smoothing over G_x and G_y is preferred. The quality over each block $q(x, y)$ is determined by its magnitude of gradient $M(x, y)$ and coherence $coh(x, y)$, formally

$$q(x, y) = f(M(x, y), M_1, M_2) \cdot f(coh(x, y), C_1, C_2) \quad (7)$$

where

$$M(x, y) = \sqrt{G_{xx}(x, y) + G_{yy}(x, y)}/blkSize \quad (8)$$

$$coh(x, y) = \frac{\sqrt{(G_{xx}(x, y) - G_{yy}(x, y))^2 + 4G_{xy}^2(x, y)}}{G_{xx}(x, y) + G_{yy}(x, y)} \quad (9)$$

and f is a piecewise linear function

$$f(x, th_1, th_2) = \begin{cases} 0 & \text{if } x \leq th_1, \\ 1 & \text{if } x > th_2, \\ \frac{x - th_1}{th_2 - th_1} & \text{otherwise.} \end{cases} \quad (10)$$

M_1, M_2, C_1, C_2 are parameters which are determined by experience.

2.2. Regularized orientation diffusion model

Due to scars, ridge breaks and low gray value contrast, the estimated local ridge orientation, $\theta(x, y)$, may not always be correct and obtaining a reliable orientation image is still a great challenge in poor quality fingerprint. Therefore, the resulting orientation image must be regularized in order to meet at least two requirements. On one hand it has better keep the orientation in the region with clear ridge and valley structure. And on the other hand it has to be smooth, especially in low-quality region. Moreover, the orientation field needs to be regularized in order to extend to the background for fingerprint classification task. In order to solve the ambiguity of θ and $\theta + \pi$, the orientation field needs to be converted into a continuous vector field $\mathbf{v} = (v_1, v_2)$, which is defined as follows:

$$v_1(x, y) = \cos(2\theta(x, y))$$

$$v_2(x, y) = \sin(2\theta(x, y)) \quad (11)$$

The goal of the regularization is to obtain a vector field $\mathbf{u} = (u_1, u_2) \in \mathcal{R}^2$ that minimizes the following joint energy functional

$$J(\mathbf{u}) = D(\mathbf{u}) + \mu e(\mathbf{u}) \quad (12)$$

where D is a distance term that measures the difference between the original vector field and the regularized vector field, e is a penalty term that measures the roughness of the solution $\mathbf{u}(x)$, and $\mu > 0$ is a regularization parameter governing the tradeoff between the distance term and the penalty term. In this work the following

distance measure is adopted:

$$D(\mathbf{u}) = \frac{1}{2} \int_{\Omega} q(\mathbf{x}) \|\mathbf{u}(\mathbf{x}) - \mathbf{v}(\mathbf{x})\|^2 d\mathbf{x} \quad (13)$$

where the squared difference is weighted by the quality at \mathbf{x} . The regularization term ϵ is used to add some prior knowledge on the desired vector field. The most used of first order regularization methods is the gradient penalization, which has been used by Horn and Schunck in their classical optical flow [34] to obtain smooth optical flow. The regularization function in this case is defined as follows:

$$\epsilon(\mathbf{u}) = \int_{\Omega} \sum_{i=1}^2 \|\nabla u_i\|^2 d\mathbf{x} \quad (14)$$

This regularization method was also adopted by gradient vector flow [35]. It is obvious that when q is large, \mathbf{u} will be close to \mathbf{v} , on the other hand, \mathbf{u} will be smooth. Using the calculus of variations, the solution of (12) can be found by solving the following Euler–Lagrange equations

$$\begin{aligned} \mu \left(\frac{\partial^2 u_1}{\partial x^2} + \frac{\partial^2 u_1}{\partial y^2} \right) - q(u_1 - v_1) &= 0 \\ \mu \left(\frac{\partial^2 u_2}{\partial x^2} + \frac{\partial^2 u_2}{\partial y^2} \right) - q(u_2 - v_2) &= 0 \end{aligned} \quad (15)$$

In order to solve the Euler–Lagrange PDE, \mathbf{u} can be treated as functions of time and solving

$$\begin{aligned} \frac{\partial u_1}{\partial t} &= \mu \left(\frac{\partial^2 u_1}{\partial x^2} + \frac{\partial^2 u_1}{\partial y^2} \right) - q(u_1 - v_1) \\ \frac{\partial u_2}{\partial t} &= \mu \left(\frac{\partial^2 u_2}{\partial x^2} + \frac{\partial^2 u_2}{\partial y^2} \right) - q(u_2 - v_2) \end{aligned} \quad (16)$$

Near a singular point, these two surfaces of the vector field contain a discontinuity, a jump from -1 to 1 . Smoothing the discontinuity will result in false orientation around the singular point [36]. Keeping the vector norm may reduce the negative impact of smoothing. The method proposed by Tschumperle and Deriche [37] is easy to understand and implement. We adopt their method in this work. The norm constraint is equivalent to

$$\|\mathbf{u}\|^2 = \text{constant} \Leftrightarrow 2\mathbf{u} \cdot \frac{\partial \mathbf{u}}{\partial t} = 0 \quad (17)$$

This indicates that the PDE velocity vector $\partial \mathbf{u} / \partial t$ must be orthogonal to the vector \mathbf{u} to preserve the norm constraint. Suppose the

velocity vector obtained in (16) is \mathbf{m} , e.g.

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{m}. \quad (18)$$

The norm constraint can be naturally satisfied by projecting the unconstrained vector \mathbf{m} on the orthogonal direction of the vector \mathbf{u} , e.g.

$$P_{\mathbf{u}}^{\perp}(\mathbf{m}) = \mathbf{m} - \left(\frac{\mathbf{m} \cdot \mathbf{u}}{\|\mathbf{m}\|^2} \right) \mathbf{u}. \quad (19)$$

Then the following PDE preserves the vector norm at each point:

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{m} - \left(\frac{\mathbf{m} \cdot \mathbf{u}}{\|\mathbf{u}\|^2} \right) \mathbf{u}. \quad (20)$$

After solving the above equation, the final orientation at (x,y) is obtained by

$$\theta(x,y) = \frac{1}{2} \tan^{-1} \frac{u_2(x,y)}{u_1(x,y)} \quad (21)$$

3. Fingerprint classification

Complex filters have been proposed for the detection of patterns with radial symmetries [38,39]. Complex filters of the first order have similar patterns with fingerprint singular points, thus they are used to detect core points and delta points in fingerprint images [5]. This kind of methods are widely used because they can provide not only the type and position of singularity but also the direction and certainty. Complex filter responses characterize fingerprint singularities while orientation image represents the directionality of ridges. They have their own superiorities and they complement each other. In this work, we combine them with ridge line flows for the fingerprint classification.

3.1. Complex filtering and feature image registration

In this work, we use the filters of the first order symmetry or parabolic symmetry, i.e. $T_c = (x+iy)g_\sigma(x,y)$ and $T_d = (x-iy)g_\sigma(x,y)$, for the complex filtering, where g_σ is a 2D Gaussian with standard deviation σ and defined as $g_\sigma(x,y) = e^{-(x^2+y^2)/2\sigma^2}$. The orientation field of T_c and T_d is shown in Fig. 3. To facilitate the complex filtering, the orientation tensor in the complex domain is obtained by $z = \cos(2\theta) + i \sin(2\theta)$. The filter responses (R_c and R_d) are obtained by convolving the orientation tensor z and the conjugate

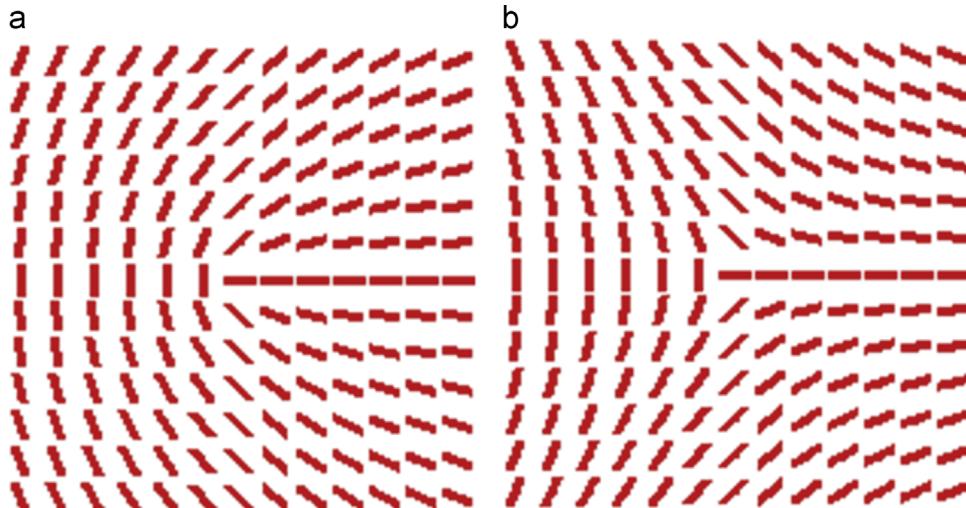


Fig. 3. Orientation fields of (a) T_c and (b) T_d .

of the templates, T_c and T_d , respectively, formally

$$R_c = z * T_c^* \quad (22)$$

$$R_d = z * T_d^* \quad (23)$$

Two heuristic rules proposed in [40] are incorporated into R_c to locate the reference point. Since the reference point is likely to occur at the center, response R_c is multiplied with the gaussian weighing function

$$g_{\sigma_1}(x, y) = \exp\left(-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma_1^2}\right) \quad (24)$$

where (x_0, y_0) is the centroid of the fingerprint, σ_1 is the kernel parameter with $\sigma_1 = \min(\text{height}, \text{width})/3$, *height* and *width* are the height and width of the orientation image, respectively. The angle of the reference point is centered around the vertical direction $\pi/2$. To embed this heuristic, R'_c obtained from the above rules is then projected onto the $\pi/2$ direction, yielding the final reference point response R''_c by using the dot product

$$R''_c = R'_c \cdot (n_{\pi/2}) \quad (25)$$

where $n_{\pi/2}$ is a unit vector in the vertical direction $\pi/2$. Then the point with the maximum value of R''_c is selected as a reference point. The orientation image as well as complex filter responses is registered by translating the reference point to the image centroid. Without introducing ambiguity, we still use z , R_c and R_d to represent the translated feature images.

3.2. Classifier

A hierachic classifier is proposed here for fingerprint classification. The detailed procedures of the proposed classification algorithm are illustrated in Fig. 4, in which there are total five stages. Compared with Right Loop, Left Loop and Tented Arch, Arch and Whorl are special as there are no singular points in Arch and there are four singular points in Whorl. Therefore, two heuristic rules are proposed to distinguish Arch and Whorl from Left, Right and Tented Arch in the first two stages. For Arch

fingerprint, there are no singular points and the magnitudes of the complex filter responses should be small. Therefore, it is very easy to distinguish Arch from other four types. If the maximum magnitudes of the complex filter responses are less (R_c and R_d) than a threshold T_1 , the fingerprint satisfies the following two conditions: (1) there are no singular points in this fingerprint and (2) the quality of the fingerprint is very good since noise will also result in large complex filter responses, which ensure the fingerprint should be Arch. Otherwise, the fingerprint may be one of other four types or still Arch and needs further classification. In order to improve the efficiency and reduce the noise near the bounder, we can only scan the blocks around the reference point.

There are two core points and two delta points in a Whorl fingerprint while there are only one core point and one delta point in a Loop or Tented Arch fingerprint. Due to these singularities, Whorl possesses very large variations, therefore, Whorl is taken as another special type and dealt with in the second stage. Firstly, core points are located based on the first order complex filter T_c . If a local maximum magnitude of R_c is larger than the threshold T_2 , it is considered to be a core point candidate. It is worthy noting that T_2 is larger than T_1 . If there are more than two core point candidates in a fingerprint, the two core points with maximum angle distance are selected as final core points. To reduce false classification rate at this stage, ridge line flows are extracted to enhance the classification. The ridge line flows are extracted from pixel-based orientation which is expanded from block-based orientation image obtained from Section 2 by bilinear interpolation. A ridge line flow with a starting point s_0 can be obtained by the following iteration:

$$s_j = s_{j-1} + d_j \cdot l \cdot o_{s_{j-1}} \quad (26)$$

where d_j , with values in $\{-1, 1\}$, is the flow direction from s_{j-1} to s_j , l is the length of the line segment and $o_{s_{j-1}}$ is the orientation vector at s_{j-1} . For each core point (x_c, y_c) , n_f ($n_f=3$ in our experiments) ridge line flows are extracted. The k th ($0 \leq k < n_f$) ridge line flow starts from $(x_c - k \cdot b \cdot \cos(\theta_c), y_c - k \cdot b \cdot \sin(\theta_c))$, where b is step size and θ_c is the direction of the core point.

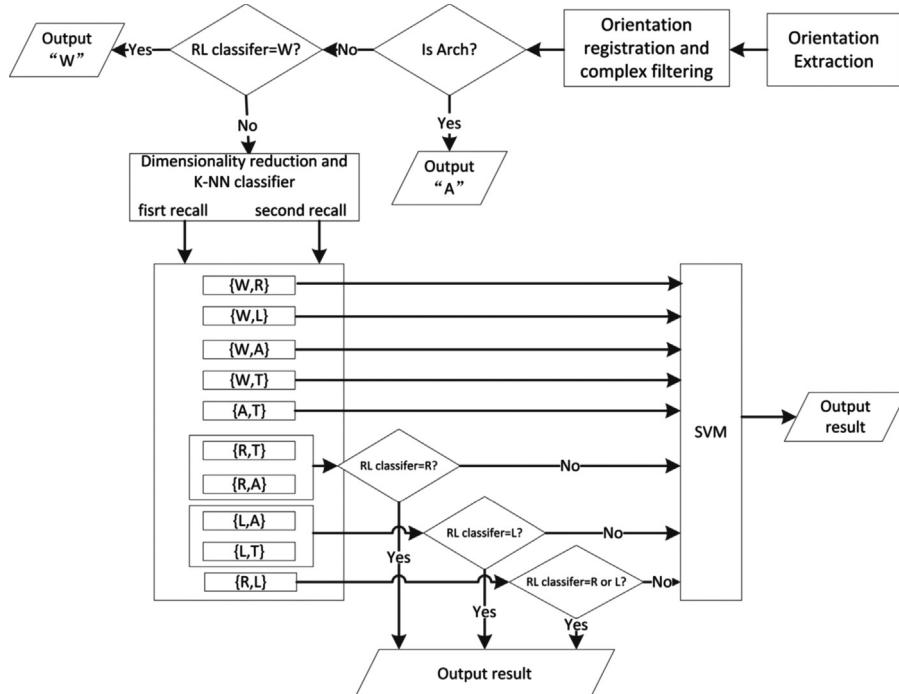


Fig. 4. Procedure of the proposed method.

Suppose that $C = \{(x_i, y_i)\}_{i=1}^n$ is the sampling point set of a ridge line flow from an end to the other end and n is the sampling number. The unit chord vector e_i is defined as $e_i = (1/\delta)(x_{i+1}-x_i, y_{i+1}-y_i)$, where δ is the norm of the vector $(x_{i+1}-x_i, y_{i+1}-y_i)$. The i th isometric map feature γ_i proposed in [13] is the cosine of the angle between e_i and e_1 which can be obtained as

$$\gamma_i = \langle e_1, e_i \rangle \quad (27)$$

where $i = 1, 2, \dots, n$, and $\langle \cdot, \cdot \rangle$ is the Euclidean inner product between e_1 and e_i . Fig. 5(a) illustrates an example of ridge line flow extraction in a Whorl fingerprint and Fig. 5(e) shows the corresponding graph of γ versus j as the point on a ridge line flow traverses from an end to the other one. A fingerprint is regarded to be Whorl if at least one of the ridge line flows satisfies the following conditions: (1) there are at least two sign-change points in the given isometric map, (2) there are at least two local maximum points close to 1 (the value of a local minimum is determined to be close to +1 if it exceeds λ) and (3) there is at least one local minimum point close to -1 (the value of a local maxima is determined to be close to -1 if its value falls below $-\lambda$). Otherwise, this fingerprint cannot be classified at this stage.

In the third stage, K -nearest neighbor (K -NN) classifier is adopted to find the top two classes which have the highest and the second highest count among the K nearest neighbors of the test pattern in the feature space. Orientation image represents the direction of the ridge flow while complex filter responses characterize the singularities of the ridge flow. They can make up the deficiencies of each other. Therefore, orientation tensor z as well as complex filter responses (R_c and R_d) is used for the feature representation. In order to increase the discriminability, two weighting methods are adopted: (1) a weight λ ($\lambda < 1$) is applied to the orientation tensor (λz) since the magnitude of the orientation tensor at each block is 1 while most magnitudes of complex filter responses are close to 0 except the singular points and noise area and (2) a Gaussian-like function (24) with σ_2 is applied to the three feature images to reduce the effects of noise close to the border. Without introducing ambiguity, we still use z , R_c and R_d to represent orientation tensor, complex filter responses. Then z , R_c and R_d are combined to form a feature vector x , formally

$$x = \text{vec}([\text{real}(y) \text{ } \text{imag}(y)]) \quad (28)$$

where $\text{vec}(\cdot)$ is the vectorization operator, $\text{imag}(\cdot)$ and $\text{real}(\cdot)$ are imaginary part and real part extractors, respectively, and

$$y = \text{vec}([\text{vec}(z) \text{ } \text{vec}(R_c) \text{ } \text{vec}(R_d)]) \quad (29)$$

The real part and imaginary part are taken as two features so that a fingerprint image of block size $m \times n$ will result in a vector of dimension $N = 2 \times 3 \times m \times n$, a very high dimensional feature vector. Principal component analysis (PCA) is used for dimensionality reduction, which is known as optimal dimensionality dimension [41]. Given an unknown pattern x , its dimensionality is first reduced from M to N ($N < M$). The K nearest neighbors of the test pattern in the reduced feature space are found. And then the classes which have the highest and the second highest count among the K nearest neighbors are selected.

If the top two classes are Left Loop and Tented (or Right Loop and Tented), the ridge line flow classifier is adopted to determine whether the fingerprint is Left (or Right) or not in the fourth stage. n_f ($n_f = 3$ in our experiments) ridge line flows are extracted by the iteration (26) with the starting points $(x_r - k \cdot b \cdot \cos(\theta_r), y_r - k \cdot b \cdot \sin(\theta_r))$, where (x_r, y_r) and θ_r are the reference point location and direction, respectively, b is step size and $0 \leq k < n_f$. The i th isometric map feature γ_i is the same as (27). Fig. 5(c) and (d) illustrates some examples of the ridge line flow extraction in Right Loop, Left Loop and Tented Arch, and Fig. 5(f)–(h) shows corresponding graphs of γ_j . A fingerprint is considered to be Left (Right) if there exists at least one ridge line flow satisfying the following conditions:

- (1) There is only one sign-change point in the given isometric map.
- (2) The maximum is close to 1 ($> \lambda$) and the minimum is close to -1 ($< -\lambda$).
- (3) Endpoints of sampling points set C are both on the left (right) side of the starting point.

Otherwise, it means the fingerprint cannot be distinguished at this stage. The same strategy is applied if the top two classes is Left and Right. It is worthy noting that ridge line flow classifier is used in the second stage and the fourth stage, and they cannot be merged in a single stage. In the second stage, we aim to distinguish Whorl from the other four classes, because Whorl fingerprints possess

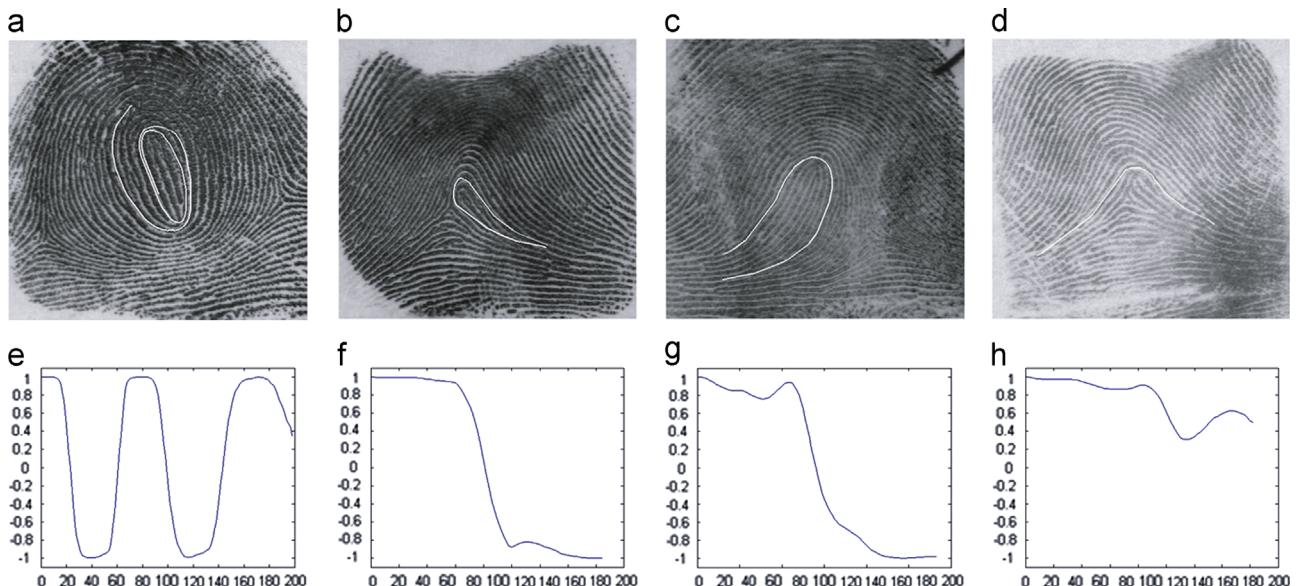


Fig. 5. Illustrations of ridge line tracing and graphs of γ_j versus j .

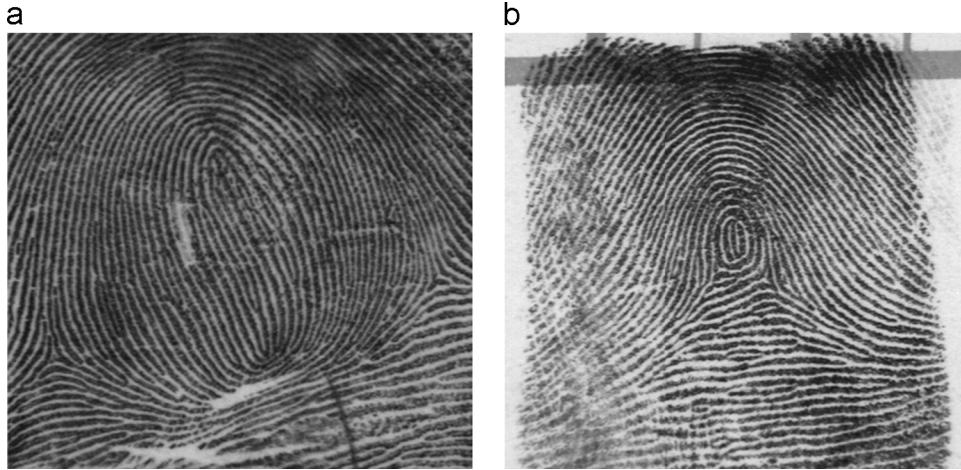


Fig. 6. Two examples of Whorl fingerprints. (a) Two core points are far away from each other and (b) the bottom core point is very close to the upper delta point.

very large variation and it is difficult for general classifiers to capture these variations. Fig. 6(a) shows an example that is mistaken as Right Loop by K-NN classifier and SVM since the bottom core point is out the image region after registration. But it can be correctly classified by ridge flow line extracted from the bottom core point. Similarly, if the fourth stage is moved to the second stage, then the fingerprint in Fig. 6(b) will be classified as Left Loop because the bottom core point is too close to the upper delta point. Although they are in different stages, they increase very little computational complexity.

Finally, support vector machine (SVM) makes the final decision between the top two classes. Totally, a set of $10 (C_2^5)$ SVMs is trained to solve 10 different two-class problems. The first and second recalls obtained by K-NN classifier is utilized to select the specific SVM which has been trained to distinguish the corresponding pair of classes. The feature vector is then fed into the selected SVM for final classification.

4. Experimental results

We evaluate our proposed algorithm from orientation estimation and classification results.

4.1. Performance of regularized orientation model

FVC-onGoing [30] is a web-based automated evaluation system for fingerprint recognition algorithms. The aim is to track the advances in fingerprint recognition technologies, through continuously updated independent testing and reporting of performances on given benchmarks. Recently, a new benchmark area has been added to assess the accuracy of fingerprint orientation extraction algorithms. There are two datasets of identical size (60 fingerprints): *Set A* and *Set B*. Each of them includes 10 fingerprints of good quality and 50 fingerprints of low and very low quality. All of the fingerprints are acquired using optical scanners. Each fingerprint includes its associated ground truth. The ground truth has been marked by a human expert. The comparison between estimated orientation image of a given fingerprint and its ground truth is performed by means of the root mean square deviation (RMSD). The accuracy over a whole dataset is denoted by the average RMSD. For the details, we refer readers to [42].

Firstly, we give the parameters used in this experiments: $\mu = 1.3$, $M_1 = 30$, $M_2 = 100$, $C_1 = 0.3$, $C_2 = 1$, $blkSize = 8$. Similar to [42], G_{xx} , G_{xy} and G_{yy} are smoothed with a Gaussian kernel ($\sigma = 3$

Table 1

The performance of fingerprint orientation extraction methods.

Method	AveErr on Set A (deg)			AveErr on Set B (deg)			Time (ms)
	Good	Bad	Ave	Good	Bad	Ave	
ROF	5.30	14.07	9.69	5.39	12.30	8.85	581
FOMFE [30]	—	—	—	6.70	21.44	14.07	1996
Adaptive-1 [42]	4.77	14.92	9.85	5.05	13.73	9.39	—
Adaptive-3 [30]	5.63	14.73	10.18	5.76	13.45	9.61	4772

in our experiments) obtain more smoothing orientation field. These parameters are selected by minimizing the average error of $AvgErr_A^G$ and $AvgErr_A^B$, where $AvgErr_A^G$ is the average error over good quality fingerprint images of Set A and $AvgErr_A^B$ is the average error over bad quality fingerprint images of Set A.

The proposed method (ROF) as well as several other approaches published on FVC-onGoing website is compared in Table 1, in which method FOMFE was submitted by the authors of Ref. [43] while Adaptive-1 and Adaptive-3 were submitted by the authors of Ref. [42]. In [42], the authors implemented and tested several well known methods and a plethora of their variants over FVC-onGoing orientation extraction benchmark, and their method worked better than others. From Table 1, we can find that the proposed approach achieves the overall best performance over both Sets A and B, especially over bad quality fingerprint images. Fig. 7 shows an example of orientation field extracted by the proposed approach over a bad quality fingerprint image.

4.2. Performance of classification

4.2.1. Dataset and performance measure

The proposed fingerprint classification algorithm is evaluated over NIST special database 4 (NIST SD 4) [31], one of the most important benchmarks for fingerprint classification. NIST SD 4 contains 4000 fingerprints captured from 2000 fingers with two instances (named "F" and "S") per finger. The images are numbered from F0001 to F2000 and from S0001 and S2000. Each image in NIST SD 4 has 8-bit gray-level with a size of 512×480 pixels. The five classes: Arch (A), Tented Arch (T), Left Loop (L), Right Loop (R) and Whorl (W) are uniformly distributed in the database. Each fingerprint in NIST SD 4 is assigned to one or two of the five classes by human experts.

In our experiments, all fingerprint images are used regardless of its quality. The first 2000 fingerprint images (f0001–f1000 and s0001–s1000) form training set, while the rest fingerprints form

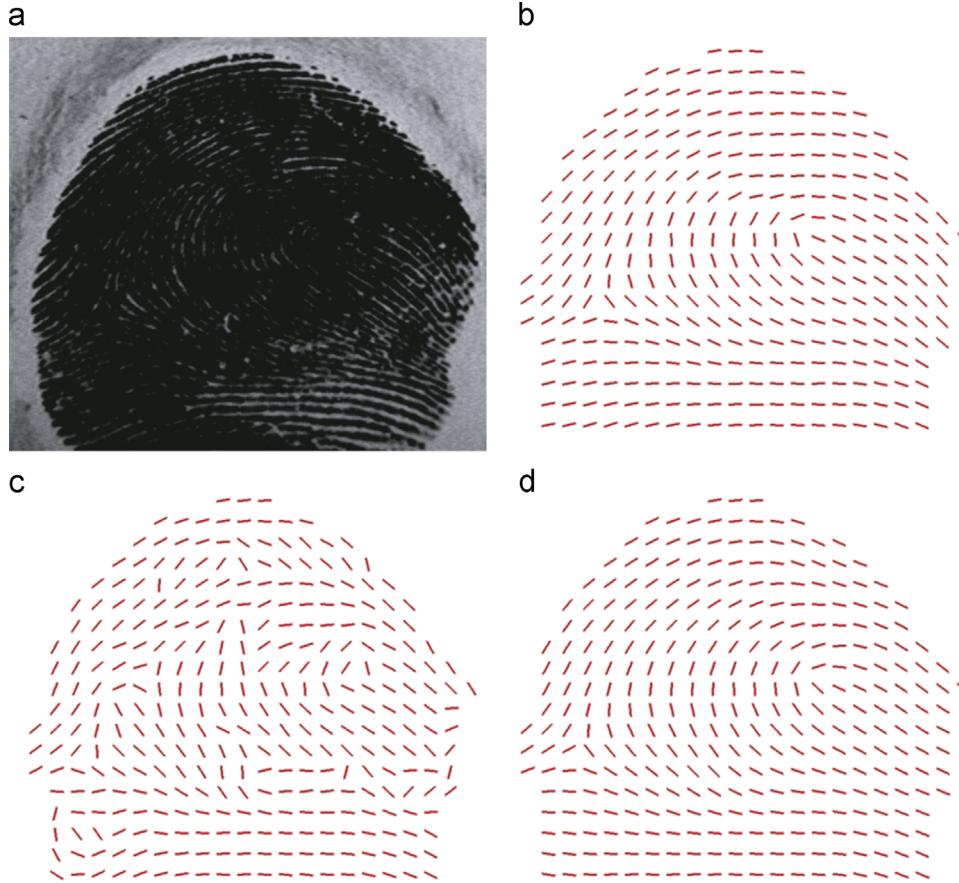


Fig. 7. Orientation extraction on a bad quality fingerprint. (a) Original fingerprint image, (b) ground truth orientation field, (c) gradient based orientation field, and (d) the proposed method.

testing set. The performance of a fingerprint classification algorithm is measured in terms of accuracy over testing set which is defined as the percentage of correctly classified. There exists about 17% ambiguous fingerprints that have two classes assigned to them in NIST SD 4. As other researchers [17,13,21,16,10], only the first label of a fingerprint is used for training while both the labels are used for test, in which, the classification is considered to be correct if the output matches any one of the labels. Since fingerprint classes A (Arch) and T (Tented Arch) have a substantial overlap, it is very difficult to separate these two classes. Therefore, we also report our results for the four-class classification problem, where classes A and T have been merged into one class.

4.2.2. Analysis of classifier parameters

The parameters in orientation model keep the same except that G_{xx} , G_{xy} and G_{yy} are not smoothed and median filtering is applied to coherence to keep singularities in fingerprint. We have performed some experiments for analysis of some other parameters. Threshold T_1 in the first stage is used to distinguish Arch from other four classes. If it is too large, many other type fingerprints will be classified as Arch. Many Arch will be rejected if it is too small. We have examined various values of T_1 over the training set. Fig. 8 shows FAR (false acceptance rate) at different values of T_1 (from 0.3 to 0.8 with step 0.02). When the value of T_1 is 0.56, the corresponding FAR is 0.93% less than 1% and when the value of T_1 changes from 0.56 to 0.58, FAR increases from 0.93% to 1.36%, a big gap. Therefore, we select $T_1=0.56$ as the best value of T_1 for the classification the following experiments. It is worth noting that FRR (false reject rate) is totally ignored in the selection because

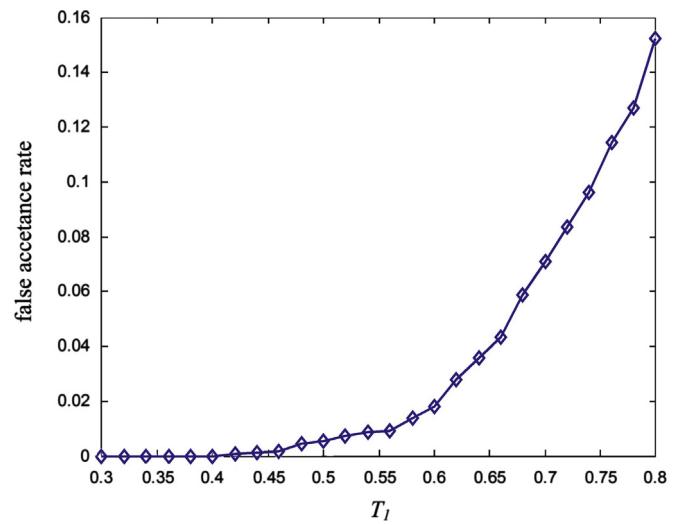


Fig. 8. False acceptance rate versus parameter T_1 .

rejected fingerprints in this stage will be fed to next stage for further classification.

There are three parameters to choose in the third stage: weighting value λ , reduced feature dimension N and the number of neighbors K in K -NN classifier. And there are two SVM parameters: C and γ in the fifth stage since the RBF kernel is selected. These parameters affect each other. K -NN classifier has been used to select two-most-likely classes in the previous work [3]. The experimental results in Ref. [3] indicated that $K=10$ was the best choice. Actually, $K=10$ is also a very good choice in our

work. λ balances the importance of orientation image and complex filter responses. $\lambda=0$ means that only orientation image is used for feature reduction and only complex filter responses are considered when $\lambda \rightarrow \infty$. Dimension N after dimensional reduction balances the energy preserving and classification accuracy. There are two approaches to select the value of N . The first one directly selects the best value of N while the second selects N by energy preserving. In this work, we use the second approach. Fig. 9 compares the performance of $\lambda=0.05$, $\lambda=0.1$, $\lambda=0.15$ with orientation image and complex filter responses at different energies, from which we can find that the best classification accuracy 95.9% is obtained when $\lambda=0.1$ and 93% energy is preserved. The kernel parameters are obtained by using the cross-validation and grid-search technique proposed by the developers when other parameters are fixed. Firstly, the training data is separated into v folds, then $v-1$ subsets are used to train the classifier for each SVM parameter pairs $C=2^{-2}, 2^{-1}, \dots, 2^{10}$, $\gamma=1, 2, \dots, 100$ and the remaining subset is used for test until the best cross-validation accuracy is obtained.

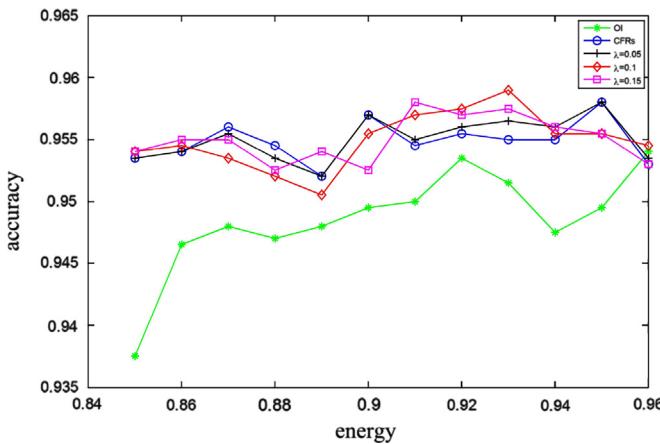


Fig. 9. Analysis of classifier parameters λ and energy, where OI denotes the orientation image and CFRs denotes complex filter responses.

Table 2
Confusion matrix for five fingerprint classes.

True class	Hypothesized class				
	A	T	L	R	W
A	418	16	1	0	0
T	10	330	6	5	0
L	1	6	385	2	5
R	2	12	1	396	3
W	3	1	4	4	389

Table 3
Classification accuracy of various fingerprint classification approaches on INST-4.

Algorithms	Features	Classifiers	5-class (%)	4-class %)	Test set
Cappelli et al. [21]	OI	MKL and SPD	95.2	96.3	2nd half
Candela et al. [11]	OI	NN	–	88.6	2nd half
Jain et al. [3]	GF	K-NN and NN	90	95.8	2nd half 1.8% rejects
Liu [10]	SP	AbDT	94.1	95.7	2nd half
Li et al. [23]	SP and OI	SVM	93.5	95	2nd half
Hong et al. [16]	SP and GF	SVM and NB	90.8	94.9	2nd half
Zhang and Yan [15]	SP and RF	Rule based	84.3	92.7	2nd half
Dass and Jain [13]	RF	Rule based	–	94.4	Whole
Chang and Fan [12]	RF	Rule based	94.8	–	First impressions
Our approach	RF, OI and CF	K-NN, rule based and SVM	95.9	97.2	2nd half

OI: orientation image; SP: singular point; RF: ridge line flow; CF: complex filter; GF: Gabor filter; NN: neural networks; K-NN: K-nearest neighbor; SVM: supported vector machine; NB: naïve Bayes classifier.

The confusion matrix for the five classes of our proposed algorithm is shown in Table 2, where A, T, L, R and W represent the Arch, Tented Arch and Left Loop, Right Loop and Whorl fingerprint classes, respectively.

4.2.3. Performance comparison and discussion

We list the results reported by other researchers along with the result of our proposed method in Table 3 for comparison. These methods were evaluated on the same database NIST SD 4. Not only the classification results on both five- and four-class problem but also the features and classifiers used in these approaches are also reported in Table 3. The number and relative position of singularities were usually used to generate heuristic rules classification with the pseudoridges as the supplement [15,16]. Recently, Ada-boost learning method was proposed to model multiple types of singularities features [10] and 94.8% classification accuracy was obtained. Jain et al. [3] proposed the Gabor filter responses, i.e. FingerCode, for the fingerprint classification and a two-stage classifier was used. K-NN classifier was used as the first stage to find the two most probable classes and neural network was then used to make the final decision. The method proposed by Cappelli et al. [21] used only orientation and the similar strategy was adopted, in which the MKL-based classifier as the first stage was used to find two most probable classes and the SPD classifier was then used to make the final decision. An accuracy of 95.2% was obtained. Instead of orientation image, coefficients of orientation model were used for fingerprint classification in [23]. However, some fingerprints from different classes have very similar orientation image except singularities region. Orientation image was then combined with singularities by some learning techniques [23,16]. Comparing to these related methods, we can see that the proposed method achieves 95.9% on five-class problem and 97.2% on four-classes problem, and it outperforms other published methods in both five- and four-class problems. we make the most of the features, and the proposed method has the following features:

(1) Accurate orientation image is necessary to improve fingerprint classification. A regularized orientation model is proposed to improve orientation extraction.

(2) Complex filter responses are used to capture the singularities information. They provide singularities, certainties and angles. The most biggest advantage is that it does not need to make the binary decision that it is a singularity or not because singularity detection itself is a challenging problem. The combination of the complex filter responses and orientation image is effective to characterize the difference between different classes.

(3) There are five stages in the proposed classification algorithm. In the first stage, the complex filter responses are utilized to decide whether it is an Arch or not. It is robust to the noise. In the second stage, singularities and ridge line flows are utilized to

distinguish Whorl from other classes, which is robust to the large intra-class variabilities of Whorl. In the fourth stage, ridge line classifier distinguish Right or Left from the other one when two classes obtained from the third stage include Right or Left, which deals with the small inter-class variabilities.

4.2.4. Computational complexity

The implementation of the proposed algorithm consists of two parts: orientation extraction and classification. The orientation

extraction is implemented in C/C++ while classification is implemented in Matlab. The average orientation extraction time over FVC-onGoing benchmark of the proposed method is 581 ms while method FOMFE [30] is 1996 ms and method Adaptive-3 [30] is 4772 ms. Our feature extraction method is much faster than other methods. The experiments over NIST 4 are conducted on the same PC with an Intel Pentium 4 processor 3.4 GHz under the Windows XP professional operating system. The average orientation extraction time over NIST 4 is about 0.88 s and the average classification

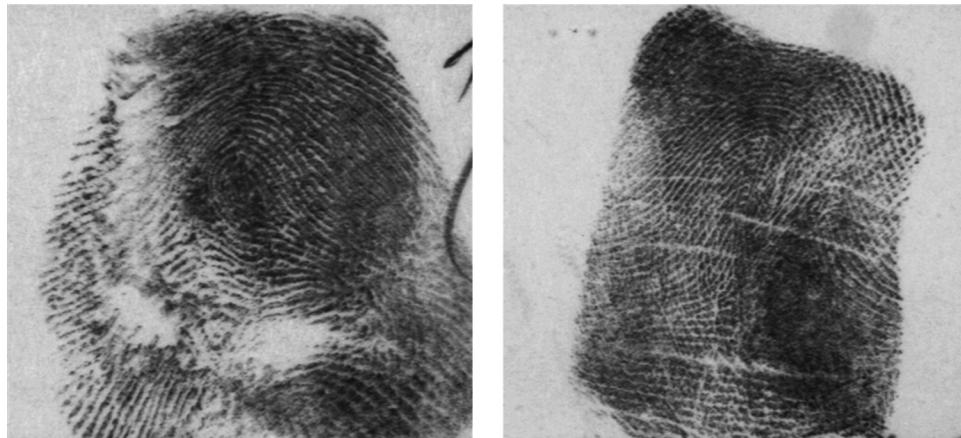


Fig. 10. Examples of low quality fingerprints.

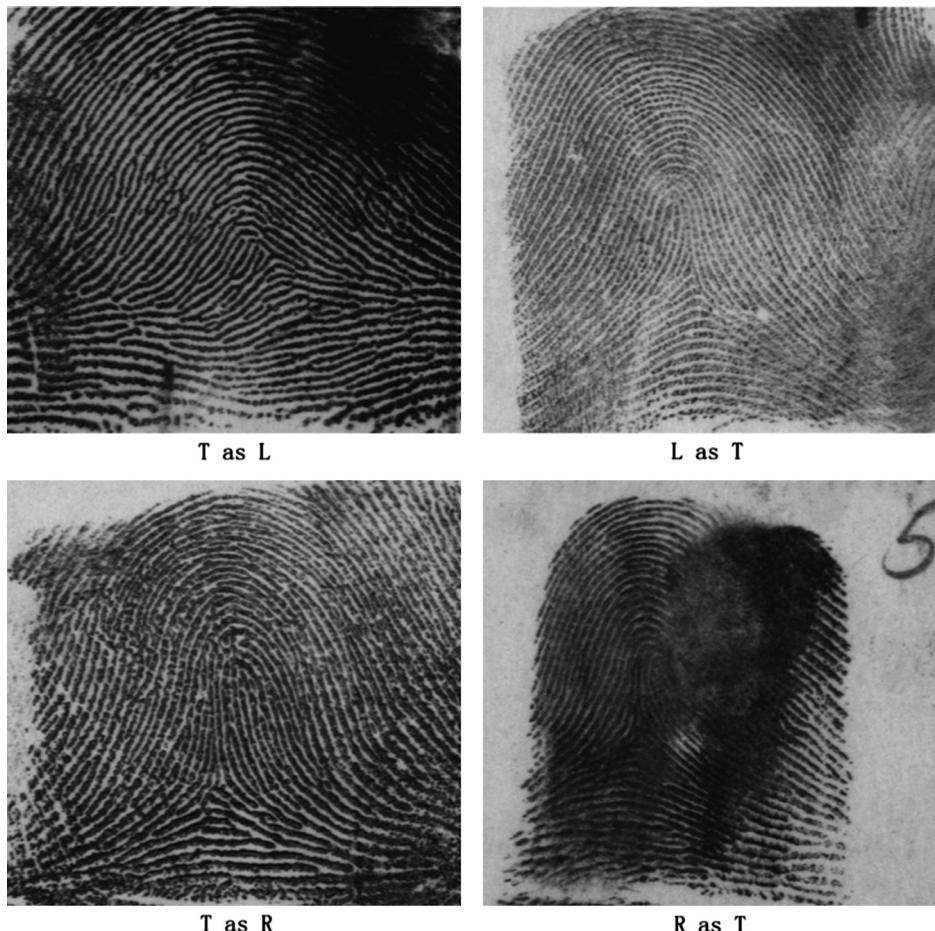


Fig. 11. Examples of misclassified fingerprints.

time is 3.43 s. In a C/C++ implementation, this classification would be much faster than in Matlab.

5. Conclusion

In this paper, a novel fingerprint classification algorithm is presented. There are mainly three contributions in this paper. Firstly, a regularized orientation model is proposed to improve fingerprint orientation extraction. The experimental results over FVC-onGoing orientation extraction benchmark demonstrate its effectiveness. Secondly, orientation image, complex filter responses as well as ridge line flows are combined to represent a fingerprint, which characterizes both ridge flow and singularities. Thirdly, a heuristic classifier is proposed for the fingerprint classification, which is robust to large intra-class variabilities and small inter-class variabilities. The experimental results over NIST SD 4 demonstrate the effectiveness of the proposed classification algorithm.

In summary, the proposed algorithm is able to make use of orientation image and complex filter response to achieve a better accuracy. Most of misclassified fingerprints are with severe noise because it is very difficult to extract accurate orientation. Fig. 10 shows some examples. In addition, some Left or Right fingerprints are mistaken as Tented Arch fingerprints and some Tented Arch fingerprints are mistaken as Left or Right fingerprints. Fig. 11 illustrates such cases. In future, elaborate classification strategy is needed to distinguish Tented Arch fingerprints from Left and Right fingerprints.

Conflict of interest

None declared.

References

- [1] D. Maltoni, D. Maio, A. Jain, S. Prabhakar, *Handbook of Fingerprint Recognition*, second edition, Springer, London, 2009.
- [2] E. Henry, *Classification and Uses of Finger Prints*, Routledge, London, 1900.
- [3] A. Jain, S. Prabhakar, L. Hong, A multichannel approach to fingerprint classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21 (4) (1999) 348–359.
- [4] M. Kawagoe, A. Tojo, Fingerprint pattern classification, *Pattern Recognition* 17 (3) (1984) 295–303.
- [5] K. Nilsson, J. Bigun, Localization of corresponding points in fingerprints by complex filtering, *Pattern Recognition Letters* 24 (13) (2003) 2135–2144.
- [6] J. Zhou, F. Chen, J. Gu, A novel algorithm for detecting singular points from fingerprint images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (7) (2009) 1239–1250.
- [7] K. Karu, A.K. Jain, Fingerprint classification, *Pattern Recognition* 29 (3) (1996) 389–404.
- [8] N.K. Ratha, K. Karu, S. Chen, A.K. Jain, A real-time matching system for large fingerprint databases, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18 (1996) 799–813.
- [9] X. Wang, M. Xie, Fingerprint classification: an approach based on singularities and analysis of fingerprint structure, in: D. Zhang, A. Jain (Eds.), *Biometric Authentication*, Lecture Notes in Computer Science, vol. 3072, Springer, Berlin/Heidelberg, 2004, pp. 1–9.
- [10] M. Liu, Fingerprint classification based on adaboost learning from singularity features, *Pattern Recognition* 43 (3) (2010) 1062–1070.
- [11] 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.
- [12] J.-H. Chang, K.-C. Fan, A new model for fingerprint classification by ridge distribution sequences, *Pattern Recognition* 35 (6) (2002) 1209–1223.
- [13] S.C. Dass, A.K. Jain, Fingerprint classification using orientation field flow curves, in: Proceedings of the ICVGIP, 2004, pp. 650–655.
- [14] L. Wang, M. Dai, Application of a new type of singular points in fingerprint classification, *Pattern Recognition Letters* 28 (2007) 1640–1650.
- [15] Q. Zhang, H. Yan, Fingerprint classification based on extraction and analysis of singularities and pseudo ridges, *Pattern Recognition* 37 (11) (2004) 2233–2243.
- [16] 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 (2008) 662–671.
- [17] R. Cappelli, D. Maio, D. Maltoni, Fingerprint classification based on multi-space kl, in: *Workshop on Automatic Identification Advances Technologies (AutoID'99)*, 1999, pp. 117–120.
- [18] 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.
- [19] C.H. Park, H. Park, Fingerprint classification using fast fourier transform and nonlinear discriminant analysis, *Pattern Recognition* 38 (4) (2005) 495–503.
- [20] C. Wilson, G. Candela, C. Watson, Neural network fingerprint classification, *Journal of Artificial Neural Networks* 1 (2) (1994) 203–228.
- [21] 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.
- [22] J. Bowen, The home office automatic fingerprint pattern classification project, in: *Proceedings of the IEE Colloquium on Neural Networks for Image Processing Applications*, 1992.
- [23] J. Li, Wei-Yun Yau, Han Wang, Combining singular points and orientation image information for fingerprint classification, *Pattern Recognition* 41 (2008) 353–366.
- [24] K.A. Nagaty, Fingerprints classification using artificial neural networks: a combined structural and statistical approach, *Neural Networks* 14 (2001) 1293–1305.
- [25] S. Bernard, N. Boujemaa, D. Vitale, C. Bricot, Fingerprint classification using kohonen topologic map, in: *Proceedings of the International Conference on Image Processing*, vol. 3, 2001, pp. 230–233.
- [26] G. Halici, U. Ongun, Fingerprint classification through self-organizing feature maps modified to treat uncertainties, *Proceedings of the IEEE* 84 (1996) 1497–1512.
- [27] N. Cristianini, J.S. Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, 2000.
- [28] 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 (2003) 397–406.
- [29] S. Shah, P.S. Sastry, Fingerprint classification using a feedback-based line detector, *IEEE Transactions on Systems, Man and Cybernetics, Part B* 34 (1) (2004) 85–94.
- [30] FVC-onGoing web site, <<https://biolab.cs.unibo.it/fvcongoing>>.
- [31] NIST Special Database 4, <<http://www.nist.gov/srd/nistsd4.cfm>>.
- [32] C. Watson, G. Candela, P. Grother, Comparison of fft fingerprint filtering methods for neural network classification, *NISTIR 5493*.
- [33] A. Bazen, S. 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.
- [34] B.K.P. Horn, B.G. Schunck, Determining optical flow, *Artificial Intelligence* 17 (1981) 185–203.
- [35] C. Xu, J.L. Prince, Snakes, shapes, and gradient vector flow, *IEEE Transactions on Image Processing* 7 (3) (1998) 359–369.
- [36] S. Ram, H. Bischof, J. Birchbauer, Curvature preserving fingerprint ridge orientation smoothing using Legendre polynomials, in: *Computer Vision and Pattern Recognition Workshops, 2008. IEEE Computer Society Conference on CVPRW 08*, 2008, pp. 1–8.
- [37] D. Tschumperle, R. Deriche, Diffusion pdes on vector-valued images, *IEEE Signal Processing Magazine* 19 (5) (2002) 16–25.
- [38] J. Bigun, Recognition of local symmetries in gray value images by harmonic functions, in: *9th International Conference on Pattern Recognition*, vol. 1, 1988, pp. 345–347.
- [39] J. Bigun, Pattern recognition in images by symmetries and coordinate transformations, *Computer Vision and Image Understanding* 68 (3) (1997) 290–307.
- [40] S. Chikkerur, N. Ratha, Impact of singular point detection on fingerprint matching performance, in: *Proceedings of the Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, 2005, pp. 207–212.
- [41] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, second edition, Academic Press, 1990.
- [42] F. Turroni, D. Maltoni, R. Cappelli, D. Maio, Improving fingerprint orientation extraction, *IEEE Transactions on Information Forensics and Security* 6 (3–2) (2011) 1002–1013.
- [43] Y. Wang, J. Hu, D. Phillips, A fingerprint orientation model based on 2d fourier expansion and its application to singular-point detection and fingerprint indexing, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (4) (2007) 573–585.

Liaojun Pang received his B.S. and M.S. degrees in Computer Science and Technology from Xidian University of China, in 2000 and 2003, respectively. In 2006, he received the Ph.D. degree in Cryptography from Xidian University of China. Now, he is with the School of Life Sciences and Technology, Xidian University. His research interests include Internet security, cryptography, secure mobile agent system and e-commerce security technology. He became a Member (M) of IEEE, in 2009.

Jimin Liang received the B.S., M.S., and Ph.D. degrees from Xidian University, Xi'an, China, in 1992, 1995, 2000, respectively, all majored in electronic engineering. In 1995, he joined Xidian University, where he is currently a Professor in the Life Science Research Center (LSRC), School of Life Sciences and Technology. During 2002, he was a Research Associate Professor in the Department of Electrical and Computer Engineering, University of Tennessee, Knoxville. His current research interests include image processing, image analysis, biometric recognition and biometric encryption. He became a member of IEEE since 2008.

Jie Tian received the Ph.D. degree (with honors) in artificial intelligence from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1992. From 1994 to 1996, he was a Postdoctoral Fellow with the Medical Image Processing Group, University of Pennsylvania, Philadelphia. Since 1997, he has been a Professor at the Institute of Automation, Chinese Academy of Sciences. Since 2007, he has also been the Chair Professor of the Cheung Kong Scholars at Xidian University, Xi'an, China. His current research interests include pattern recognition, machine learning, image processing and their applications in biometrics, etc.