Efficient Rectification of Distorted Fingerprints

Shan Gu, *Student Member, IEEE*, Jianjiang Feng, *Member, IEEE*, Jiwen Lu, *Senior Member, IEEE*, and Jie Zhou, *Senior Member, IEEE*

***Abstract*— Recently, distortion rectification based on a single fingerprint image has been shown to be able to significantly improve the recognition rate of distorted fingerprints. However, the computational complexity of such a method is too high to be useful in practice. In this paper, we propose a novel method for the rectification of distorted fingerprints, whose speed is over 30 times faster than the existing method. This significant speedup is due to a Hough-forest-based two-step fingerprint pose estimation algorithm and a support vector regressor-based fingerprint distortion field estimation algorithm. Experimental results on public domain databases show that our method can achieve as good rectification performance as the existing method but meanwhile is significantly faster.**

***Index Terms*— Distortion rectification, support vector regression, pose estimation, Hough forest.**

I. Introduction

FINGERPRINT recognition was originally used for criminal investigation and has gradually extended to applications such as border control, computer logon, and mobile payment [1]. One of the main reasons for the widespread adoption of fingerprint recognition techniques is that the error rates of state-of-the-art fingerprint recognition algorithms are very low on fingerprints of high or medium quality [2].

However, recognition rate for low quality fingerprints is still far from satisfactory, and low quality fingerprints are not uncommon [1], [3]. One type of low quality fingerprints are distorted fingerprints, which are usually caused when users press their fingerprints on sensors improperly. The recognition rate on distorted fingerprints is still low with most existing fingerprint recognition algorithms [4]. This limitation has different influence on different types of biometric systems. In negative recognition systems, people in the watch-list may purposely distort their fingerprints to avoid identification [5]. In positive recognition systems, users may need to press their fingerprints many times to get verified, and that may reduce the user experience.

The idea of distortion rectification [6] is to transform a distorted fingerprint into a normal one so as to increase the recognition rate of existing fingerprint recognition algorithms for distorted fingerprints. A recent study shows that distortion rectification can effectively complete this task. However, this method is very slow and thus is difficult to find real applications [7].

In this paper, we present an efficient distortion rectification method, whose speed is significantly faster than the method of Si *et al.* [7]. Given an input fingerprint, the fingerprint should be registered firstly by pose estimation, and then distortion detection is performed. If it is detected as normal, the original fingerprint is directly returned; otherwise, the distortion field is estimated, and rectification is performed to transform the distorted fingerprint into a normal one. The whole flowchart is given in Fig. 1.

Although the overall flowchart is similar to [7], the proposed algorithm is significantly faster because of two novel methods for pose estimation and distortion field estimation:

1) Pose estimation consists of two successive steps: Hough forest [8] based center estimation is first performed and followed by regression based direction estimation. Such a two-step estimation method reduces the search space of pose significantly and is much faster than the multi-reference based pose estimation algorithm in [7].
2) Thanks to the accurate pose estimation algorithm, the distortion field can be estimated directly from feature vectors extracted from registered fingerprints with a support vector regressor, which is much more efficient than the time-consuming nearest neighbor search step in [7].

We conducted extensive experiments on public databases, including Tsinghua distorted fingerprint database (Tsinghua DF), FVC2004 DB1, and FVC2006 DB2_A. The results show that the proposed method can well rectify distorted fingerprints and improve the matching performance of existing fingerprint matcher. Meanwhile, the speed of this method is over 30 times faster than the method in [7].

The following sections of this paper are organized as follows. In Section 2, we review researches related to distorted fingerprints. In Section 3, we introduce how to estimate the pose information of fingerprints. In Section 4, we present the details of the distorted fingerprint detection and rectification.

Manuscript received March 24, 2017; revised July 5, 2017; accepted August 11, 2017. Date of publication August 29, 2017; date of current version November 22, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61622207, Grant 61373074, and Grant 61527808, and in part by the Shenzhen Fundamental Research Fund (subject arrangement) under Grant JCYJ20170412170438636. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Domingo Mery. *(Corresponding author: Jianjiang Feng.)*

S. Gu, J. Lu, and J. Zhou are with the Department of Automation, Tsinghua University, Beijing 100084, China, and also with the Tsinghua National Laboratory for Information Science and Technology, Beijing 100084, China (e-mail: gus16@mails.tsinghua.edu.cn; lujiwen@tsinghua.edu.cn; jzhou@tsinghua.edu.cn).

J. Feng is with the Department of Automation, Tsinghua University, Beijing 100084, China, also with the Tsinghua National Laboratory for Information Science and Technology, Beijing 100084, China, and also with the Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China (e-mail: jfeng@tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2017.2745685

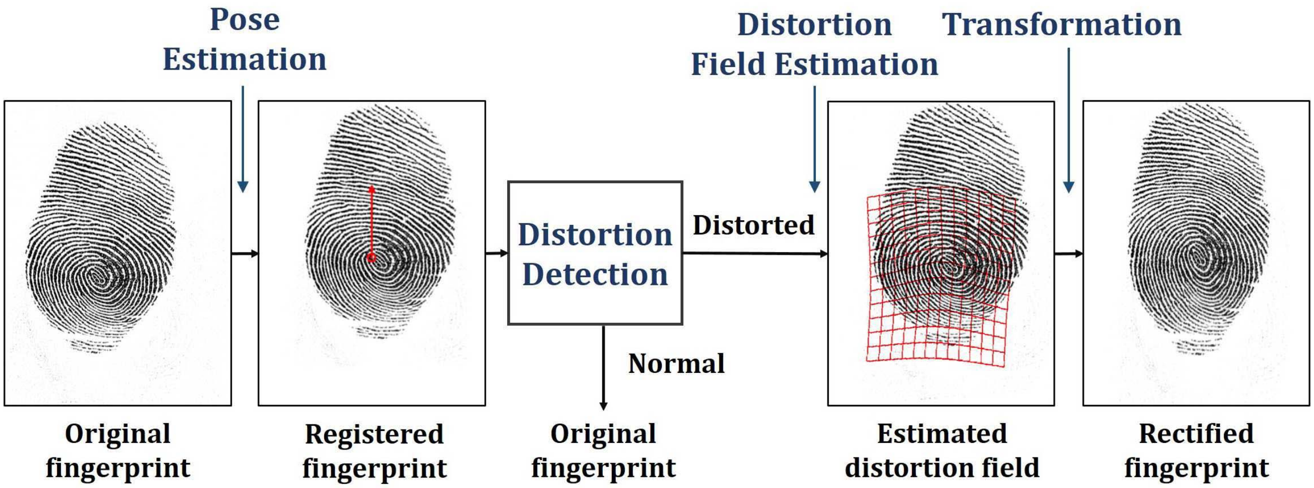


Fig. 1. Flowchart of the proposed distorted fingerprint rectification system.

In Section 5, we describe the experiments and evaluate the rectification performance of the proposed method. In Section 6, we summarize the study and discuss possible future research.

II. Related Work

There exist several types of methods to overcome the limitations of automatic fingerprint recognition systems in recognizing distorted fingerprints, including fingerprint matching with distortion tolerated, distorted fingerprint detection using special sensors, and distortion rectification.

Many researchers [9]–[14] modify fingerprint matching algorithms to tolerate a certain degree of distortion, such that fingerprints with little distortion can match well with the mated fingerprints. Most approaches [9]–[12] are based on minutiae matching because minutiae are believed to be discriminating and reliable. Common methods dealing with distortion include: globally model the transformation by rigid transformation [9], [10] or thin plate spline [11] to compensate for distortion, and locally constrain the distortion [12]. Matching approaches based on image [13] or ridge skeleton [14] also employ certain distortion tolerant strategies. However, these methods may produce a high matching score for non-mated fingerprints, thus may increase the false match rate. Another drawback of these methods is that distortion tolerant matching strategy often decreases the matching speed.

Some researchers [15]–[18] proposed hardware solutions for distortion detection when users are pressing fingerprints. Distorted fingerprints would be rejected when detected, and the user may be asked to press fingerprints again. This type of method has two limitations: (1) hardware replacement is usually hard to implement and/or expensive, and (2) it cannot deal with existing distorted fingerprints in databases.

Distortion rectification aims to transform distorted fingerprints into normal fingerprints based on a single fingerprint image. This third type of method has two advantages: (1) it requires no changes to existing automatic fingerprint recognition systems; (2) it can be applied to existing distorted fingerprints. However, there are very few studies on this direction.

Senior and Bolle [6] proposed the first distortion rectification method based on the assumption that ridges of normal fingerprints are distributed uniformly. They normalize the ridge density of the input fingerprint into a fixed value to remove the distortion. However, their assumption is incorrect as ridge density of a fingerprint varies significantly across different regions. Meanwhile, this method does not detect whether or not the input fingerprint is distorted and is simply applied to all input fingerprints. In this way, normal fingerprints may lose ridge density information, and the matching performance may decrease.

The method of Si *et al.* [7] consists of distortion detection and distortion rectification. Fingerprint center and direction are first estimated by combing singularity detection with multi-reference based alignment method. Then the distortion detection is taken as a classification problem solved by support vector classification. For a fingerprint detected as distorted, the reference fingerprint with most similar feature map is retrieved from a distorted reference fingerprint database by the nearest neighbor approach. Then the distorted input fingerprint is rectified into a normal one with the distortion field of the retrieved reference fingerprint. Experimental results show that this method is much more effective than the method proposed by Senior and Bolle [6]. But this method is very time-consuming due to the need to align and compare the input image with all the reference images both in distortion detection and rectification process.

III. Fingerprint Pose Estimation

The proposed pose estimation algorithm consists of two steps: finger center estimation and finger direction estimation. To register a fingerprint, the center of a fingerprint is estimated firstly by a Hough forest based approach. After the center is moved to the origin of a fixed coordinate system, the finger direction is then estimated by a support vector regression method. Both steps use local ridge orientation and period information, which are extracted using the method in [19]. The flowchart of the pose estimation process is shown in Fig. 2.

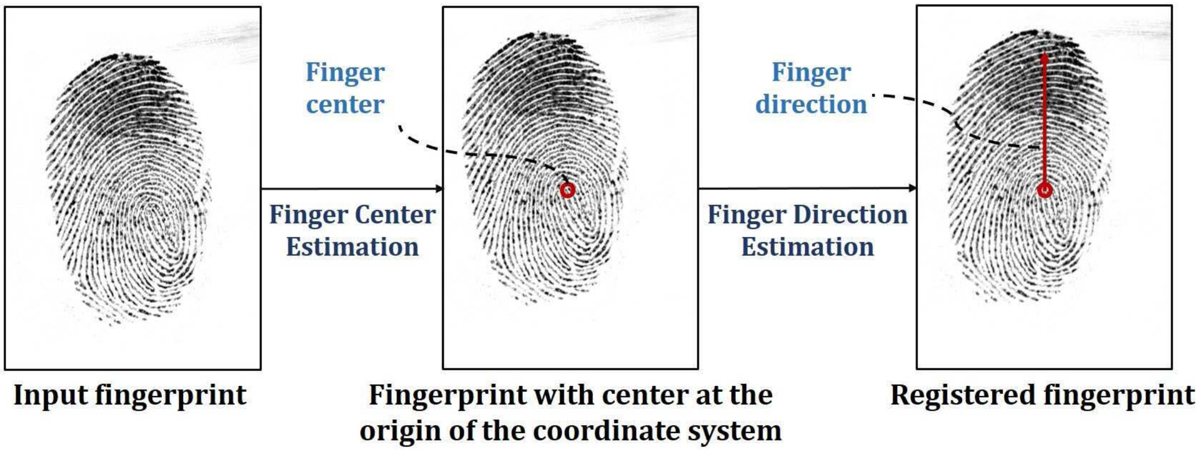


Fig. 2. Flowchart of the proposed pose estimation algorithm.

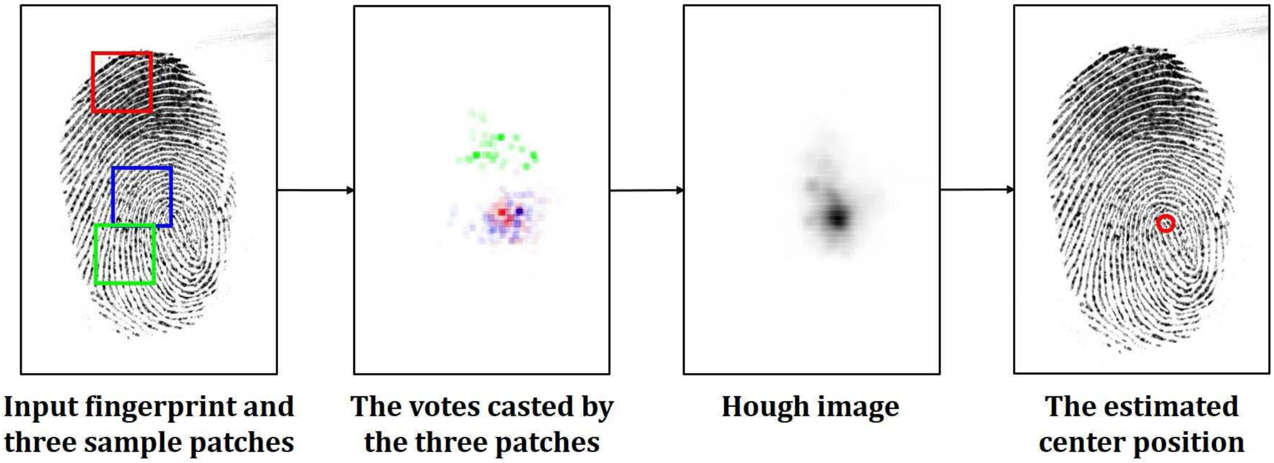

Input fingerprint and three sample patches
The votes casted by the three patches
Hough image
The estimated center position

Fig. 3. Flowchart of the proposed center estimation algorithm.

In the following part, we introduce the details of the approach of center estimation and finger direction estimation.

A. Finger Center Estimation

We propose a Hough forest based method to estimate the finger center. Hough forest is a classical method of detecting the position of an object in the computer vision and has attracted a lot of attention [8]. It is a Hough transform based approach, where features extracted from local image patches can cast probabilistic votes for the possible location of the center of an object.

For an input fingerprint, we firstly sample its ridge orientation and period maps to obtain feature patches and extract the feature vector from each patch. Then the feature patches go across each tree in the Hough forest and finally reach the leaf nodes. The offset vector information stored in each leaf node would cast probabilistic votes to the probable position of the center. We can then obtain the Hough image by adding the probabilistic votes from all patches and estimate the fingerprint center with it. The whole flowchart of fingerprint center is shown in Fig. 3.

In the following, we describe how to train the Hough forest and how to estimate the center of a fingerprint with the Hough image. In the forest training part, we describe the details of patch sampling, feature extraction, and tree construction.

1) Hough Forest Training: Hough forest is a Hough based random forest. Thus its realization is similar to that of a random forest. We choose image patches uniformly sampled in training images as the training subset. Each patch corresponds to a feature vector $\boldsymbol{x}$, a patch label c, and an offset vector $\boldsymbol{d}$.

The binary patch label c represents whether the patch is near the center of the fingerprint. $c = 1$ when the horizontal and vertical distances between the center of the patch and that of the fingerprint are both less than 3 times of the patch size (patch size $=$ 40 pixels). The patch label is not used to discriminate the fingerprint area and the background, because patches far away from the finger center may provide inaccurate information. The offset vector $\boldsymbol{d}$ represents the offset vector between the center of a local patch and the finger center.

The random forest is trained under the supervision of patch labels c and offset vectors $\boldsymbol{d}$. When training each tree, we randomly select features for judging at each non-leaf node. According to the judging result, each sample chooses to go to

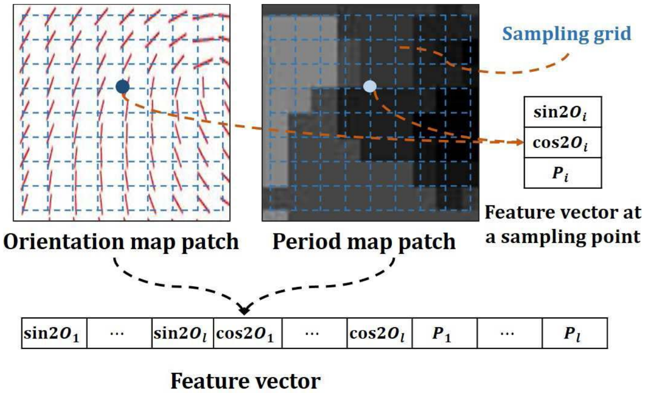

Sampling grid
sin2O_i
cos2O_i
P_i
Feature vector at a sampling point
Orientation map patch
Period map patch
sin2O_1 ··· sin2O_l cos2O_1 ··· cos2O_l P_1 ··· P_l
Feature vector

Fig. 4. The sampling grid for extracting the feature vectors covers the whole image patches extracted from the ridge orientation map and period map. The feature vector extracted from the two patches is defined as $\{[\sin\ (2O_i)\ \cos(2O_i)\ P_i]\}$ where O_i represents the orientation of the sampling point and P_i represents the period of the sampling point. l represents the number of the sampling points in two patches.

the left child or the right child of the node until it reaches the leaf node. A constructed node is declared as a leaf node when the depth of the node reaches a maximum ($\text{depth}_{\max} = 15$) or the number of the samples to reach the node is equal to the minimum ($\text{num}_{\max} = 20$). The leaf node stores the information of the samples which reach the node, including the proportion of the samples whose patch label is 1 and their offset vectors. After the forest is constructed, each test sample goes across each tree in the same way as during the training. The information stored in the node that the input sample finally arrives at casts the votes for the probable position of the center.

a) Patch sampling: A total of 400 manually registered fingerprints are chosen as training images in the training subset of Tsinghua DF database [7], which is publicly available at http://ivg.au.tsinghua.edu.cn/dataset/TDFD.php.

For each training fingerprint, we sample uniformly in the feature figures to obtain the training patches. The interval of the sampling grid is 40 pixels, and the size of a local patch is 80 × 80 pixels. To avoid sampling too many patches in the background, sampling is limited only in the fingerprint area. Patches in the area below the center are not used for training, because the lower area of the orientation map of a fingerprint is similar to the upper area while their orientation of the offset vector is different, which may mislead the center estimation.

b) Feature selection: To increase the efficiency of training and testing, feature vectors sampled in each local feature patch is used as the training input, which is shown in Fig. 4. The interval of sampling grid is 16 pixels. In this way, we can get a $3l$ dimensional feature vector $\{[\sin(2O_i)\ \cos(2\ O_i)\ P_i]\}$ where O_i and P_i represent the local ridge orientation and period at the ith sampling point, and l is the number of the sampling points in an image patch.

c) Tree construction: The process of the construction of a tree in the Hough forest is similar to that in [8].

At each non-leaf node in a tree, the training sample chooses to go to the left or the right child according to a binary judgment. Here, we compare the value of the feature vector for the judgment. Let x_p and x_q denote two features in the feature vector, and τ be the comparison threshold, the judgment is: if $x_p - x_q > \tau$ or not. Here, the comparison threshold τ is chosen in the optional range to maximize the purity of the two kinds of the samples after split.

Two methods under the supervision of patch labels and offset vectors are used for purity quantization. As for the purity of patch labels, we use the fundamental entropy based measurement method:

$$U_c = -c\log c - (1-c)\log(1-c). \tag{1}$$

The purity of the offset vectors is defined as:

$$U_d = \sum_{i:c_i=1} (d_i - \overline{d})^2 \tag{2}$$

where the $\overline{d}$ represents the mean of all the offset vectors, and patches in background whose label $c_i = 0$ are neglected. The purity of the splitting is calculated by combining the two methods above:

$$U = U_c + U_d. \tag{3}$$

In consideration of decreasing both the purity of the classification label and the offset vector, all the samples which arrive at the leaf node have a small degree of uncertainty.

2) Center Estimation With Forest Testing: Given an input fingerprint, the test feature patches and the test feature vectors are both sampled in the same manner as at the training time. Due to the unknown position of center, patches are sampling in the upper 3/5 fingerprint area to avoid sampling many in the lower area. Each test sample goes across each tree in the Hough forest and finally reaches a leaf node. The Hough image is then obtained by adding the probabilistic votes to the probable position of the center predicted by the offset vectors stored in the leaf node. The position of the maximum of the Hough image is then seen as the estimated center of the test fingerprint.

Usually, the closer to the real center the patch is, the more reliable vote it will provide. In fact, it is difficult to find the image patches near the real center to cast reliable votes. Thus here two iterative calculations are used. The processes of the two calculations are nearly the same, which is shown in Fig. 3, but the method of obtaining the input local patches and how to use the Hough image are different.

In the first calculation, the whole fingerprint is sampled to get the input local patches. As the local patches near the real center are expected as the input for the next calculation, the points that the first ten maximum values in the Hough image are at are selected as the sample points in the second calculation. Generally, most of the ten points are near the real center though the position of the maximum is not the accurate estimation. Therefore, it is possible to select the patches near the real center to cast votes to achieve a more accurate result after the first calculation (See Fig. 5 for an example). In the second calculation, the new input is chosen from the sample points achieved before, and the position of maximum value in the Hough image is chosen as the estimated center of the input fingerprint.

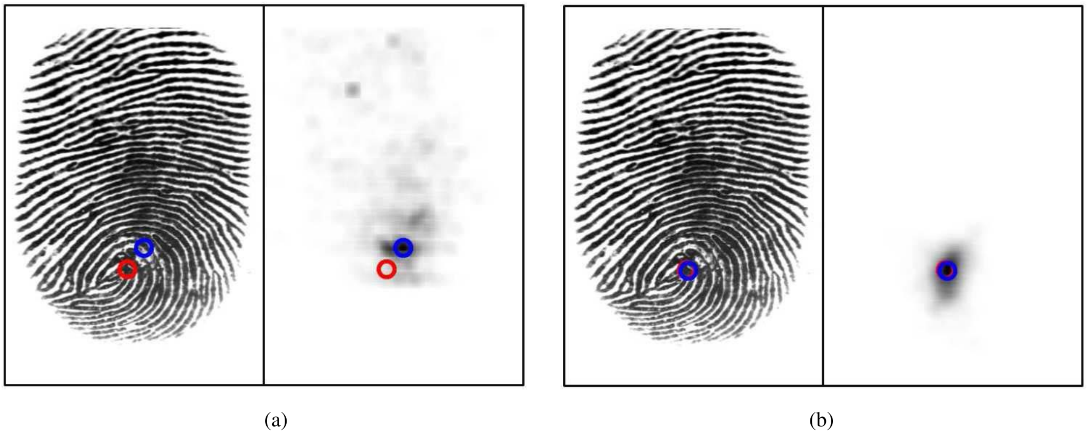


Fig. 5. The estimated center and the Hough image in two iterations. After the first iteration, the vote distribution in the Hough image is not concentrated and thus the estimated center (blue circle) is far away from the true center point (red circle). After the second iteration, the vote distribution in the Hough image is more concentrated and the center estimation is more accurate. (a) The first iteration. (b) The second iteration.

B. Finger Direction Estimation

The finger direction is estimated by the support vector regression after the center is estimated. It is defined by the direction which is vertical to the finger joint. When given an input image, we just need to estimate the rotation angle between the finger direction and the vertically upward direction.

To generate the training data, a total of 400 manually registered fingerprints are chosen, which are the same as the training data for center estimation. The rotation angle of these registered fingerprints are regarded as 0 degree. Then we artificially add rotation to the 400 registered images, and the rotation angle is from -30 degrees to 30 degrees by 5 degrees. In this way, each fingerprint image can generate 12 more rotated images. All the 5200 images are used for the regressor training, and their corresponding rotation angles are taken as the training targets. In contrast to the discrete rotation angles estimated by the method of Si *et al.* [7], the angles estimated here are continuous, which is more in line with the real situation.

The feature vector is obtained by sampling only the orientation map, as the orientation map provides main information on the rotation angle. The sampling grid covers the whole fingerprint and the sampling interval is 16×16 pixels. The feature vector is defined as $\{[\sin(2\ O_i)\ \cos(2\ O_i)]\}$, where O_i is the sampled orientation value.

Given the training data, the least square support vector method is used to train the regressor. We use the LSSVM toolbox [20] directly, and the parameters are set to RBF kernel, regularization parameter $= 10$, and kernel function parameter $= 100$. Fig. 6 shows the estimated poses of six fingerprints with different pattern types. Quantitative evaluation on large databases will be presented in the experiment section.

IV. Distortion Detection and Rectification

Distorted fingerprint detection is taken as a two-class classification problem, which is solved by support vector classification. Distortion rectification, of which the key is distortion field estimation, is taken as a regression problem, which is solved by support vector regression. With the estimated distortion field, rectified fingerprints can be obtained by reverse image transformation. In the following part, we introduce how to select the features, detect distorted fingerprints, estimate the distortion field, and perform image transformation.

A. Feature Selection

We use the same feature vectors in distortion detection and rectification. Features are uniformly sampled in the ridge period map and orientation map of the registered fingerprint. The sampling interval is 16×16 pixels in both maps.

Similar to [7], the sampling grid in the orientation map covers only the area above the center because the upper area of the orientation map is similar to the lower area and usually produces greater variations than the lower area if the fingerprint is distorted. The sampling grid in the period map covers all the fingerprint area. The feature vector is defined as $\{[\sin(2\ O_i)\ \cos(2\ O_i)\ P_i]\}$. It is similar to that in the procedure of pose estimation, but it is a $2\ l_1 + l_2$ dimensional vector, where l_1 and l_2 represent the number of the sampling points in the orientation map and period map, respectively.

B. Distortion Detection

A total of 500 fingerprints are used for training, which include 200 pairs of normal and distorted fingerprints from the training subset of Tsinghua DF database and 100 normal fingerprints in FVC2002 DB1_A. The distorted fingerprints are seen as positive samples, and the normal fingerprints are seen as negative samples.

LibSVM [21] is used to train a support vector classifier. All parameters used in the LibSVM are default ones. Specifically, the RBF kernel rather than the quadratic polynomial kernel is used.

C. Distortion Rectification

1) Regression Target: The regression target of the distortion rectification is the distortion field of the distorted fingerprints. We use the coefficients on the principle components of the

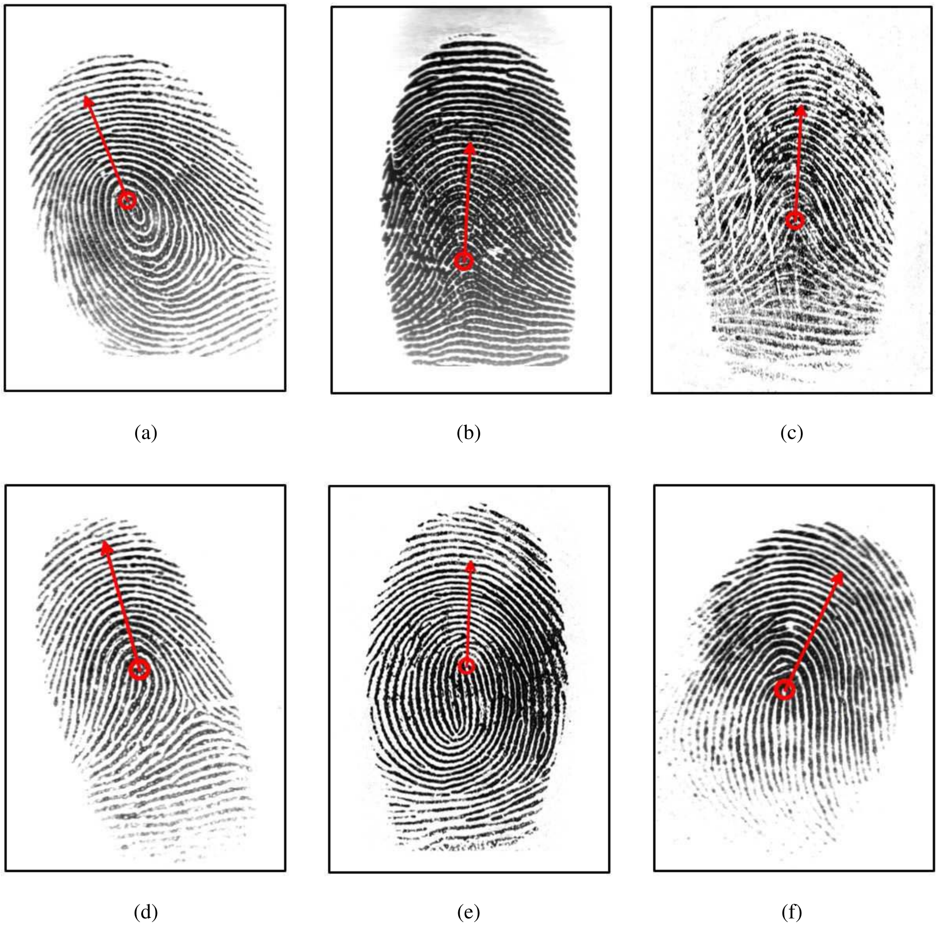


Fig. 6. Estimated poses for six fingerprints of different pattern types. The center is indicated by the circle and the finger direction is indicated by the arrow. (a) Whorl. (B) Plain arch. (c) Tented arch. (d) Left loop. (e) Twin loop. (f) Right loop.

distortion field to represent the distortion field, following the method of Si *et al.* [7]. A total of 200 pairs of normal and distorted fingerprints from the training subset of Tsinghua DF database are used to compute the distortion fields between them. The distortion field is estimated by the thin plate spline method with the minutiae pairs tracked in the process of fingerprint collection. It is defined as the difference between the grid vector of the normal fingerprint and the corresponding distorted fingerprint. Let f_i $(i = 1, \ldots, n_{\text{train}})$ denotes the distortion field of the ith pair of training fingerprints, where $n_{\text{train}} = 200$. After all the distortion field of the distorted fingerprints is calculated, the principal components are analyzed. Let e_j denotes the eigenvector and λ_j denotes the eigenvalue of the covariance matrix $cov(F)$, where F is the overall difference matrix of all the distortion fields. Then the distortion field of each fingerprint can be represented as:

$$f_i = \bar{f} + \sum_{j=1}^{T} k_j \sqrt{\lambda_j} e_j \tag{4}$$

where k_i is the coefficient on the eigenvector, T is the number of all the principle components, and $\bar{f}$ is the mean distortion field. The coefficients $\{k_i\}$ $(i = 1, \ldots, t)$ are taken as the regression target, where t is the number of the selected principle components.

To determine the value of t, we change the value from two to six and compare both the accuracy and speed of the experiments. The results show that choosing the first two principal components is the best.

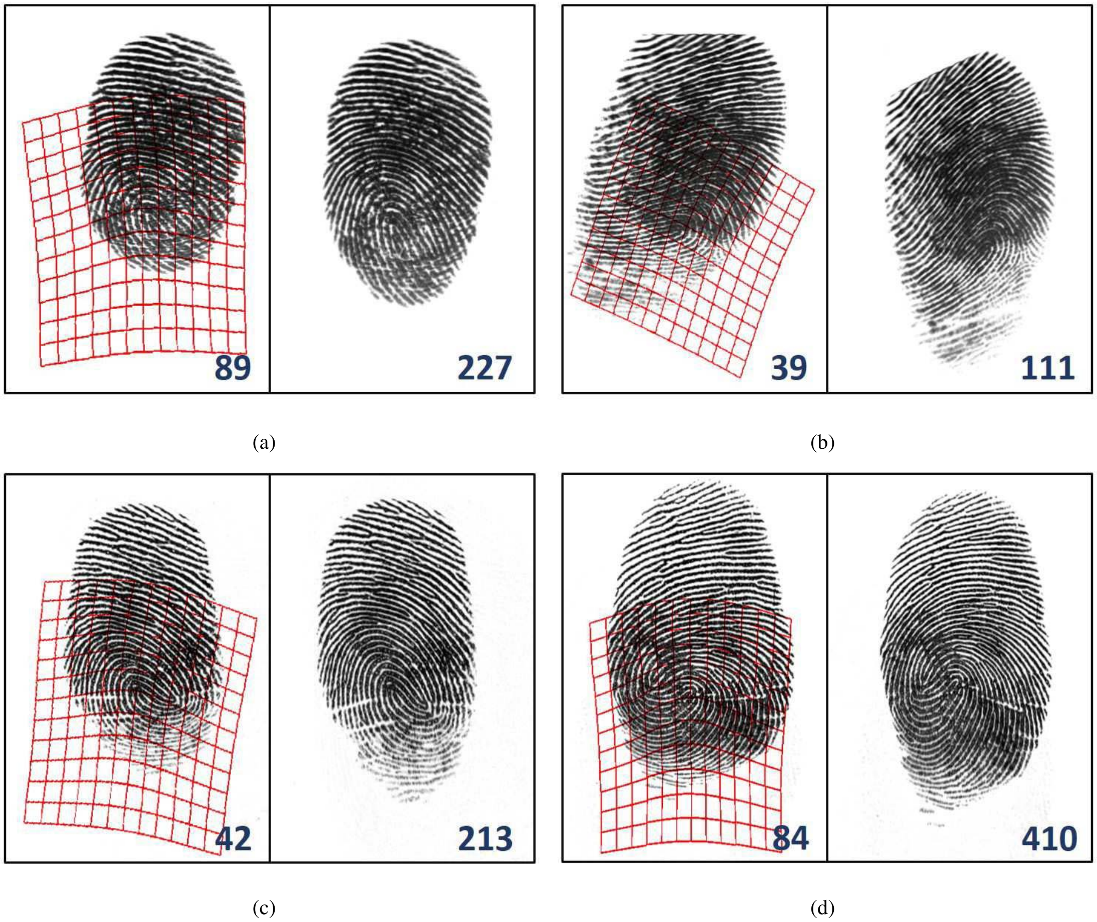
It is important to note that the coefficient here is continuous while that in the method of Si *et al.* [7] is discrete, though both methods choose the first two principal components as the approximation of the distortion field.

2) Regressor Training: After the input feature vector and the regression target are obtained, the least square support vector method is used to train the regressor. We use the LSSVM toolbox [20] directly, and the parameters are also set to RBF kernel, regularization parameter = 10, and kernel function parameter = 100.

A total of 400 fingerprint images (200 normal fingerprints and 200 distorted fingerprints) are used for training, the same as the training images in the pose estimation process. Taking normal fingerprints into account can prevent the matching scores of normal fingerprints, which are misclassified as distorted fingerprints from decreasing after rectification during the test.

D. Image Transformation

The distorted fingerprint can be transformed to a normal one with the estimated distortion field by image transformation.



(a) (b)

(c) (d)

Fig. 7. Four examples of distortion field estimation and rectification. For each example, the left are the origin fingerprints and the red transformation grids overlaid on them are the distortion fields estimated by the proposed algorithm. The right are the fingerprints after rectification. The number below each fingerprint is the VeriFinger matching score between it and its mated fingerprint.

When we obtain the estimated coefficients of the distortion field $\{k_i\}$, the estimated distortion field can be approximated with the similar calculation as Equation (4), but using only selected principle components. With the estimated coefficients of the distortion field, we can obtain a transformation grid, with which a rectified fingerprint can be obtained by the B-spline transformation.

The matching score is used for the evaluation of rectification. Fig. 7 shows four examples of distortion field estimation and the rectified fingerprints. Matching scores are computed using a commercial fingerprint recognition SDK, VeriFinger 6.2 [22]. The matching scores of the four distorted fingerprints greatly increase after rectified with the estimated distortion field. In other words, the distortion between the distorted fingerprint and its mated fingerprint is greatly reduced, because the higher matching score suggests the rectified fingerprint be more similar with its mated fingerprint, which is considered normal.

V. Experiment

In this section, we first evaluate the performance of pose estimation and then evaluate the performance of distortion rectification. The evaluation is based on experiments on three public domain fingerprint databases: FVC2004 DB1, FVC2006 DB2_A, and the testing subset of Tsinghua DF database. The details of the databases are described in Table I.

A. Performance of Fingerprint Pose Estimation

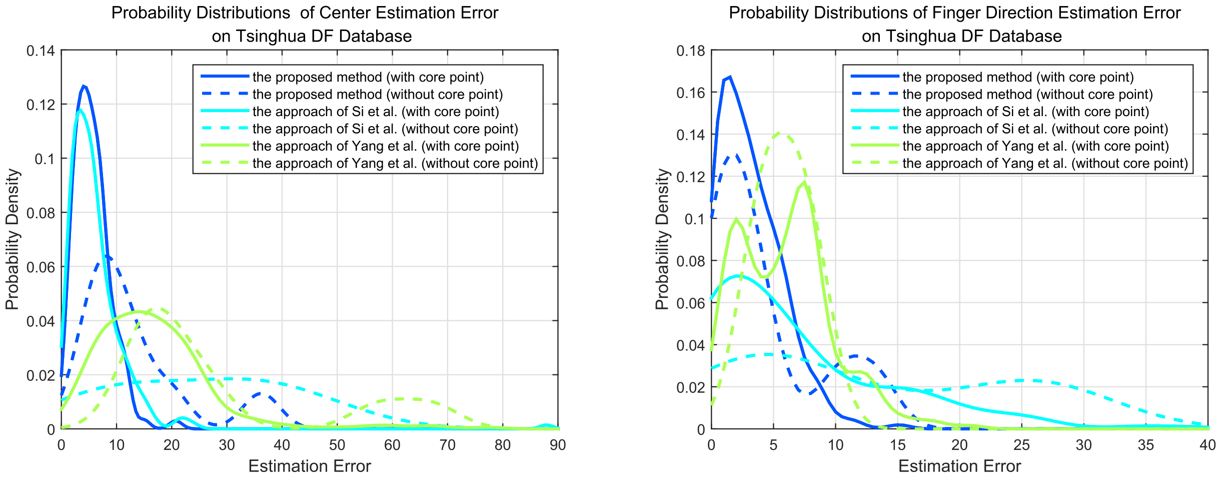
We compare the estimation errors of the proposed pose estimation approach with that of the method of Si *et al.* [7] and the method of Yang *et al.* [23] to evaluate the performance. The estimation error is computed by comparing the estimated finger center and finger direction with those manually marked.

The probability distribution of estimation errors on Tsinghua DF is shown in Fig. 8. In the method of Si *et al.* [7], the fingerprint center is estimated by the Poincaré index based algorithm [24] if the core point can be detected, otherwise is estimated by a full search. Therefore, the comparison is made for two cases whether the core point can be detected. From Fig. 8, it is clear that:

1) For the fingerprints with core point, the performance of the proposed finger center estimation is slightly better than that of the Poincaré index based algorithm and far better than the approach of Yang *et al.*

TABLE I
FINGERPRINT DATABASES USED IN THE EXPERIMENT AND MATCHING PROTOCOL

Database	Description	Genuine Match	Impostor Match
FVC2004 DB1	880 fingerprints from 110 different fingers, 8 fingerprints per finger; 89 distorted fingerprints and mated normal fingerprints compose the distorted subset	each fingerprint is matched with other fingerprints from the same finger, and the symmetric matches are avoided; $(8 \times 7)/2 \times 110 = 3080$ matches in total and 89 matches in the distorted subset	the first fingerprint of each finger is matched with the first fingerprint of other fingers, and the symmetric matches are avoided; totally $(110 \times 109)/2 = 5995$ matches
FVC2006 DB2_A	1680 fingerprints from 140 different fingers, 12 fingerprints per finger	each fingerprint is matched with other fingerprints from the same finger, and the symmetric matches are avoided; totally $(12 \times 11)/2 \times 140 = 9240$ matches	the first fingerprint of each finger is matched with the first fingerprint of other fingers,and the symmetric matches are avoided; $(140 \times 139)/2 = 9730$ matches in total
Tsinghua DF	320 pairs of normal fingerprints and mated distorted fingerprints from 185 different fingers; the first 200 pairs are used as training set, and the rest 120 pairs are used as testing set	each normal fingerprint is matched with its mated distorted fingerprint; totally 120 matches in the testing set	a normal fingerprint is matched with all distorted fingerprints except those from the same finger; totally 14252 matches in the testing set



(a) (b)

Fig. 8. The probability distributions of estimation errors of finger center and finger direction estimation error on Tsinghua DF database.

2) For the fingerprints without core point, the proposed center estimation approach produces the most accurate results among three pose estimation approaches.
3) For all the fingerprints, the proposed finger direction estimation method greatly increase the estimation accuracy.

The estimation error is analyzed in Table II. Taking into account the error of manual annotation, the estimation is deemed as wrong when the distance between the estimated center and the center manually marked is more than 15 pixels or the angle between the estimated finger direction and that manually marked is more than 10 degrees. From Table II, the proposed method produces fewer estimation error of the finger center and direction for fingerprints with and without core point.

B. Performance of Distortion Rectification

To make a quantitative and objective evaluation of distortion rectification algorithms, four fingerprint matching experiments are conducted on three public databases: FVC2004 DB1, FVC2006 DB2_A, and Tsinghua DF database. To get a more reasonable assessment, we choose all the 89 distorted fingerprints (along with 89 mated normal fingerprints) in FVC2004 DB1 database to compose a distorted subset and also do experiments on it. The input images of the four matching experiments are the original fingerprints without rectification, fingerprints rectified by the method of Si *et al.* [7], fingerprints rectified by the method of Senior and Bolle [6], and fingerprints rectified by the proposed approach. Matching scores are computed using a commercial fingerprint recognition SDK, VeriFinger 6.2. Matching protocol on three databases is given in Table I.

We firstly use the scatter plot to compare the matching scores with and without rectification (by the proposed algorithm) on the distorted subset of FVC2004 DB1 and Tsinghua DF database. As mentioned before, the matching score is an objective measurement of the performance of distortion rectification. As we can see from Fig. 9, the matching scores of most fingerprints increase after rectification (points above

TABLE II
STATISTICS OF POSE ESTIMATION ERROR ON TSINGHUA DF DATABASE

	Finger Center Estimation		Finger Direction Estimation	
	With core point (231)	**Without core point (9)**	**With core point (231)**	**Without core point (9)**
The method of Si *et al.*	9	6	53	4
The method of Yang *et al.*	164	9	27	0
The proposed method	3	2	3	2

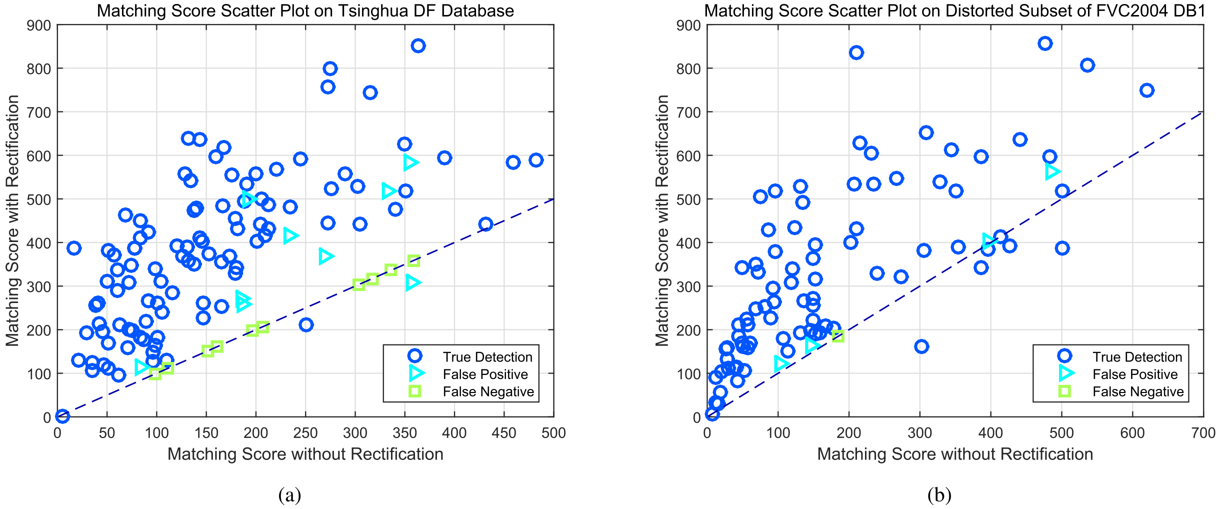


Fig. 9. The scatter plot of VeriFinger matching scores with/without rectification on two databases: (a) Tsinghua DF database and (b) distorted subset of FVC2004 DB1.

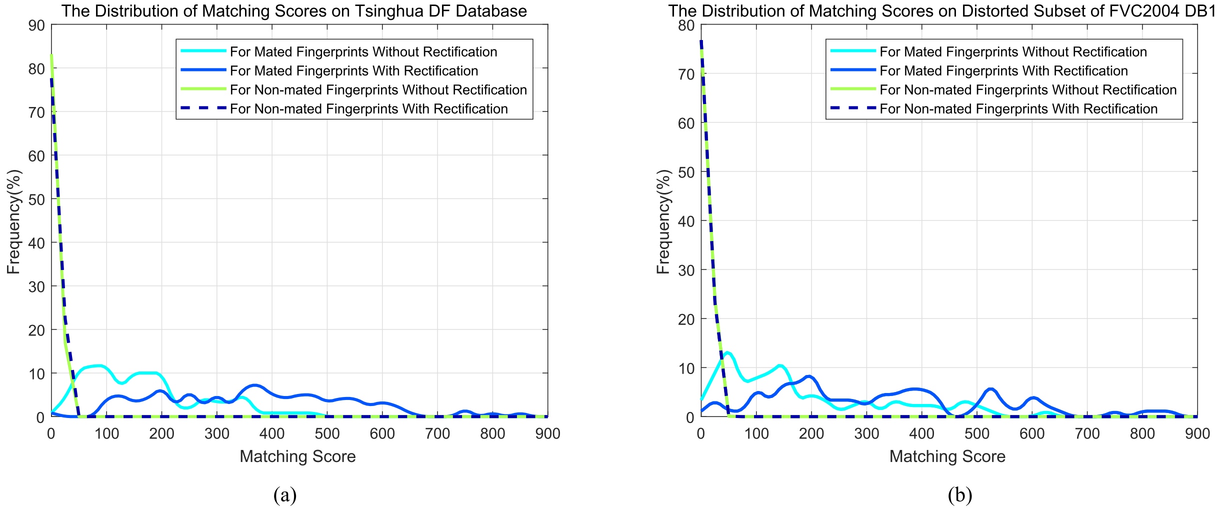
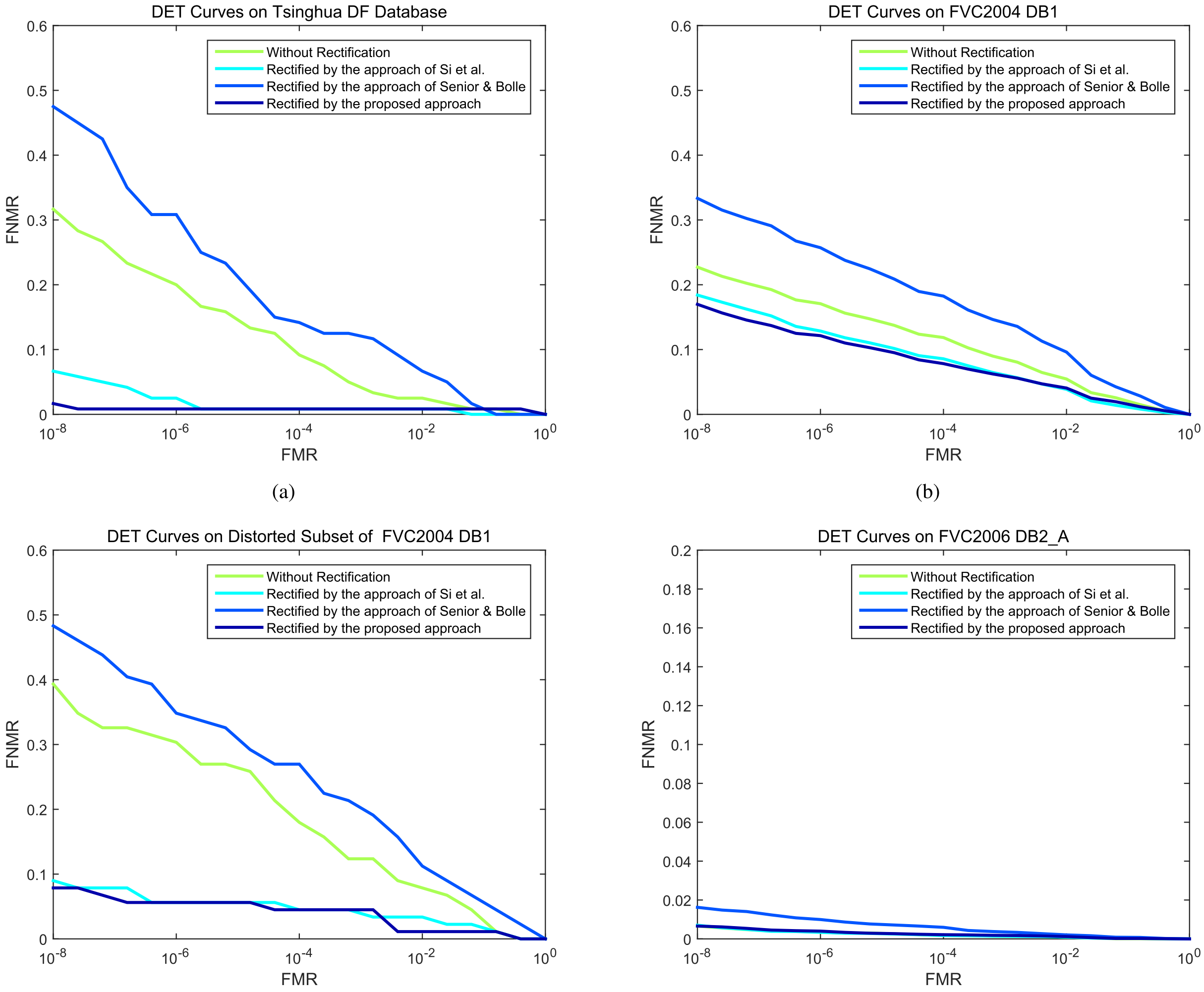


Fig. 10. The VeriFinger matching score distributions with/without rectification on two databases: (a) Tsinghua DF database and (b) distorted subset of FVC2004 DB1.

the dotted line), which suggests that most distorted fingerprints have been rectified successfully.

The distributions of matching scores with and without rectification on Tsinghua DF database and the distorted subset of FVC2004 DB1 are shown in Fig. 10 in order to show the effect of rectification on the matching scores of non-mated fingerprints. It is clear that the matching scores of non-mated fingerprints do not increase while the matching scores of mated fingerprints increase a lot after rectification. The distribution of matching scores of non-mated fingerprints after rectification is almost unchanged because it is dominated by very low matching scores.

Then the Detection Error Tradeoff (DET) curves using VeriFinger 6.2 SDK on four databases are plotted to compare the performance of the proposed method with that of previous method. Since the matching scores of VeriFinger are linked to the false match rate (FMR), this allows us to examine the false non-match rates (FNMR) at very low FMRs without performing impostor matches (very limited number of impostor matches can be done at our databases). As shown in Fig. 11, on all the three databases containing many distorted fingerprints, both the method of Si *et al.* and the proposed approach greatly increase the matching accuracy (the proposed method is slightly better) while the method of Senior and



(a) (b)

(c) (d)

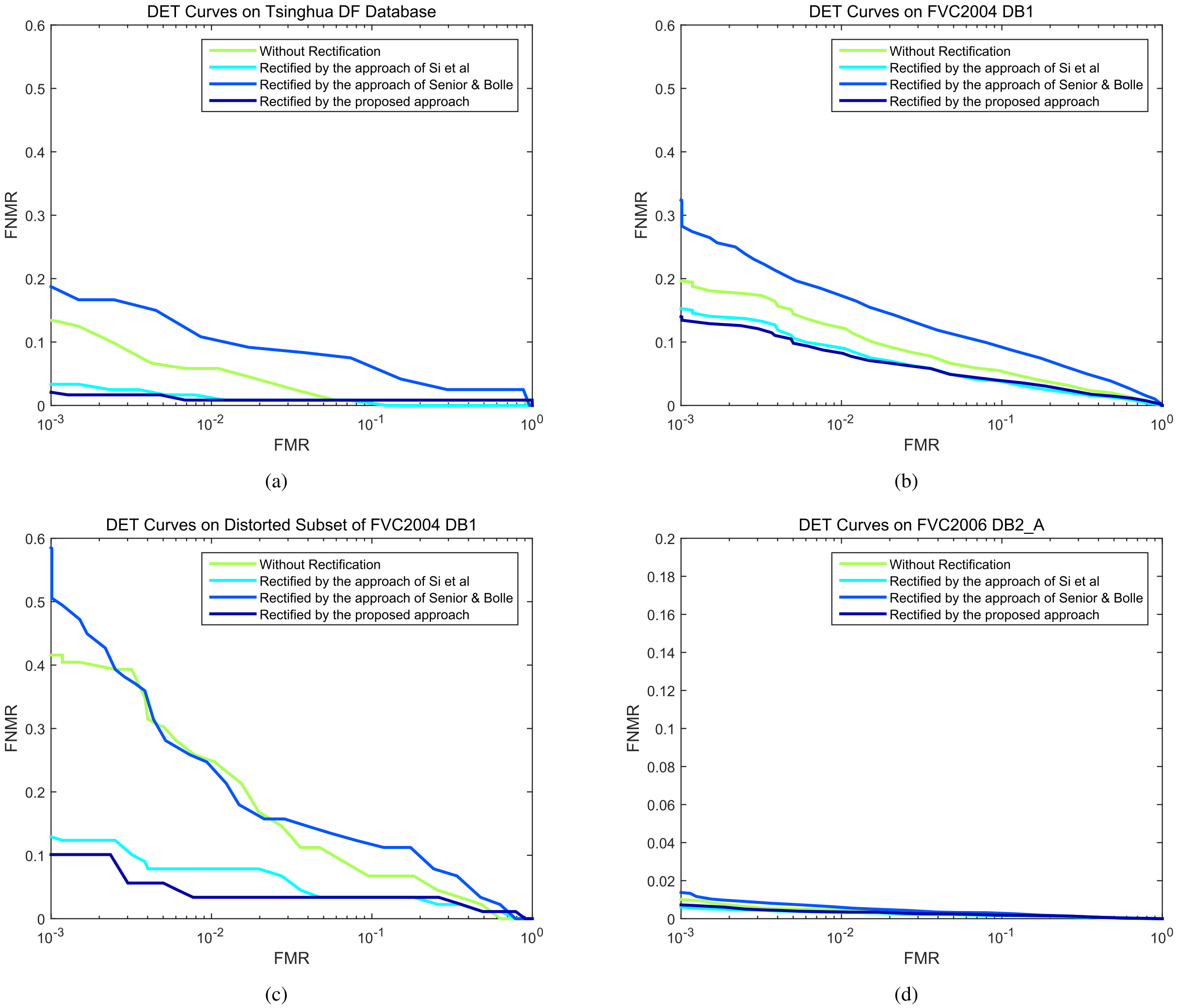
Fig. 11. The DET curves of four matching experiments using VeriFinger 6.2 SDK on four databases: (a) Tsinghua DF database, (b) FVC2004 DB1, (c) distorted subset of FVC2004 DB1, and (d) FVC2006 DB2_A.

Bolle reduces the matching accuracy. On FVC2006 DB2_A, which mainly contains normal fingerprints, both method of Si *et al.* and the proposed approach do not have obvious negative impact. Compared with the method of Si *et al.*, the performance of the proposed algorithm is only slightly better, because (1) on Tsinghua DF database, the improvement room is already very small, and (2) on FVC2004 DB1, most of the remaining low genuine matching scores are due to poor image quality, not distortion.

In order to make the conclusions more convincing, matching experiments are also conducted using the MCC SDK [25], which is a state of the art fingerprint matcher with published algorithm. Minutiae are still extracted using VeriFinger SDK. We used the LSA-R (the most accurate version of MCC) to compute the similarity between two fingerprints. The DET curves using MCC SDK on four databases are shown in Fig. 12, and the same conclusions can be drawn. The proposed approach achieves better performance on the databases which contain many distorted fingerprints and does not reduces the matching accuracy of normal fingerprints.

We also give several examples to show the performance of rectification, which are shown in Fig. 13. In these cases, the matching scores of the fingerprints rectified by the proposed approach are much higher than rectified by the nearest neighbor approach proposed by Si *et al.* [7].

Though the proposed rectification method has already well rectified most distorted fingerprints, the matching scores of several fingerprints drop after rectified, which are also shown in Fig. 9 (the points under the blue line), and the reasons for the failure are also shown. Reasons for the unsuccessful rectification include: the normal fingerprint is detected as distorted (false positive of detection) and rectified unnecessarily, and the distorted fingerprint is wrongly rectified. Distorted fingerprints which are detected as normal (false negative of detection) are not rectified, and thus the matching scores maintain unchanged. Note that false positive detection is not necessarily bad because even normal fingerprints may contain distortion of certain degrees, and thus rectification is beneficial for many normal fingerprints. The number of



(a)

(b)

(c)

(d)

Fig. 12. The DET curves of four matching experiments using MCC SDK on four databases: (a) Tsinghua DF database, (b) FVC2004 DB1, (c) distorted subset of FVC2004 DB1, and (d) FVC2006 DB2_A.

TABLE III
STATISTICS OF RECTIFICATION ERROR

Database	Tsinghua DF (120)		Distorted Subset of FVC2004 DB1 (89)	
	Normal fingerprints	Distorted fingerprints	Normal fingerprints	Distorted fingerprints
The method of Si *et al.*	10	9	0	12
The proposed method	1	2	0	6

the fingerprints rectified unsuccessfully on two databases, and the cause of error are analyzed in Table III. The result is also compared with that of the method of Si *et al.* On both databases, the proposed approach produces fewer errors.

The rectification error is mainly influenced by the poor quality of the input fingerprints. Based on our observation, the fingerprints with poor quality can be divided into two categories: fingerprints with bad pose, which have low centers or are non-frontal and fingerprints with low image quality. Fingerprints with low centers are usually locally distorted, which cannot be correctly rectified by global distortion correction. Fingerprints which are non-frontal or with low image quality have insufficient information while the regressor is trained using fingerprints of good quality. Two examples of these two cases are shown in Fig. 14. In addition, the error of center estimation and distortion detection may also lead to unsuccessful rectification.

C. Efficiency Analysis

We analyze the efficiencies of the proposed approach and the method of Si *et al.* on Tsinghua DF database and FVC2004 DB1. We calculate the average time of distortion detection and rectification on a PC with 3.30 GHz CPU. It should be noted that, rectification is not applied to

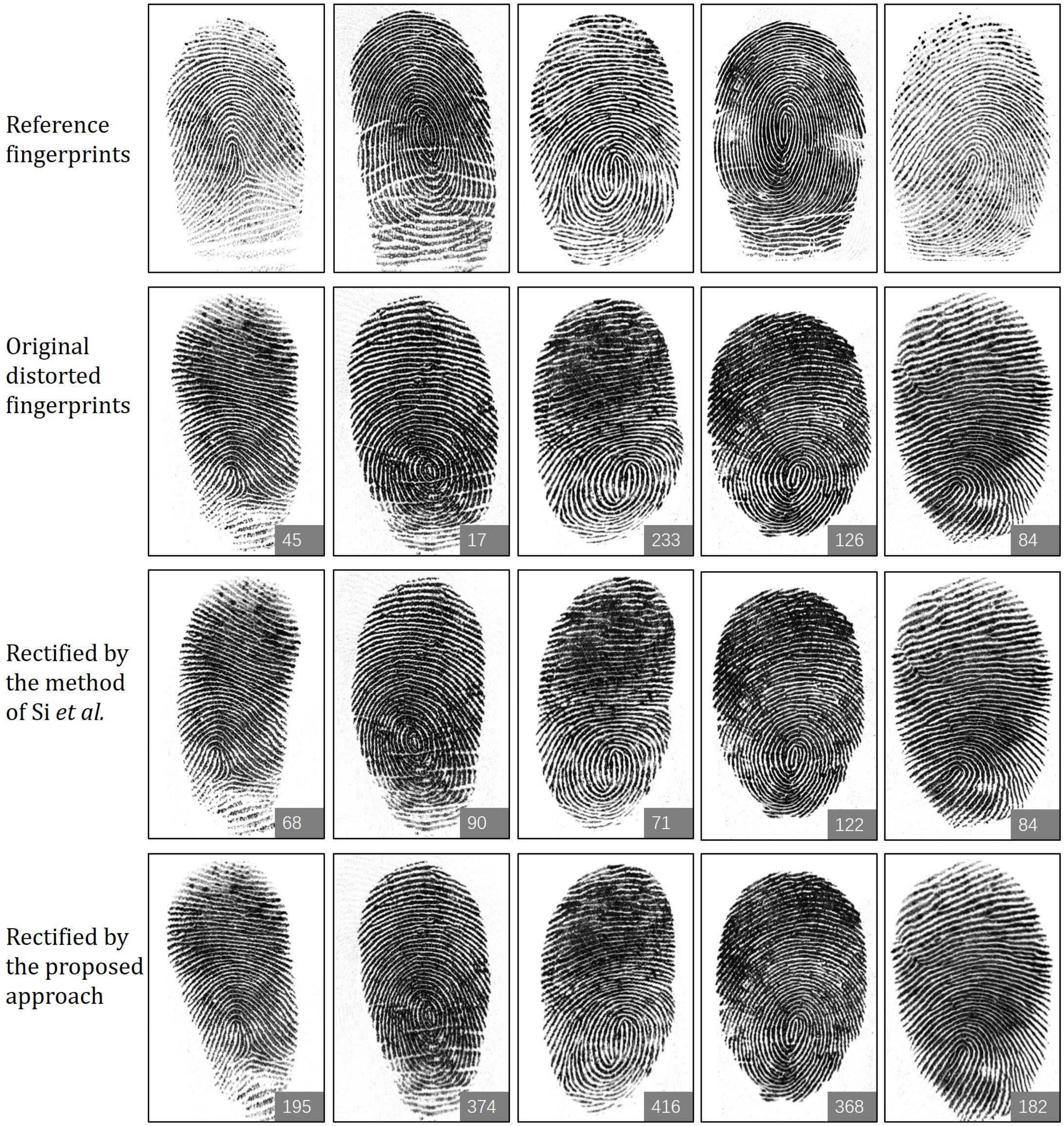
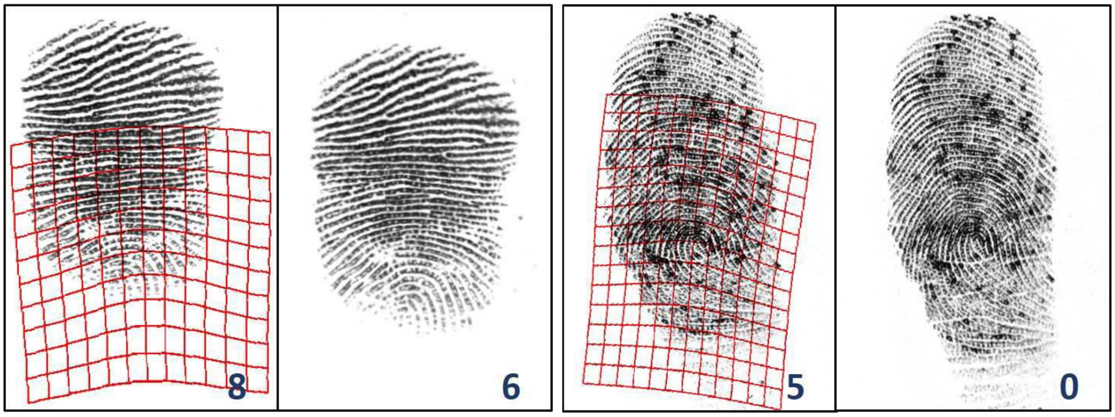

Reference fingerprints
Original distorted fingerprints
45
17
233
126
84
Rectified by the method of Si *et al.*
68
90
71
122
84
Rectified by the proposed approach
195
374
416
368
182

Fig. 13. Five examples of performance of rectification. The number under each fingerprint is the VeriFinger matching score between it and the mated reference fingerprint.

fingerprints which are detected as normal.Therefore, the average time of rectification only refers to distorted fingerprints.

As is shown in Table IV, the method of Si *et al.* has very different running times for fingerprints with and without core point as they use different methods for two different cases. For the distortion detection process, it takes about 15 times longer time for fingerprints without core point than for fingerprints with core point. For the distortion rectification process, it takes very long time for both kinds of fingerprints.

Our approach can apply to both cases, and the efficiency is significantly better. For the distortion detection process, only 0.4∼0.5 seconds are needed for both cases. Specifically, the efficiency increases by about 30 times for fingerprints without core point. For the distortion rectification process, the efficiency for both cases increases greatly by over 150 times. Due to the efficient distortion detection and rectification processes, the total running time for a distorted fingerprint is less than one second and is over 30 times faster than the method of Si *et al.*



(a) (b)

Fig. 14. Two examples of unsuccessful rectification: (a) fingerprint with bad pose (only the fingertip is captured), (b) fingerprint with low quality. The number under each fingerprint is the VeriFinger matching score between it and the mated fingerprint.

TABLE IV

COMPUTATIONAL COSTS (SECOND) OF MAJOR STEPS OF THE METHOD OF SI *et al.* AND THE PROPOSED METHOD

	FVC2004 DB1		**Tsinghua DF**	
	Si *et al.*	**Proposed**	**Si *et al.***	**Proposed**
Distortion Detection	0.87/14.93	0.41/0.41	1.08/15.72	0.50/0.53
Distortion Rectification	17.40/20.41	0.12/0.11	17.32/25.12	0.10/0.11
Whole system	18.27/35.34	0.53/0.52	18.40/40.84	0.60/0.64

[1] Speed is reported respectively for fingerprints with/without core point.

[2] The computation time of pose estimation is included in that of distortion detection and the computation time of distortion field estimation is included in that of rectification.

VI. CONCLUSION

In a fingerprint recognition system, distortion greatly influences the matching performance of fingerprints. A recent study shows that distortion rectification can effectively improve the recognition rate of distorted fingerprints. However, this method is too slow to be used in practical applications.

In this paper, we present an efficient distortion rectification method, whose speed is over 30 times faster than the existing method. The proposed method comprises two main parts: a two-step pose estimation method including Hough forest based center estimation and regression based finger direction estimation, and a support vector regression based distortion field estimation method.

We do several experiments to evaluate the performance of the proposed rectification method on three databases: FVC2004 DB1, FVC2006 DB2_A and Tsinghua DF database. The results show that the rectification performance of the proposed method is comparable with that of the existing method, but the speed improves significantly. The total detection time and rectification time for a distorted fingerprint is less than one second.

The limitation of the proposed pose estimation method is that the error of finger center estimation greatly influences the accuracy of finger direction estimation. Besides, fingerprints which are obtained by smaller sensors (such as fingerprint sensors in smart phones) usually have very small effective areas. The proposed pose estimation method may not be applicable for these fingerprints.

The limitation of the proposed rectification method is that the types of distortion in the training set of Tsinghua DF are limited and cannot cover all possible distortion fields in practice, especially distortion fields of some latent fingerprints from crime scenes. Another limitation is that the current algorithm is trained and tested on plain fingerprints whose image quality is not very low. It is not very effective for distorted latent fingerprints.

Future work can focus on: (1) more accurate pose estimation for low quality fingerprints; (2) the rectification for non-frontal fingerprints and latent fingerprints, which have very limited information and different distortion types; (3) the rectification for fingerprints directly obtained by cameras of smart phones, whose distortion is caused by rotated fingers or changing viewpoints.

REFERENCES

[1] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, 2nd ed. New York, NY, USA: Springer-Verlag, 2009.

[2] *FVC-onGoing: On-Line Evaluation of Fingerprint Recognition Algorithms*. [Online]. Available: https://biolab.csr.unibo.it/FVCOnGoing/

[3] V. N. Dvornychenko and M. D. Garris, "Summary of NIST latent fingerprint testing workshop," NISTIR, Gaithersburg, MD, USA, Tech. Rep. 7377, Nov. 2006.

[4] R. Cappelli, D. Maio, D. Maltoni, J. L. Wayman, and A. K. Jain, "Performance evaluation of fingerprint verification systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 3–18, Jan. 2006.

[5] L. M. Wein and M. Baveja, "Using fingerprint image quality to improve the identification performance of the U.S. Visitor and Immigrant Status Indicator Technology Program," *Proc. Nat. Acad. Sci. USA*, vol. 102, no. 21, pp. 7772–7775, 2005.

[6] A. W. Senior and R. M. Bolle, "Improved fingerprint matching by distortion removal," *IEICE Trans. Inf. Syst.*, vol. E84-D, no. 7, pp. 825–832, Jul. 2001.

[7] X. Si, J. Feng, J. Zhou, and Y. Luo, "Detection and rectification of distorted fingerprints," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 555–568, Mar. 2015.

[8] J. Gall and V. Lempitsky, "Class-specific Hough forests for object detection," in *Decision Forests for Computer Vision and Medical Image Analysis*. London, U.K.: Springer-Verlag, 2013, pp. 143–157.

[9] N. K. Ratha, K. Karu, S. Chen, and A. K. Jain, “A real-time matching system for large fingerprint databases,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 799–813, Aug. 1996.

[10] X. Chen, J. Tian, and X. Yang, “A new algorithm for distorted fingerprints matching based on normalized fuzzy similarity measure,” *IEEE Trans. Image Process.*, vol. 15, no. 3, pp. 767–776, Mar. 2006.

[11] A. Almansa and L. Cohen, “Fingerprint image matching by minimization of a thin-plate energy using a two-step algorithm with auxiliary variables,” in *Proc. WACV*, Dec. 2000, pp. 35–40.

[12] Z. M. Kovacs-Vajna, “A fingerprint verification system based on triangular matching and dynamic time warping,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1266–1276, Nov. 2000.

[13] L. R. Thebaud, “Systems and methods with identity verification by comparison and interpretation of skin patterns such as fingerprints,” U.S. Patent 5 909 501 A, Jun. 1, 1999.

[14] J. Feng, Z. Ouyang, and A. Cai, “Fingerprint matching using ridges,” *Pattern Recognit.*, vol. 39, no. 11, pp. 2131–2140, 2006.

[15] R. M. Bolle *et al.*, “System and method for distortion control in live-scan inkless fingerprint images,” U.S. Patent 6 064 753 A, May 16, 2000.

[16] N. K. Ratha and R. M. Bolle, “Effect of controlled image acquisition on fingerprint matching,” in *Proc. 14th Int. Conf. Pattern Recognit.*, vol. 2. Aug. 1998, pp. 1659–1661.

[17] Y. Fujii, “Detection of fingerprint distortion by deformation of elastic film or displacement of transparent board,” U.S. Patent 7 660 447 B2, Feb. 9, 2010.

[18] C. Dorai, N. K. Ratha, and R. M. Bolle, “Dynamic behavior analysis in compressed fingerprint videos,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 58–73, Jan. 2004.

[19] L. Hong, Y. Wan, and A. Jain, “Fingerprint image enhancement: Algorithm and performance evaluation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 777–789, Aug. 1998.

[20] K. De Brabanter *et al.*, “LS-SVMlab toolbox user’s guide version 1.8,” ESAT-SISTA, KU Leuven, Leuven, Belgium, Internal Rep. 10-146, 2010.

[21] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, 2011. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[22] Neurotechnology Inc. *VeriFinger*. Accessed: Sep. 7, 2017. [Online]. Available: http://www.neurotechnology.com

[23] X. Yang, J. Feng, and J. Zhou, “Localized dictionaries based orientation field estimation for latent fingerprints,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 955–969, May 2014.

[24] A. M. Bazen and S. H. Gerez, “Systematic methods for the computation of the directional fields and singular points of fingerprints,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 905–919, Jul. 2002.

[25] R. Cappelli, M. Ferrara, and D. Maltoni, “Minutia cylinder-code: A new representation and matching technique for fingerprint recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2128–2141, Dec. 2010.

Shan Gu received the B.E. degree from the Department of Automation, Tsinghua University, Beijing, China, in 2012, where she is currently pursuing the Ph.D. degree with the Department of Automation. Her research interests include fingerprint recognition, computer vision, and pattern recognition.

Jianjiang Feng received the B.S. and Ph.D. degrees from the School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, China, in 2000 and 2007, respectively. From 2008 to 2009, he was a Post-Doctoral Researcher with the PRIP Laboratory, Michigan State University. He is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing. His research interests include fingerprint recognition and computer vision. He is an Associate Editor of the *Image and Vision Computing*.

Jiwen Lu received the B.Eng. degree in mechanical engineering and the M.Eng. degree in electrical engineering from the Xian University of Technology, Xian, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2012. From 2011 to 2015, he was a Research Scientist with the Advanced Digital Sciences Center, Singapore. He is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing, China. His current research interests include computer vision, pattern recognition, and machine learning. He has authored or co-authored over 150 scientific papers in these areas, where 42 were IEEE TRANSACTIONS papers. He was a recipient of the National 1000 Young Talents Plan Program in 2015. He is an Elected Member of the Information Forensics and Security Technical Committee of the IEEE Signal Processing Society. He is the workshop chair/special session chair/area chair for over ten international conferences. He serves as an Associate Editor of *Pattern Recognition Letters*, *Neurocomputing*, and IEEE ACCESS, a Managing Guest Editor of *Pattern Recognition and Image and Vision Computing*, and a Guest Editor of *Computer Vision and Image Understanding*.

Jie Zhou received the B.S. and M.S. degrees from the Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the Ph.D. degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, China, in 1995. From 1995 to 1997, he served as a Post-Doctoral Fellow with the Department of Automation, Tsinghua University, Beijing, China. Since 2003, he has been a Full Professor with the Department of Automation, Tsinghua University. In recent years, he has authored over 100 papers in peer-reviewed journals and conferences. Among them, over 40 papers have been published in top journals and conferences, such as IEEE PAMI, TIP, and CVPR. His current research interests include computer vision, pattern recognition, and image processing. He received the National Outstanding Youth Foundation of China Award. He is an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the *International Journal of Robotics and Automation*, and two other journals.