A Multiple Feature Fusion System for Fingerprint Recognition

Franglin Chen* and Ming Li

College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha, Hunan, P.R. China, 410073.

Abstract: A number of different features, besides minutiae, have been used for fingerprint matching. Previous studies have shown that the performance of a fingerprint recognition system can be improved by combining these features with minutiae through a fusion strategy. However, most of these studies extract only a single type of feature for fingerprint recognition, and then fuse it with conventional minutiae-based method to improve performance. In this paper, in order to select the best fusion of fingerprint features, a comparative study of combining multiple features is firstly carried out on several fingerprint databases. All the comparing results show that, generally the more features being used, the better the performance is. However, beyond four features, the performance improvement is negligible. Based on these observations, a multi-feature based fingerprint recognition system using the best combination of the four features is proposed. The experimental results comparing to the-state-of-art algorithm show the effectiveness of the proposed system.

Keywords: Fingerprint recognition, Multiple features, Fusion, Minutiae, Equation error rate.

1. INTRODUCTION

Biometric technologies are playing more and more important role in various security applications [1]. Among these technologies, fingerprint recognition is considered to be one of the most reliable. As a result, it has been extensively used in personal identification [2, 3].

There are many different features which can be used to represent fingerprints. The most common representation is based on minutiae which are ridge endings or bifurcations on the fingerprints. Almost all the fingerprint recognition systems store the minutiae template (sometimes with singular point together) combining of minutiae position and orientation in the database [2]. Minutiae-based fingerprint recognition systems consist of two steps, *i.e.*, minutiae extraction and minutiae matching. In the minutiae matching process, the minutiae feature of a given fingerprint is compared with the minutiae template, and the number of matched minutiae are determined. If the matching score exceeds a predefined threshold, the two fingerprints can be regarded as belonging to the same finger. However, minutiae-based algorithms do not utilize all the information present in a fingerprint. Pankanti *et al*. [4] showed that minutiae based representation cannot provide desired distinguishing ability for large-scale fingerprint identification tasks.

There have been a number of studies on extracting new features from a fingerprint to improve the matching performance. These fingerprint features can be generally classified into two categories: local features and global features. Minutiae [2] and minutia descriptor [5] are widely used local features, while FingerCode [6], ridge feature map [7], orientation map [8], and density map [9] are global ones. Feng [10] proposed an algorithm to combine minutiae-based descriptor and texture-based descriptor. Choi *et al*. [11] combined ridge feature with minutiae feature for fingerprint matching. Jain and Feng [12] proposed a fusion scheme for latent fingerprint matching, which can improve the accuracy. These studies showed the superiority of the combined two features to single feature. However, most of these studies extract only a single type of feature for fingerprint recognition, and then combine it with the conventional minutiae-based method to improve the performance. The questions that we are interested in are: (1) Can we further improve the recognition performance by combining a large number of features? (2) Among the various combinations of features, which one is the best choice? (3) Is the performance of feature fusion seriously depending on the fusion method? These questions are important issues for the design of practical fingerprint recognition systems.

This paper undertakes a study on feature combination for fingerprint recognition and proposed a multi-feature fusion system for fingerprint recognition. To select the best combination of features, a comparative study is conducted to analyze the performance of combining more than two features for fingerprint recognition. Different combinations of features are analyzed and compared by using different fusion schemes and fingerprint databases. Based on

*Address correspondence to this author at the College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha, Hunan, P.R. China, 410073; Tel: (+86)731-84574977; Fax: (+86)731-84574977; E-mail: chenfanglin83@163.com

E-ISSN: 2410-2938/15

the experimental results and analysis, a multi-feature fusion based method for fingerprint recognition is proposed. The experimental results show that a better performance can be obtained in the proposed system.

Preliminary version of this paper appeared in [13]. The present paper contains more experiments conducted on FVC02 DB1 and DB2 as well as a refined discussion. In Section 3, we also proposed a feature-fused fingerprint recognition system which can largely improve the system's performance.

The rest of the paper is organized as follows: In Section 2, the best feature combination selection is introduced. The proposed multiple feature fusion system is presented in Section 3. Section 4 reports the performance of the proposed system. We provide conclusion and discussion in section 5.

2. BEST FINGERPRINT FEATURE COMBINATION SELECTION

2.1. Feature Candidates

There are many features to represent fingerprints. The most widely used features are minutiae and features based on texture and ridges. The other features, such as level 3 features [2] (*i.e.*, pores, incipient ridges, creases) are very difficult to detect in medium-quality or low-quality fingerprints. Several features that are variations of minutiae, texture and ridges, have also been derived. Considering that features selected for the fusion scheme should be diverse, and mutually complementary, we have selected the following features as candidates: minutiae, orientation-based minutia descriptor, FingerCode, ridge feature map, orientation map, density map [13].

2.2. Comparing of Different Combinations

In this sub section, experiments that compare different combinations of fingerprint features are conducted for best fingerprint feature combination selection.

A variety of fusion rules have been proposed [14, 15], such as product rule, sum rule, max rule, min rule, median rule and majority voting rule, which can be used to combine the scores from the matchers based on different features. These heuristic fusion strategies do not perform well in situations where there are more than two classifiers [16]. Many studies have shown that by using Neyman-Pearson rule [17] or Support Vector Machine (SVM) [18], a robust fusion can be obtained. So in our study, we will use these two fusion schemes.

To avoid incidentally good result on just one dataset, comparing experiments are conducted on three databases, FVC02 DB1 and DB2 [19], and the THU database [20].

A subset of 3200 (400×8) fingerprints are randomly chosen from THU database to form the training set for the fusion schemes. Our experiments and former studies [8] have shown that this number of fingerprints for training is sufficient, since the performance improves very little by increasing the training set. The fingerprints in DB1, DB2 and the rest of the fingerprints in THU database (including $427\times8=3416$ fingerprints) are used for the testing.

The receiver operating curves (ROC) that plot FAR versus FRR is useful for comparing fusion verification schemes. The ROC curves of the best combinations using different number of features are plotted in Figure **1**. The labels of the curves in short form are described in Table **1**. For example, *MOR* means fingerprint verification using three features: minutiae, orientation and ridge density map.

Table 1: The Abbreviation for Different Features [13]

Abbreviation	Feature
M	Minutiae
D	Minutia Descriptor
C	FingerCode
F	Ridge Feature Map
O	Orientation
R	Ridge Density Map

The results illustrate that combining multiple features for fingerprint recognition can significantly improve the performance of the system. In all the combinations, minutiae are included. This is because minutiae are the most important representation of fingerprints and has the best discriminability for fingerprint recognition.

When combining two features, *MR* has the best performance among all the combinations. Ridge density map is a set of ridge distances in the fingerprint, which describes the ridge denseness or sparseness, so it is more complementary to minutiae features (containing position and direction information) than the others. Thus the combination *MR* has the best performance in general.

The performance of the two fusion strategies is roughly equivalent. The advantage of SVM is that it has

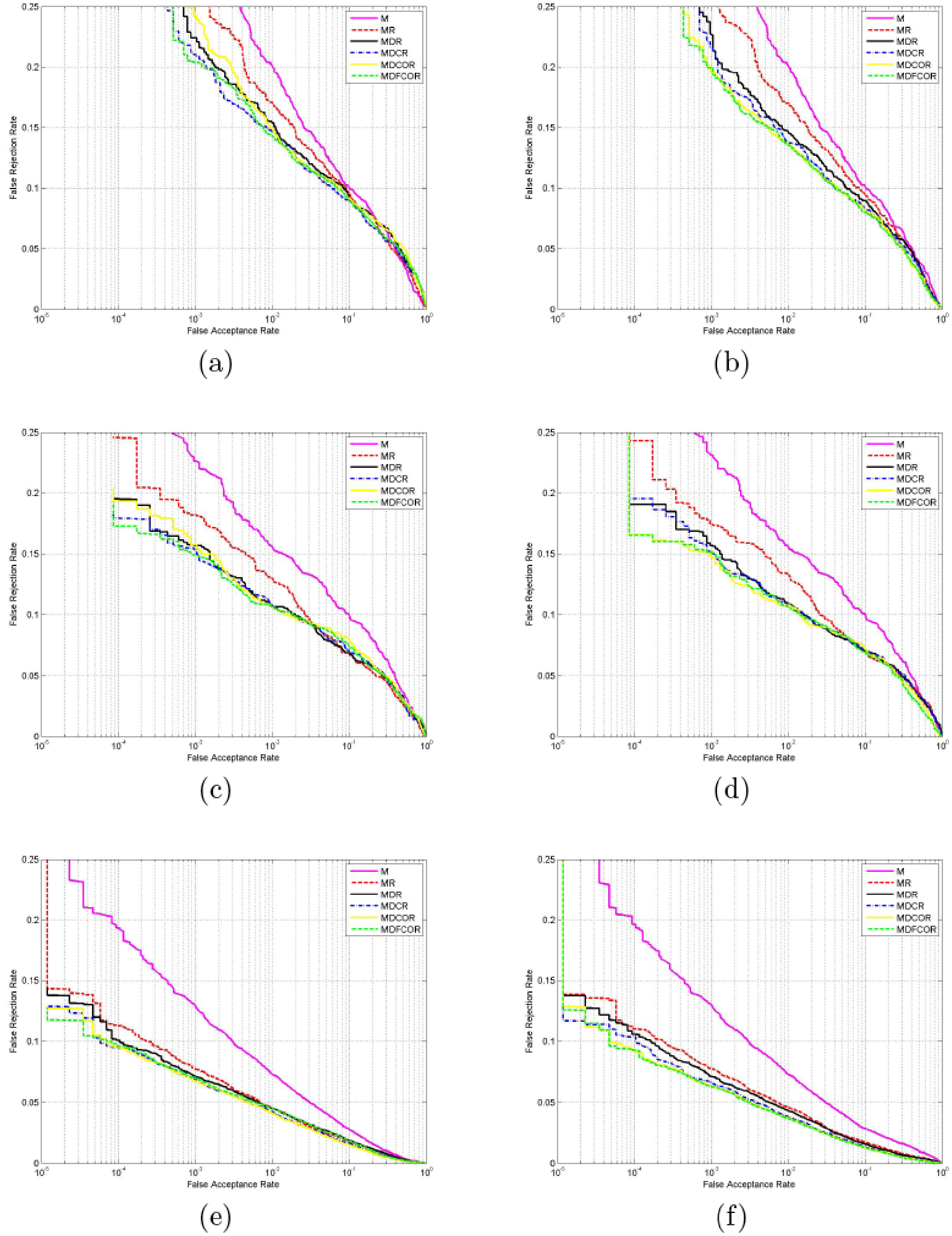

Figure1: ROCs of the best combinations using different number of features: (**a**), (**c**), (**e**) for Neyman-Pearson rule, while (**b**), (**d**), (**f**) for SVM algorithm, on FVC02 DB1 (first row), DB2 (second row), and THU testing database (last row), respectively.

the strong ability to learn from small-numbered samples [21].

The best feature combinations are listed in Table **2**. Their computational cost and storage cost are also given. In general, the experimental results suggest that utilizing more feature information can result in a better performance. However, it is not encouraged to use too many features, since more features mean additional computational cost and storage cost. Based on Table **3**, we suggest that four features be used in the combination scheme, since the computational cost and especially the storage cost are increased greatly when more than four features are fused.

Table 2: The Best Combination Schemes using Different Number of Features

Number of Features	Best Combinations	Computational Cost (ms)	Storage Cost (KB)
1	M	417.33	0.25
2	MR	608.41	0.33
3	MDR	716.30	0.64
4	MDCR	1261.75	0.80
5	MDFCR	1571.03	2.56
6	MDFCOR	1699.68	2.64

3. THE PROPOSED MULTIPLE FEATURE FUSION BASED FINGERPRINT RECOGNITION SYSTEM

Based on the above comparative studies and observations, we proposed a multiple feature fusion based method for fingerprint recognition. We also compared the proposed algorithm with a well-known commercial method Neurotechnology Verifinger 6.2 SDK [22].

Since having more features means additional computational cost and storage cost, our fusion system uses four features in the combination scheme. Based on the comparing experimental results, the selected combination is *MDCR*. Compared with Neyman-Pearson rule, SVM does not need to estimate the probability density functions. Thus, SVM is chosen as the fusion strategy based on its time efficiency [13].

Let $s = (s_1, s_2, \cdots, s_n)$ denote the scores from n classifiers based on different features. Let ω_G denote the genuine class, while ω_I denote the imposter class. Suppose that the original data space of SVM is *L*, and the feature space is *H* (here we use *L* as a hint for "low dimensional", and *H* for "high dimensional"). Let Φ be the transforming function between the two spaces

$$\Phi : \mathrm{L} \mapsto \mathrm{H}. \tag{1}$$

Let N be the number of training samples. Denoting the set of training data as $\{s_i, y_i\}, i = 1,2,\cdots,N, s_i \in \mathrm{L},\ and\ y_i \in \{-1,+1\}$ (denotes the imposter and genuine class, respectively). The SVM calculates the sign of $f(s)$ as the decision result, where $f(s)$ is calculated as

$$f(s) = \sum_{i=1}^{N} \alpha_i y_i K(s_i, s) + b. \tag{2}$$

The nature of the decision surface is mainly defined by the kernel function $K(s_i, s)$, which should satisfy Mercer's conditions. The commonly used kernels include polynomial kernels $K(s_i, s) = (s_i^t s + 1)^d$, where d is a positive integer to define the degree of a polynomial decision surface, and Gaussian kernels $K(s_i, s) = e^{-g\|s_i - s\|^2}$. The kernel function $K(s_i, s)$ can be easily computed by an inner product of the non-linear transform function [23].

The classical SVM is a technique for binary classification in the field of pattern recognition, since the fusion strategy relies on computing the sign of $f(s)$ in Eqn. (2). In this study, a modification in order to obtain a fusion score, but not a binary classifier

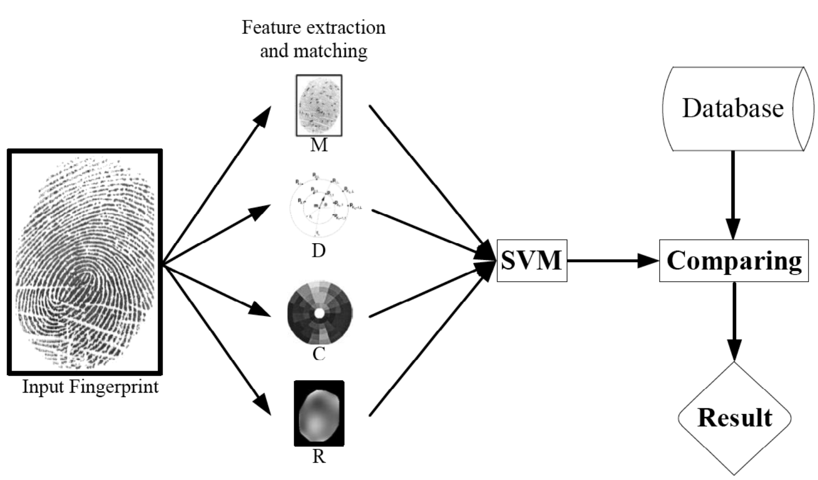

Figure 2: The flowchart of the proposed multiple feature fusion system.

decision is proposed. The fusion score s_f is normalized from $f(s)$ as

$$s_f(s) = \frac{\tanh(f(s))+1}{2}. \tag{3}$$

Based on this modification, for a given score s_0, the classification rule is as

$$s_0 \in \begin{cases} \omega_G, & if \quad s_f(s_0) > \lambda, \\ \omega_I, & otherwise. \end{cases} \tag{4}$$

where λ is the threshold to minimize FRR under a given FAR.

The flowchart of the proposed system is shown in Figure **2**.

4. THE PERFORMANCE OF THE PROPOSED SYSTEM

The average performance on three testing databases of the proposed feature fusion system is shown in Table **3**. The average computational cost, storage cost, and EER for these methods have been computed. The algorithms are implemented in C on an AMD 2.0 GHz, 2.0 GB PC. The results show that, requiring a little more computational and storage cost, the proposed algorithm can greatly improve the matching performance of fingerprint recognition.

Table 3: The Performance Comparison of the Proposed Fusion Scheme with the State-of-Art

The Comparing Algorithms	EER (DB1) (%)	EER (DB2) (%)	EER (THU) (%)
Verifinger 6.2 [22]	0.71	0.69	0.64
the proposed	0.48	0.45	0.37

Table 4: The Computation and Storage Cost Comparison of the Proposed Fusion Scheme with the State-of-Art

The Comparing Algorithms	Computational Cost (ms)	Storage Cost (KB)
Verifinger 6.2 [22]	417.33	0.25
the proposed	1261.75	0.80

5. CONCLUSION

In this paper, the performance of combining multiple features for fingerprint recognition is comparatively studied. Two widely used fusion schemes (Neyman-Pearson rule and SVM) are implemented for the feature fusion independently. Irrespectively whether Neyman-Pearson rule or SVM is used, experimental results show that a significant improvement can be obtained by combining features. All the results show that, generally the more features being used, the better the performance is. However, beyond four features, the performance improvement is negligible. Thus, we suggest to limit the number of feature types to four due to performance and computing effectiveness. Based on these comparative studies, we proposed a multiple feature fusion fingerprint recognition system, which can improve the recognition performance significantly.

ACKNOWLEDGEMENT

This work was supported by the Natural Science Foundation of China under Grant 61203263.

APPENDIX

The features used in this study are described as follows:

1. Minutiae

Minutiae are local ridge characteristics that occur at either a ridge bifurcation or a ridge ending. In our study, we use the algorithm proposed by Jain *et al*. [2] to extract minutiae, which has been shown to be an effective and widely used algorithm.

Most minutiae-based approaches count the number of matching minutiae pairs and normalize it with the number of minutiae in each fingerprint to get the matching score [8]. Two minutiae are regarded as matched when: 1) the differences of their coordinates are less than Δ_x and Δ_y for the x-axis and y-axis, respectively, and 2) the angular difference between their directions does not exceed Δ_θ. The parameters Δ_x, Δ_y and Δ_θ are empirically determined. Let the numbers of the minutiae located in the intersection of the template fingerprint (F_t) and query fingerprint (F_q) be N_t and N_q, respectively. The number of matched minutiae pairs is denoted as N. The matching score is usually computed as

$$s_M = \frac{N^2}{N_t N_q}. \tag{.1}$$

This score lies in the interval $[0,1]$, and the value represents the degree of agreement between the two minutiae sets.

2. Minutia Descriptor

Tico and Kuosmannen [24] built an orientation-based minutia descriptor for each minutiae, which consists of the original minutiae point and a set of local orientation values uniformly sampled around this point. The sampling points assigned to each minutia is organized in a circular pattern around the minutia, $m=[x,y]^T$. Denoting $\theta_{k,l}$ as the orientation estimated at $p_{k,l}$, the minutia descriptor is defined as follows:

$$f = \left\{ \{\lambda(\theta_{k,l},\theta)\}_{k=1}^{K_l} \right\}_{l=1}^{L}, \tag{.2}$$

where $\lambda(\alpha,\beta)$ denotes the difference between the angles α and β.

When comparing two fingerprints, they are first aligned. A similarity function between minutiae is derived to identify the corresponding features and estimate the similarity between the two fingerprint impressions. Let a and b denote the labels associated with two minutiae whose descriptors (Eqn. (.2)) are, $f(a)=\{\alpha_{k,l}\}$ and $f(b)=\{\beta_{k,l}\}$, respectively. The similarity function between the two minutiae is defined as

$$S(a,b)=\frac{1}{K}\sum_{l=1}^{L}\sum_{k=1}^{K_l}s(x_{k,l}), \tag{.3}$$

where $K=\sum_{l=1}^{L}K_l$, $x_{k,l}=(2/\pi)\lambda(\alpha_{k,l},\beta_{k,l})$, and $s(x)$ is define as [24]

$$s(x)=\exp(-16x). \tag{.4}$$

If we denote a_i and b_j as the matched minutiae pairs, N_t and N_q as the number of minutiae from two fingerprints, respectively, and C is the set of matched pairs, the matching score can be calculated as

$$s_D=\frac{1}{N_tN_q}\left(\sum_{(i,j)\in C}S(a_i,b_j)\right)^2. \tag{.5}$$

3. FingerCode

FingerCode was proposed by Jain *et al*. [6]. They first detect a reference point in the fingerprint (generally, the core) and extract the region of interest (divided in sectors) around it, then filtered the image with a bank of Gabor filters with different orientations. The filtered results are discretely coded as a feature vector (FingerCode) by computing the average absolute deviation from the mean of gray values in individual sectors.

Fingerprint matching is based on finding the Euclidean distance between the corresponding FingerCodes. Since the features are not rotationally invariant, the database stores ten templates for each fingerprint. These ten templates correspond to various rotations of the fingerprint image. The matching distance d_C is set as the minimum of the ten scores, *i.e*., matching of the input FingerCode with each of the ten templates, which corresponds to the best alignment of the two fingerprints being matched. The final matching score is obtained by normalizing the distance score:

$$s_C=\exp(-d_C). \tag{.6}$$

4. Ridge Feature Map

To avoid the sensitivity of the reference point detection in FingerCode, Ross *et al*. [7] proposed to use the entire filtered images in computing the feature, called “ridge feature map”. They first filter the fingerprint image with eight Gabor filters each with different direction. Then the image is tessellated into square cells, and features from each of the cells correspond to the variance of each cell. An eight-dimensional feature map is obtained corresponding to the eight filtered images.

In the absence of a reference point, the transformation parameters (translation and rotation) are estimated by minutiae matching. Then the parameters are used to rotate and translate the feature of the query image. Given two ridge feature maps of the template and the query fingerprint images, the matching distance d_F is computed as the sum of the Euclidean distances of the eight-dimensional feature vectors. A high distance score indicates a poor match. The final distance score is normalized as

$$s_F=\exp(-d_F). \tag{.7}$$

5. Orientation Map

Another important type of global features in fingerprints is the orientation map (also called orientation field) [25], which can directly describe the global structure of the fingerprint ridge pattern. It is defined as a matrix whose elements are the ridge direction at the corresponding pixel (block) in the original image. The direction is defined in $[0,\pi]$. In this study, we use a so-called model-based method for the computation of orientation field, which was proposed in [20]. When the coarse field is computed by using the gradient-based algorithm, a further advantage can be gained by using the model for a weighted approximation. Due to the global approximation, the model-based orientation field estimation algorithm has a robust performance.

In the matching step, the correlation between two aligned orientation fields, O_t (template) and O_q (query), is computed as below. Let Ω_O denote the intersection of the two effective regions of fingerprints after alignment (usually based on minutiae matching), and N_O is the total area of Ω_O. The matching score between the two orientation fields is defined as

$$s'_O = \frac{1}{N_O} \sum_{(i,j)\in\Omega_O} \delta(i,j). \tag{.8}$$

In Eqn. (.8), $\delta(i,j)$ is the difference between the orientation values at the point (i,j) in fingerprint images F_t and F_q, which is formulated as follows:

$$\delta(i,j) = \begin{cases} \delta_0(i,j), & \text{if } \delta_0(i,j) \leqslant \frac{\pi}{2}, \\ \pi - \delta_0(i,j), & \text{otherwise,} \end{cases} \tag{.9}$$

and $\delta_0(i,j)$ is defined as:

$$\delta_0(i,j) = | O_t(i,j) - O_q(i,j) |, \tag{.10}$$

where $O_t(i,j)$ and $O_q(i,j)$ are the directions associated with pixels $F_t(i,j)$ and $F_q(i,j)$. Usually, s'_O is normalized by

$$s_O = \frac{\pi - 2s'_O}{\pi}. \tag{.11}$$

A large value of s_O indicates that there is a high degree of correspondence between two aligned orientation fields.

6. Density Map

Density map is a set of ridge distances in the fingerprint, where the ridge distance is usually defined as the length of the segment connecting the centers of two adjacent and parallel ridges along the line perpendicular to the ridges. Density map describes the denseness or sparseness of ridges in a fingerprint. Wan *et al*. [9] proposed a polynomial model to approximate the density map and then utilize it into the matching stage. Given two aligned ridge density maps, R_t (template) and R_q (query), let Ω_R denote the intersection of the two effective regions after alignment. Let N_R be the total area of Ω_R. The dissimilarity score between the two density maps is computed as

$$s'_R = \frac{1}{N_R} \sum_{(i,j)\in\Omega_R} | R_t(i,j) - R_q(i,j) |, \tag{.12}$$

where $R_t(i,j)$ and $R_q(i,j)$ are the ridge density at $F_t(i,j)$ and $F_q(i,j)$. The matching score is normalized as

$$s_R = \exp(-s'_R). \tag{.13}$$

In this study, ridge feature map, orientation map, and density map use minutiae alignment to fulfill the registration, while orientation-based minutia descriptor and FingerCode do not need minutiae alignment.

REFERENCES

[1] Hu R, Damper R. Optimal weighting of bimodal biometric information with specific application to audio-visual person identification, Information Fusion 2009; 10(2): 172 - 182. URL http://www.sciencedirect.com/science/article/pii/S1566253508000420 http://dx.doi.org/10.1016/j.inffus.2008.08.003

[2] Maltoni D, Maio D, Jain A, Prabhakar S. Handbook of Fingerprint Recognition (Second Edition), Springer, London, 2009. http://dx.doi.org/10.1007/978-1-84882-254-2

[3] Chin Y, Ong T, Teoh A, Goh K. Integrated biometrics template protection technique based on fingerprint and palmprint feature-level fusion, Information Fusion 18 (0) (2014) 161 - 174. URL http://www.sciencedirect.com/science/article/pii/S1566253513000870 http://dx.doi.org/10.1016/j.inffus.2013.09.001

[4] Pankanti S, Prabhakar S, Jain AK. On the individuality of fingerprints, IEEE Transactions on Pattern Analysis and Machine Intelligence 2002; 24(8): 1010-1025. http://dx.doi.org/10.1109/TPAMI.2002.1023799

[5] Cappelli R, Ferrara M, Maltoni D. Fingerprint indexing based on minutia cylinder-code, IEEE Transactions on Pattern Analysis and Machine Intelligence 2011; 33(5): 1051-1057. http://dx.doi.org/10.1109/TPAMI.2010.228

[6] De Zhi VT, Suandi SA. Fingercode for identity verification using fingerprint and smart card, in: Control Conference (ASCC), 2015 10th Asian, IEEE, 2015, pp. 1-6. http://dx.doi.org/10.1109/ascc.2015.7244877

[7] Ross A, Jain AK, Reisman J. A hybrid fingerprint matcher, Pattern Recognition 2003; 36(7): 1661-1673. http://dx.doi.org/10.1016/S0031-3203(02)00349-7

[8] Gu J, Zhou J, Yang C. Fingerprint recognition by combining global structure and local cues, IEEE Transactions on Image Processing 2006; 15(7): 1952-1964. http://dx.doi.org/10.1109/TIP.2006.873443

[9] Wan D, Zhou J. Fingerprint recognition using model-based density map, IEEE Transactions on Image Processing 2006;

15(6): 1690-1696.
http://dx.doi.org/10.1109/TIP.2006.873442

[10] Feng J. Combining minutiae descriptors for fingerprint matching, Pattern Recognition 2008; 41(1): 342-352.
http://dx.doi.org/10.1016/j.patcog.2007.04.016

[11] Choi H, Choi K, Kim J. Fingerprint matching incorporating ridge features with minutiae, IEEE Transactions on Information Forensics and Security 2011; 6 (2): 338-345.
http://dx.doi.org/10.1109/TIFS.2010.2103940

[12] Jain AK, Feng J. Latent fingerprint matching, IEEE Transactions on Pattern Analysis and Machine Intelligence 2011; 33(1): 88-100.
http://dx.doi.org/10.1109/TPAMI.2010.59

[13] Chen F, Zhou J. A comparative study of combining multiple features for fingerprint recognition, in: Emerging Techniques and Challenges for Hand-Based Biometrics (ETCHB), 2010 International Workshop on, 2010, pp. 1-5.
http://dx.doi.org/10.1109/etchb.2010.5559281

[14] Ross A, Nandakumar K, Jain A. Handbook of multibiometrics, Springer, 2006.

[15] Rajanna U, Erol A, Bebis G. A comparative study on feature extraction for fingerprint classification and performance improvements using rank-level fusion, Pattern Analysis and Applications 2010; 13(3): 263-272.
http://dx.doi.org/10.1007/s10044-009-0160-3

[16] Chen F, Huang X, Zhou J. Hierarchical minutiae matching for fingerprint and palmprint identification, Image Processing, IEEE Transactions on 2013; 22(12): 4964-4971.

[17] Prabhakar S, Jain AK. Decision-level fusion in fingerprint verification, Pattern Recognition 2002; 35(4): 861-874.
http://dx.doi.org/10.1016/S0031-3203(01)00103-0

[18] Jia J, Cai L, Lu P, Liu X. Fingerprint matching based on weighting method and the svm, Neurocomputing 2007; 70(4): 849 - 858.
http://dx.doi.org/10.1016/j.neucom.2006.10.028

[19] Maio D, Maltoni D, Cappelli R, Wayman J, Jain A. FVC2002: Second Fingerprint Verification Competition, in: International Conference on Pattern Recognition, 2002, pp. 811-814.
http://dx.doi.org/10.1109/icpr.2002.1048144

[20] Zhou J, Gu J. A model-based method for the computation of fingerprints' orientation field, IEEE Transactions on Image Processing 2004; 13(6): 821-835.
http://dx.doi.org/10.1109/TIP.2003.822608

[21] Zhou J, Chen F, Gu J. A novel algorithm for detecting singular points from fingerprint images., IEEE Transactions on Pattern Analysis and Machine Intelligence 2009; 31(7): 1239-1250.
http://dx.doi.org/10.1109/TPAMI.2008.188

[22] Inc N, Verifinger 6.2 (2010).

[23] Meyer D, Wien FT. Support vector machines, The Interface to libsvm in package e1071.

[24] Tico M, Kuosmanen P. Fingerprint matching using an orientation-based minutia descriptor, IEEE Transactions on Pattern Analysis and Machine Intelligence 2003; 25(8): 1009-1014.
http://dx.doi.org/10.1109/TPAMI.2003.1217604

[25] Chen F, Zhou J, Yang C. Reconstructing Orientation Field From Fingerprint Minutiae to Improve Minutiae-Matching Accuracy, IEEE Transactions on Image Processing 2009; 18(7): 1665-1670.
http://dx.doi.org/10.1109/TIP.2009.2017995

Received on 16-11-2015 Accepted on 23-12-2015 Published on 05-01-2016

http://dx.doi.org/10.15379/2410-2938.2015.02.02.04