# Fingerprint matching based on weighting method and SVM

Jia Jia, Lianhong Cai, Pinyan Lu, Xuhui Liu

Key Laboratory of Pervasive Computing (Tsinghua University), Ministry of Education
Beijing 100084, P.R.China
{jiajia}@mails.tsinghua.edu.cn
{clh-dcs}@tsinghua.edu.cn
{lupinyan}@hotmail.com
{lxh}@csnet1.cs.tsinghua.edu.cn

**Abstract.** Fingerprint verification is an important biometric technology. In this paper, an improved fingerprint matching approach that uses weighting method and support vector machine (SVM) is presented. The traditional minutiae-based matching task is transformed to a classification task in the proposed approach using SVM. Furthermore, a new weight feature is introduced based on the distance between minutiae to supplement the minutiae information, especially for fingerprint images of poor quality. To give an objective assessment of the approach, both international and domestic fingerprint verification competition databases have been used for the evaluation. Experimental results show substantial improvements in the accuracy and performance of fingerprint verification.

## 1 Introduction

Fingerprint authentication is one of the most important biometric technologies [1]. A fingerprint is the pattern of ridges and valleys (furrows) on the surface of the finger. In automatic fingerprint verification system (AFVS), the characteristic features obtained from the test fingerprint are to be matched against those from a template fingerprint. As the fingerprint of a person is unique and immutable, the AFVS can be widely used in both anti-criminal and civilian applications. Therefore, accuracy and performance improvements are the key points in AFVS current research.

The uniqueness of a fingerprint can be determined by the global pattern of ridges and valleys, and by the local pattern of bifurcations and endings which are called minutiae (Fig 1). The minutiae are extracted from the thinned image that is obtained from fingerprint preprocessing [2, 5, 6]. Usually, the similarity between two fingerprints is determined by computing the total number of matched minutiae, the process of which is called minutiae-based [6]. However, general minutiae-based matching algorithms (GMMA) in AFVS only make use of minutiae localizations (positions and orientations).


(a) (b)

**Fig. 1.** Examples of fingerprint minutiae. (a) A ridge ending. (b) A ridge bifurcation

Our main work focuses on the minutiae-based matching scheme. We present a fingerprint matching approach, which uses not only the minutiae localizations, but also a weight feature which is the distance between a minutia and its nearest neighbor minutia. Considering that the matching process can be regarded as a two-class classification problem (matched or not), using the extracted minutiae positions, orientations and weights as features, we define a vector standing for the similarity of two fingerprints, and choose SVM as the classifier. The proposed approach is motivated by the following observations:

(1) The minutiae information in fingerprint images may not be discriminative because of the different sensors and skin conditions. Most of the sensors especially capacitive sensors capture only a small area of the fingertip, which means some minutiae information outside the area is missing. Further in practice, due to variations in skin conditions like postnatal marks or occupational marks, and impression conditions, a significant percentage of fingerprint images are in poor quality. This leads to the problem that a large amount of errors in minutiae positions and orientations may be introduced. In such cases, the weight based on two minutiae's distance is not only an estimate of fingerprint structure, but also a supplement for minutiae information.

(2) After getting the total number $t$ of matched minutiae, a judgment must be made: are these two images matched? The normal method is to compare $t$ with a certain threshold $\lambda$, if $t \geq \lambda$, then the two images are matched, otherwise not. That means, the value of $\lambda$ determines the final conclusion actually. In order to reduce the influence of evaluating $\lambda$, the appropriate way is using a machine learning technique to obtain the threshold for different databases. In addition, SVM is a powerful classification method which can properly label the result with matched or not.

In summary, we present a fingerprint matching scheme that uses both minutiae localizations and estimated weights as features, transforming the matching problem to a classification problem and using SVM to solve it. The experiments with both international and domestic fingerprint verification competition data show substantial improvements in accuracy and performance of fingerprint verification.

The rest of this paper is organized as follows: in section 2 we give the module analysis of GMMA. Section 3 outlines the problems of GMMA and the proposed solution. In section 4, our approach based on weighting method and SVM is presented in detail. Before concluded with discussions, experimental results and analysis are given in section 5.

## 2 Module analysis of general minutiae-based matching algorithms

All the GMMA [4, 5, 6] can be broadly classified into the following stages: (1) extracting minutiae, (2) generating the transformation parameters that relate the test image and the template image, (3) aligning the two images under these parameters to get the total number of matched minutiae, (4) determining the final result according to the result of stage (3).

**Minutiae extraction:** Most feature extraction methods are based on thinned images. The minutiae are detected by tracing the ridge contours. Each minutia is characterized by its location coordinate $(x, y)$ and orientation of the ridge on which it is detected [3].

**Generating transformation parameters:** To ensure an overlap of common regions, the two images need to be aligned first. This is done by determining the transformation parameters $(t_x, t_y, \rho, \theta)$, where $t_x, t_y$ indicate the adjustable distances in x-axis and y-axis, $\rho$ indicates the flex coefficient and $\theta$ indicates the rotated angle. $(t_x, t_y, \rho, \theta)$ is computed by coordinates of two pairs of minutiae (usually delta points and core points) from both images [4]. These two pairs of minutiae are called reference points.

**Aligning the test image and the template image:** Once the transformation parameters $(t_x, t_y, \rho, \theta)$ are obtained, the test image can be aligned. Let $(x, y)$ represent the original coordinate, then the aligned coordinate $(u, v)$, is obtained as Eq.1.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \rho \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \qquad \mathbf{(1)}$$

After the images are aligned, the total number of matched minutiae can be computed.

**Result determination:** Due to the structures of fingerprints themselves and the conditions of sensors, some fingerprint images do not have delta points or center points [5]. In such cases, every two pairs of minutiae from test image and template image, as reference points, should be chosen to get the corresponding transformation parameters $(t_x, t_y, \rho, \theta)$. For different reference points, there will be different numbers of matched minutiae, and the maximum number will be compared with a certain threshold $\lambda$ to decide whether the two images are matched. That means, if we adopt the method of exhaustion, this determining process will need an $n^2m^2$ times of computing, where $n$ and $m$ indicate the number of minutiae of the two images.

## 3 Problem statement and solution

From the analysis of GMMA, we point out three noticeable problems: (1) fake minutiae do have a bad effect on the result. (2) the determining process presented in Section 2 will hurt the algorithm performance. (3) an unsuitable threshold $\lambda$ will lead to a wrong conclusion.

### 3.1 Using weighting method to solve the problem of fake minutiae

Fake minutiae are always from structures like spacings, bridges and pores (Fig 2(a)). Through observation, we find an intercommunity of these structures such that the fake minutiae on them are usually much closer to each other than real ones (Fig 2(b)). In other words, if the distances between a minutia and its neighbors are very short, this minutia may be a fake one.

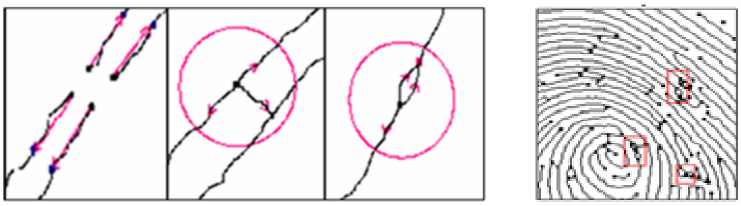

(a) (b)

**Fig. 2.** Examples of fake minutiae. (a) the structures of spacing, bridge and pore. (b) agglomerate fake minutiae points marked by panes

Therefore, to supplement minutiae information, we define a weight feature *w* besides the minutiae localization.

**Definition 1.** A minutia's weight *w* is the distance between it and its nearest neighbor minutia.

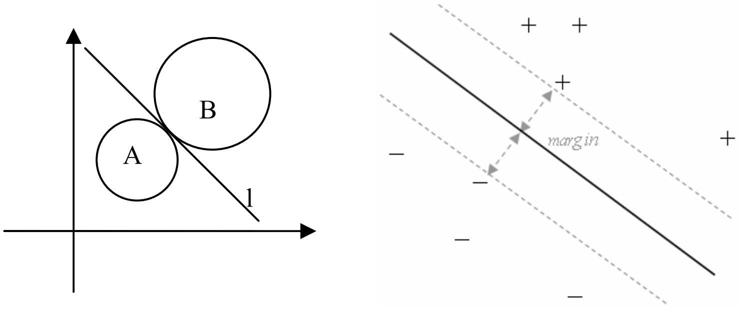
The value of *w* is normalized in the [0， 100] range. For a minutia, the greater the value of its weight *w*, the higher the possibility of being a real one.

### 3.2 Using ranking strategy to improve performance

As presented in Section 2, for different reference points, there will be different transformation parameters and numbers of matched minutiae. Our experiment shows that it will take about 30 seconds to get the maximum number of matched minutiae when using exhaustion algorithm. To reduce the number of operation, we present a ranking strategy that sorts the minutiae by descending order according to their weights *w*, and choose only the top 20 minutiae as the reference points. This strategy can reveal most of the real minutiae. And experiment shows that the computing time for matching two images can reduce to only about 0.5 second, which means the performance is significantly improved.

### 3.3 Using SVM to solve the problem of result determination

GMMA normally determine the final result by comparing threshold $\lambda$ and the total number of matched minutiae *t*. That is a one-dimension method using two numbers, which means the value of $\lambda$ plays a key role in determining the final result. To solve the problem of evaluating the threshold $\lambda$, we present a method that uses a hyperplane and a set of matching vectors which stand for the similarity of fingerprints. That means we transform the problem which is hard to be solved in one-dimension space to a multidimensional space. This can be explained clearly by Fig 3(a).



(a) (b)

**Fig. 3.** Explanations for SVM. (a)For circles A and B, their projection to either axis has superposition, but in two-dimensional space, they are linear- dividable, which means the line *l* can divide them. (b)An SVM is a hyperplane that separates the positive and negative examples with maximum margin. The examples closest to the hyperplane are called support vectors

Therefore, we choose SVM which has shown outstanding classification performance in practice as the classifier [7, 8]. SVM is based on a solid theoretical foundation – *structural risk minimization* [9], and its simplest linear form is shown in Fig 3(b). Large margin between positive and negative examples has been proven to lead to good generalization [9].

## 4 A matching scheme based on weighting method and SVM

Before matching steps, fingerprint preprocessing must be accomplished first. Here we use an enhancing algorithm based on estimated local ridge orientation and frequency, and filter the image by Gabor filter [2]. Our matching approach includes the following stages:

### 4.1 Extracting the minutiae features

For each minutia, we define a 5-tuple (*x, y, type, theta, w*) to describe its features. The *x* and *y* indicate the minutia's coordinate. The value of *type*, 1 or 2, indicates that the minutia is an ending or a bifurcation. The *theta* indicates the tangent angle of the ridge where the minutia is located. And *w* indicates the minutia's weight. The 5-tuple (*x, y, type, theta, w*) is obtained by the following steps (Fig 4):

Let $\boldsymbol{I}_{test}$ represents the test image, $\boldsymbol{I}_{temp}$ represents the template image. For $\boldsymbol{I}_{test}$ and $\boldsymbol{I}_{temp}$:

*Step 1:* Do image normalization, estimate the local orientation and frequency, filtering, and then get the threshold image [2].

*Step 2:* Do ridge thinning, and extract the coordinate (*x, y*) and orientation *theta* of each minutia.

*Step 3:* Compute the distance between each minutia and its nearest neighbor minutia as its weight $w$.

*Step 4:* Sort the minutiae by descending order according to their weights $w$.

That means, for each fingerprint image, we get a minutiae sequence $p_1, p_2......p_n$ with degressive weights.


(a) (b) (c)

**Fig. 4.** Fingerprint image preprocessing and minutiae extraction. (a)raw image.(b)threshold image.(c)thinned image with minutiae

### 4.2 Matching the minutiae under the optimal transformation parameters

Let $p_1, p_2......p_n$ represent the minutiae sequence of $\boldsymbol{I}_{test}$ and $q_1, q_2......q_n$ represent the sequence of $\boldsymbol{I}_{temp}$. We compute the total number of matched minutiae by the following steps:

*Step 1:* Choose the top 20 minutiae $\{p_i\}$ $(1 \le i \le 20)$ and $\{q_j\}$ $(1 \le j \le 20)$ from the two sequences as the reference points.

*Step 2:* For $p_i$, $q_j$ $(1 \le i, j \le 20)$, if they have the same value of *type*, add $(p_i, q_j)$ to set **A**.

For every two members $(p_{i1}, q_{j1}), (p_{i2}, q_{j2}) \in \mathbf{A}$, do *Step3* to *Step5*.

*Step 3:* Compute the transformation parameters $(t_x, t_y, \rho, \theta)$ according to $(p_{i1}, q_{j1})$ and $(p_{i2}, q_{j2})$ by Eq. (1).

*Step 4:* Select $(t_x, t_y, \rho, \theta)$, if $|\rho - 1| > 0.1$ or $\theta > \pi/3$, skip *Step5*. This selection ensures that the flex coefficient $\rho$ approximates to 1 and the rotation angle $\theta$ is less than $\pi/3$.

*Step 5:* For $p_1, p_2......p_n$, compute their new coordinates and orientations according to Eq. (1), then match them with $q_1, q_2......q_m$. Record the number of matched minutiae pairs.

When every two members $(p_{i1}, q_{j1}), (p_{i2}, q_{j2}) \in \mathbf{A}$ have been chosen to finish the *Step3* to *Step5*, record the maximum number of matched minutiae pairs $t$ and the corresponding translation parameters $(t'_x, t'_y, \rho', \theta')$.

### 4.3 Determining the results using matching vector and SVM

We define a matching vector $\boldsymbol{V}(n, m, t, ave, err)$ to describe the similarity of $\boldsymbol{I}_{test}$ and $\boldsymbol{I}_{temp}$. This vector $\boldsymbol{V}$ is obtained by the next steps:

*Step 1:* Record the minutiae number $n$ of $\boldsymbol{I}_{test}$ and $m$ of $\boldsymbol{I}_{temp}$, and maximum matched minutiae number $t$.

*Step 2:* For the matched minutiae pairs, the weights of the minutiae in $\boldsymbol{I}_{test}$ are $v_1, v_2, \cdots, v_t$ and in $\boldsymbol{I}_{temp}$ are $u_1, u_2, \cdots, u_t$. Let *ave* represent an average of the weight values of these matched minutiae, and *ave* is calculated by Eq.2.

$$ave = \sum_{i=1}^{t} \frac{2u_i v_i}{u_i + v_i} \tag{2}$$

*Step 3:* Get a $100\text{pixel} \times 100\text{pixel}$ sub-image $\boldsymbol{I}_{sub}$ from the center of the threshold image of $\boldsymbol{I}_{test}$. Translate $\boldsymbol{I}_{sub}$ by parameters $(t'_x, t'_y, \rho', \theta')$ to $\boldsymbol{I}_{sub}'$, and let *err* represent the number of pixels in $\boldsymbol{I}_{sub}'$ that have the same intensity as its corresponding pixel in $\boldsymbol{I}_{temp}$. The *err* describes an estimation of the matching error.

For a matching vector $\boldsymbol{V}(n, m, t, ave, err)$, we need to label it with “matching success” or “matching failure”. The decision function of an SVM is shown in Eq.3.

$$f(\boldsymbol{V}) = \langle \boldsymbol{w} \bullet \boldsymbol{V} \rangle + b \tag{3}$$

$\langle \boldsymbol{w} \bullet \boldsymbol{V} \rangle$ is the dot product between w (the normal vector to the hyperplane) and $\boldsymbol{V}$ (the matching vector). The margin for an input vector $\boldsymbol{V}_i$ is $y_i f(\boldsymbol{V}_i)$ where $y_i \in \{-1, 1\}$ is the correct class label for $\boldsymbol{V}_i$. Seeking the maximum margin can be expressed as minimizing $\langle \boldsymbol{w} \bullet \boldsymbol{w} \rangle$ subject to $y_i(\langle \boldsymbol{w} \bullet \boldsymbol{V}_i \rangle + b) \geq 1, \forall i$. We allow but penalize the examples falling to the wrong side of the hyperplane.

## 5 Experiments and discussion

We conducted experiments with data of fingerprint verification competitions, to demonstrate the advantages of our proposed approach to fingerprint verification.

### 5.1 Datasets

We have collected 5 datasets from FVC2002[1] (The Second International Fingerprint Verification Competition) and BVC2004[2] (Biometrics Verification Competition 2004). In order to prove the influence of different image qualities to our matching approach, 4 subsets of BVC2004 are chosen as db1 to db4. And db5, which is a subset of FVC2002, is chosen to prove the influence of image amount. The information of each dataset, including the number of different fingers and total images, sensor types, image size, etc. is shown in Table 1.

---

[1] http://bias.csr.unibo.it/fvc2002/

[2] http://www.sinobiometrics.com/chinese/conferences/sinobiometrics%2704.htm

**Table 1.** The information of datasets

| | The source of the datasets | different fingers / total images | Sensors | Image size | Resolution |
|---|---|---|---|---|---|
| 1st db | BVC2004 DB1 | 40/400 | Optical sensor | 412 x 362 | 500 dpi |
| 2nd db | BVC2004 DB2 | 40/400 | CMOS sensor | 256 x 300 | 500 dpi |
| 3rd db | BVC2004 DB3 | 40/400 | Thermal sweeping sensor | 300 x 480 | 500 dpi |
| 4th db | BVC2004 DB4 | 40/400 | Fingerpass | 380 x 460 | 500 dpi |
| 5th db | FVC2002 DB1 | 230/1840 | Optical sensor | 388 x 374 | 500 dpi |

Each fingerprint image allows a rotation angle that belongs to $[-\pi/4, \pi/4]$ (compared with the vertical line). Every two images from one finger have an overlap of common region. But there may be no delta points or core points in some fingerprint images.

### 5.2 Experiments setup

We posed 2 experiments. For db1 to db5, we compared our approach with GMMA mentioned in Section 2. Both of them use the same image preprocessing algorithms [2]. Both the experiments are done by the method of 5-folder cross validation, but have differences in the size of test sets and training sets.

**Experiment 1.** For db1 to db4, 400 images are divided into 5 parts, each of which has 80 images. Both the algorithms run five times. For each time, **four** of the five parts are used as training sets (our approach only)**,** and the other **one** part is used as test set. The averaged verification result will be reported over these 5 times.

**Experiment 2.** For db5, 1840 images are divided into 5 parts, each of which has 368 images. Both the algorithms run five times. For each time, **one** of the five parts is used as training set (our approach only), and the other **four** parts are used as test sets. The averaged verification result will be reported over these 5 times.

We use SVMlight[3] for the implementation of SVM [8, 10], and take linear kernel in experiments.

### 5.3 Measures

The accuracy and performance of a fingerprint verification algorithm can be measured by FNMR (False Non Match Rate: each sample in the subset A is matched against the remaining samples of the same finger), FMR (False Match Rate: the first sample of each finger in the subset A is matched against the first sample of the remaining fingers in A) and the average time of matching two images. Especially for Experiment 2, we also measure the maximum memory size of our approach. The configuration of running computer is PIV1.0G, 256M DDR.

---

[3] http://svmlight.joachims.org/

### 5.4 Results and discussion

**Table 2.** Experimental results of db1 to db4

| | FNMR | | FMR | Avg match time (our approach) |
|---|---|---|---|---|
| | Our approach | GMMA | | |
| 1st database | 2.03% | 6.67% | ~ 0 | 1.003 sec |
| 2nd database | 12.78% | 30.28% | ~ 0 | 0.603 sec |
| 3rd database | 7.78% | 24.17% | ~ 0 | 1.024 sec |
| 4th database | 5.28% | 13.05% | ~ 0 | 0.831 sec |

**Table 3.** Experimental results of db5

| FNMR | | FMR | Avg match time (our approach) | Max match Mem (our approach) |
|---|---|---|---|---|
| Our approach | GMMA | | | |
| 4.08% | 13.84% | ~ 0.1% | 0.583sec | 6028 Kbytes |

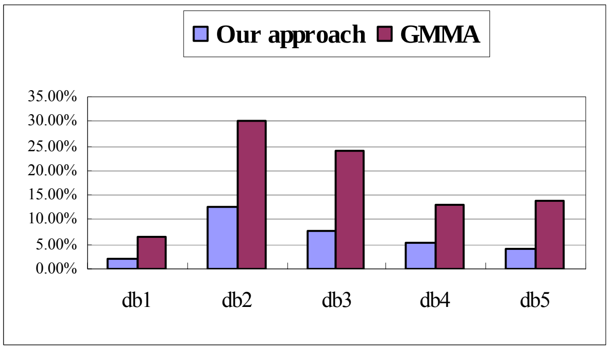


**Fig. 5.** Comparing the FNMR of our approach and GMMA in five different datasets

The experimental results of db1 to db4 are shown in Table 2. We see that our approach really can achieve much better accuracy than GMMA for fingerprint verification, and the average time for matching a pair of fingerprints is also acceptable. As shown in Table 1, fingerprints of these four datasets are captured by sensors of different types. So the images have different qualities. This strongly suggests that our feature extraction and SVM methods capture well the information needed for fingerprint verification, and have a low influence by fingerprint image quality.

The experimental results of db5 are shown in Table 3. We see that although the proportion of training sets is reduced, and the number of test members is increased, our approach still works better than GMMA for fingerprint verification. The average time and the maximum memory for matching a pair of fingerprints are also acceptable. We think this is because SVM makes effective use of the matching vectors to enhance classification.

The experimental results of our approach and GMMA are compared in Figure 5. It is clear that our matching approach based on weighting method and SVM outperforms GMMA consistently and significantly.

## 6 Conclusion

Our main contributions to fingerprint verification are: 1) transforming the traditional minutiae-based matching task to a classification task, and using a powerful classifier SVM to solve it; 2) proposing a weight feature to supplement minutiae information for fingerprint images of poor quality.

Future work may include: considering use of TSVM (transductive SVM) [10] instead of SVM to improve the matching accuracy and performance, because TSVM is especially beneficial when the number of training examples is small, and so forth.

## References

1. Newham, E.: The Biometric Report. SJB Services. New York (1995)
2. Lin, H., Yifei W., Jain, A.: Fingerprint Image Enhancement: Algorithm and Performance Evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No.8 (1998)
3. Ratha, N.K., Chen, S., Jain, A.: Adaptive Flow Orientation-Based Feature Extraction in Fingerprint Images. Pattern Recognition, Vol.28, No.11 (1995)
4. Chongwen, W., Jianwei, L.: Fingerprint Identification Using Point Pattern Matching. Journal of Chongqing University (Natural Science Edition), Vol. 25, No.6 (2002)
5. Jain, A., Lin, H.: On-line fingerprint verification. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No.4 (1997) 302-314
6. Jain, A., Ross A., Prabhakar S.: Fingerprint Matching using Minutiae and Texture Features. In: International Conference on Image Processing (ICIP). Thessaloniki, Greece (2001) 282-285
7. Cristianini, N., Shawe-Taylor J.: An Introduction to Support Vector Machines. Cambridge University Press. Cambridge, UK (2000)
8. Yuan, Y., Paolo, F., Massimiliano P.: Fingerprint Classification with Combinations of Support Vector Machines. Lecture Notes in Computer Science, Vol. 2091. Springer-Verlag, Berlin Heidelberg New York (2001) 253–258
9. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag, Berlin Heidelberg New York (2000)
10. Joachims, T.: Transductive Inference for Text Classification using Support Vector Machines. In: Proceedings of the 16th International Conference on Machine Learning (ICML). Bled, Slovenia (1999) 200-209