Fingerprint matching
based on weighting method and the SVM

Jia Jia*, Lianhong Cai, Pinyan Lu, Xuhui Liu

Key Laboratory of Pervasive Computing (Tsinghua University), Ministry of Education
Beijing 100084, P.R.China

Abstract

Fingerprint verification is an important biometric technology. In this paper, an improved fingerprint matching approach that uses both the weighting method and the support vector machine (SVM) is presented. A new weighting feature based on the distance between minutiae is introduced to supplement the minutiae information, which is particularly useful for fingerprint images of poor quality. Furthermore, the traditional minutiae-based matching task is studied as a classification task in the proposed approach by using SVM. 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.

Keywords: Fingerprint verification; Minutiae-base; weighting feature; SVM

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 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 where precision is important. 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). These two types of minutiae are considered by the Federal Bureau of Investigation for identification purposes [2]. The minutiae are extracted from the thinned image obtained from fingerprint preprocessing [3, 4, 5]. Usually, the similarity between two fingerprints is determined by computing the total number of matching minutiae, and the corresponding process is called minutiae-based matching

*Corresponding author.
E-mail address: jiajia@mails.tsinghua.edu.cn (J.Jia)

[5]. However, general minutiae-based matching algorithms (GMMAs) 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 that 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 (two fingerprint images are either 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, particularly capacitive sensors, capture only a small area of the fingertip, which means some minutiae information outside the area is missing. Furthermore, in practice a significant percentage of fingerprint images are of poor quality due to variations in skin conditions like postnatal marks or occupational marks, and impression conditions. This may lead a large number of errors in minutiae positions and orientations, which may cause problems in next matching stage. 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 matching minutiae, a judgment must be made: do these two images match? The normal method is to compare t with a certain threshold λ. If $t \geq \lambda$, then the two images match. If t is not greater than or equal to the threshold, than the images do not match. That means the value of λ is critical in the decision making process. In order to reduce the influence of λ, a machine learning technique can be used to determine the threshold for different databases. In addition, SVM is a powerful classification method that can properly label matching results.

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 the next section we introduce important background information. In section 3 we give the analysis of GMMA. Section 4 outlines the problems of GMMA and the proposed solution. In section 5, our approach based on weighting method and SVM is presented in detail. Before concluded with discussions, experimental results and analysis are given in section 6.

2. Related works

Fingerprint matching techniques may be broadly classified as being either minutiae-based or image-based [6] (for a good survey see [7]). Minutiae-based approaches first extract the minutiae from the fingerprint images; then, the matching between two fingerprints is made using the two sets of minutiae localizations. Image-based approaches usually extract the features directly from the raw image since a grey-level fingerprint image is available; then, the matching decision is made using these features. AFVSs are usually based on minutiae matching [8, 9, 10, 11]. Minutiae-based fingerprint matching approach involves different stages (see Fig.2 for an illustration).

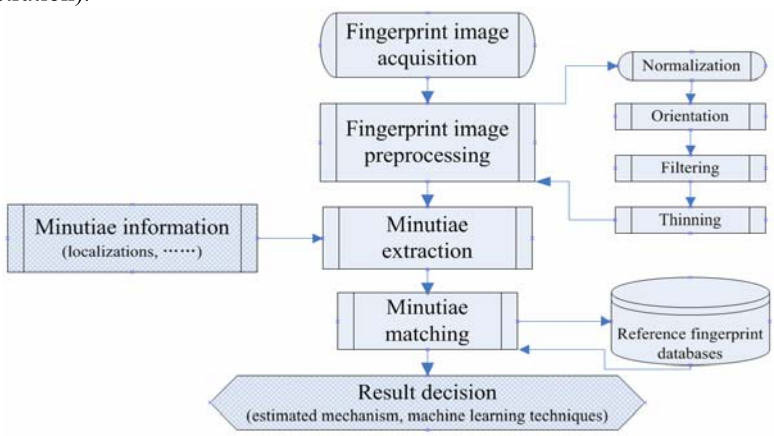


Fig. 2. A flowchart showing different phases of minutiae-based fingerprint matching approach. The highlighted modules show the area of our work.

Our main work focuses on minutiae extraction and result decision stages.

Most of the fingerprint minutiae extraction methods are thinning-based by which the skeletonization process converts each ridge contour to one pixel wide. The minutiae points are detected by tracing the thin ridge contours. When the trace stops, a ridge ending point is marked. Bifurcation points are those with more than two neighbors [12, 13, and 14]. In practice, thinning methods have been found to be sensitive to noise, and the skeleton structure does not match up with the intuitive expectation.

Several approaches to automatic minutiae extraction have been proposed in the literature using different types of enhancement approaches. O'Gorman and Nickerson [15] present a technique for enhancement based on convolution of image with a filter oriented according to the ridge dominant direction. Sherlock et al. [16] and Lee and

Wang [17] define a technique for fingerprint enhancement and binarization that performs a frequency domain filtering through position-dependent filters [18]. The most recent works reported in the literature [3] are based on the usage of Gabor filters. Lee and Wang [17] proposed a Gabor-filter-based method for fingerprint recognition. Jain et al. [3] present a fast fingerprint enhancement algorithm for use by Gabor filters. Although the third method employs an isotropic constructing element and as a result keeps the original shape of the fingerprint, the impulsive noise cannot be completely eliminated. However, the time complexity of these three is at least $O(N^2 d^2)$ for an image with $N \bullet N$ pixel entries and a filter whose radius is d. It is time consuming. In this paper, we first propose a simple and linear time complexity method, which employs generalized morphological operators (GMO) [19] based on distance transform and integral image [20], to eliminate the impulsive noise.

In the result decision stage, most of minutiae-based fingerprint matching approaches use estimate mechanisms to determine the threshold. In addition to the estimate mechanisms, a great deal of literature has presented some ingenious technologies related to the use of machine learning methods for final decision making. From a practical point of view, fingerprint verification has also been studied as a two class classification problem by using a number of machine learning paradigms, for example: neural networks, decision trees and support vector machines (SVM) [21, 22, 23, 24, 25]. For example, the approach presented in [26] combines the features extracted by FingerCode with Parzen Window Classifier (PWC) [27]. These studies have shown performance gains with trained classifiers, and favored support vector machines over neural networks and decision trees.

3. General Minutiae-based Matching Algorithm

All the GMMAs [4, 5, and 28] can be generally 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 matching minutiae, and (4) determining the final result according to the result of stage (3).

Extracting minutiae: Most feature extraction methods are based on thinned images. The minutiae are detected by tracing the ridge contours. The features include: each minutia's location coordinate (x, y) and orientation of the ridge on which the minutia is detected [29].

Generating transformation parameters: To ensure that the common regions overlap, the two images need to be aligned first. This is done by determining the transformation parameters (t_x, t_y, ρ, θ), where t_x, t_y indicate the adjustable distances in x-axis and y-axis, ρ indicates the flex coefficient and θ indicates the rotated angle. (t_x, t_y, ρ, θ) is computed by coordinates of two pairs of minutiae (usually delta points and core points) from both images [28]. 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, ρ, θ) 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.

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

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

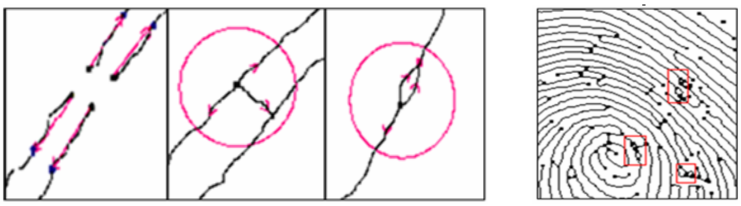
Determining the results: Due to the structures of fingerprints themselves and the conditions of sensors, some fingerprint images do not have delta points or center points [28]. When this occurs, every possible two pairs of minutiae from the test image and template image are used as reference points to obtain all of the potential corresponding transformation parameters (t_x, t_y, ρ, θ). For different reference points, there will be different numbers of matching minutiae, and the maximum number will be compared with a certain threshold λ to decide whether the two images are matched. That means if we adopt the method of exhaustion, this determining process will need to process $O(n^2m^2)$ calculations, where n and m indicate the number of minutiae of the two images.

4. Problem statement and solution

From the analysis of GMMA, we find three noticeable problems: (1) fake minutiae do have a negative effect on the result; (2) the process of determination presented in Section 2 will hurt the algorithm performance; and (3) an unsuitable threshold λ will lead to an incorrect conclusion.

4.1. A new weighting method for solving the problem of fake minutiae

Fake minutiae always come from structures like spacings, bridges and pores created during the thinning process (Fig 3(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 3(b)). In other words, if the distances between a minutia and its neighbors are very small, the minutia in question may be a fake one.



(a) (b)

Fig. 3. 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 into [0，1]. For a minutia, the greater the value of its weight *w* is, the greater the probability of it being a real.

4.2. A new ranking strategy for improving performance

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

To further improve the performance, the Least Square Method is used to obtain transformation parameters (t_x, t_y, ρ, θ). Let P= $\{p_i\}$ represents the minutiae sequence of test image and Q= $\{q_i\}$ represents the sequence of template image (Eq.2).

$$\begin{aligned} P &= \{(x_1, y_1), \ldots (x_n, y_n)\} \\ Q &= \{(u_1, v_1), \ldots (u_n, v_n)\} \end{aligned} \tag{2}$$

where *n* represents the number of minutiae in a image. Let $a = \rho\cos\theta, b = \rho\sin\theta$, $e = t_x$, $f = t_y$, then Eq.1 can be described as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \tag{3}$$

If $T_{rans}(x_i, y_i)^T = (u_i^{'}, v_i^{'})^T$, then the square error ε can be given by:

$$\begin{aligned} \varepsilon^2 &= \sum_{i=1}^{n}((u_i - u_i^{'})^2 + (v_i - v_i^{'})^2) \\ &= \sum_{i=1}^{n}((u_i - ax_i - by_i - e)^2 + (v_i + bx_i - ay_i - f)^2) \end{aligned} \tag{4}$$

Using Least Square Method, the transformation parameters of *a*, *b*, *c*, *f* are

$$\begin{bmatrix} a & b & e & f \end{bmatrix}^T = A^{-1}C \tag{5}$$

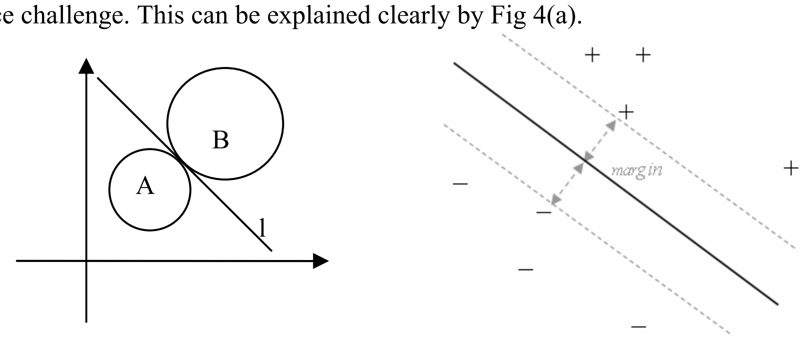
where *A* and *C* can be computed as:

$$
A = \begin{bmatrix} \sum (x_i^2 + y_i^2) & 0 & \sum x_i & \sum y_i \\ 0 & \sum (x_i^2 + y_i^2) & \sum y_i & -\sum x_i \\ \sum x_i & \sum y_i & n & 0 \\ \sum y_i & -\sum x_i & 0 & n \end{bmatrix}
$$

$$
C = \begin{bmatrix} \sum (u_i x_i + v_i y_i) \\ \sum (u_i y_i - v_i x_i) \\ \sum u_i \\ \sum v_i \end{bmatrix} \qquad \textbf{(6)}
$$

4.3. SVM classifier

GMMAs normally determine the final result by comparing threshold λ to the total number of matching minutiae *t*. That is a one-dimension method, which means the value of λ plays a key role in determining the final result. Instead of estimating a threshold λ , 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 solved in one-dimensional space into a multidimensional space challenge. This can be explained clearly by Fig 4(a).



(a) (b)

Fig. 4. Explanation 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 [30, 31]. SVM is based on a solid theoretical foundation – *structural risk minimization* [32], and its simplest linear form is shown in Fig 4(b). The large margin between positive and negative examples has been proven to lead to good generalizations [32].

5. A matching scheme based on weighting method and SVM classifier

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 using Gabor filters [3]. Our matching approach is outlined in the following sections.

5.1. 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 whether the minutia is an ending or a bifurcation. *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 using the following steps (Fig 5):

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

For $\boldsymbol{I}_{test}$ and $\boldsymbol{I}_{temp}$:

Step 1: Perform image normalization, and then estimate the local orientation and frequency. After that, a threshold image can be obtained by filtering the image[3].

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

Step 3: Compute the distance between each minutia and its nearest neighbor minutia to obtain each minutiae's 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. 5. Fingerprint image preprocessing and minutiae extraction. (a)raw image.(b)threshold image.(c)thinned image with minutiae.

5.2. Matching the minutiae under the optimal transformation parameters

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

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

Step 2: For p_i, q_j ($1 \leqslant i, j \leqslant l$), 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}) $\in$ **A**, do *Step3* to *Step5*.

Step 3: Compute the transformation parameters (t_x, t_y, ρ, θ) according to (p_{i1}, q_{j1}) and (p_{i2}, q_{j2}) by Eq.5 and Eq.6.

Step 4: Select (t_x, t_y, ρ, θ), if $|\rho - 1| > 0.1$ or $\theta > \pi/3$, skip *Step5*. This selection ensures that the flex coefficient ρ approaches 1 and the rotation angle θ 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 matching minutiae pairs.

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

Algorithm minutiae matching

Objective: compute the total number of matching minutiae between the test image and the template image

Input:

I_{test} : the test image.

I_{temp} : the template image.

$\{p_n\}$: the minutiae sequence of I_{test} with degressive weights.

$\{q_m\}$: the minutiae sequence of I_{temp} with degressive weights.

Output:

the maximum number of matching minutiae pairs: t.

Method:

```
{
    int i, j, t[i];
    int l=20;
    For (1≤ i≤ l , 1≤j≤ l)
        if type [p_i] = type [q_j]
            add (p_i , q_j) to set A
    For ( ∀ (p_i1, q_j1), (p_i2, q_j2) ∈ A)
        {
            compute (t_x ,t_y, ρ, θ) by Eq. 5 and Eq.6;
            if(|ρ-1| ≤ 0.1 || θ ≤ π/3) )
```

```
        {
            For (0<i<n+1)
                {
                compute (x_i , y_i , theta_i)_new by Eq. 1;
                {p_i}_new = {(x_i , y_i , theta_i)_new };
                }
            match {p_i}_new with {q_j};
            t[i] = the number of matching minutiae pairs;
        }
    }
  t = Max ( {t[i]});
  return t;
}
```

5.3. 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 $\boldsymbol{I}_{temp}$. Vector $\boldsymbol{V}$ is obtained using the following steps:

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

Step 2: For the matching minutiae pairs, let $v_1, v_2, \cdots, v_t$ present the weights of the minutiae in $\boldsymbol{I}_{test}$ and $u_1, u_2, \cdots, u_t$ present the weights of the minutiae in $\boldsymbol{I}_{temp}$. Let ave represents an average of the weight values of these matching minutiae, and ave is calculated by Eq.7.

$$ave = \sum_{i=1}^{t} \frac{2u_i v_i}{u_i + v_i} \tag{7}$$

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.8.

$$f(\boldsymbol{V}) = \langle \boldsymbol{w} \bullet \boldsymbol{V} \rangle + b \tag{8}$$

$\langle \boldsymbol{w} \bullet \boldsymbol{V} \rangle$ 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 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.

The flowchart of our approach is shown as Fig.6.

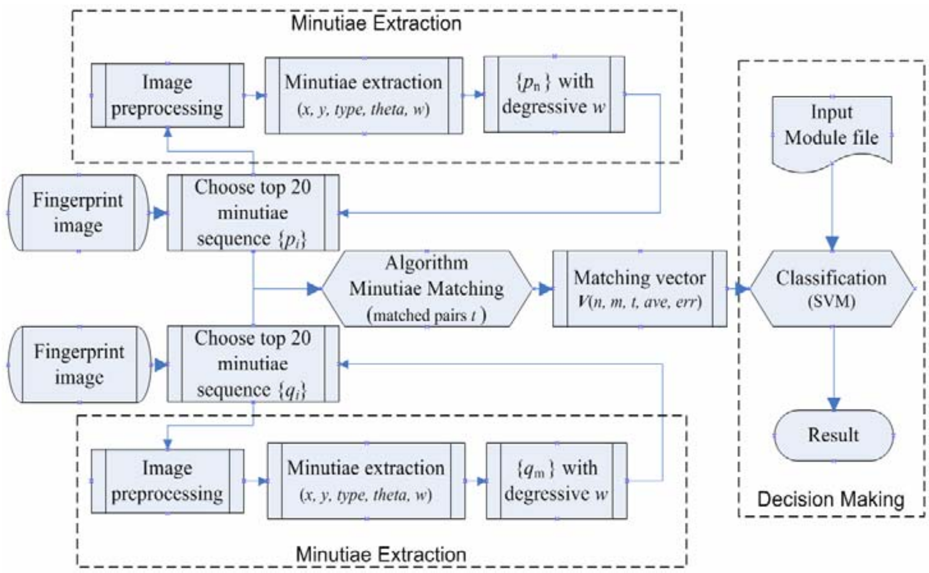


Fig. 6. A flowchart showing different phases of our approach.

6. Experiments and discussions

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

6.1. 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 were selected and named db1 to db4. Three of these

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

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

databases were acquired by various sensors including low-cost and high quality, optical and capacitive. The fourth database contains synthetically generated images. The fifth database db5, which is a subset of FVC2002, was chosen to prove the influence of number of fingerprint images. For each dataset, we show in Table 1 the number of different fingers and total fingerprint images.

Table 1
The number of fingerprint images for each dataset

	The source of the datasets	The number of different fingers	Total images
1st database	BVC2004 DB1	40	400
2nd database	BVC2004 DB2	40	400
3rd database	BVC2004 DB3	40	400
4th database	BVC2004 DB4	40	400
5th database	FVC2002 DB1	230	1840

For each dataset, we show in Table 2 the information of sensors and the condition of images.

Table 2
The information of sensors and images for each dataset

	Sensor type	Manufacturer	Image size	Resolution
1st database	Optical sensor	URU 4000	412 x 362	500 dpi
2nd database	CMOS sensor	Fujitsu MBF200	256 x 300	500 dpi
3rd database	Thermal sweeping Sensor	Atmel	300 x 480	500 dpi
4th database	—	Fingerpass	380 x 460	500 dpi
5th database	Optical sensor	TouchView II	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.

6.2. Experimental setup

We conducted 3 experiments. All the matching approaches in our experiments used the same image preprocessing algorithms [3]. All the experiments were done by the method of 5-folder cross validation.

For db1 to db4, 400 images were divided into 5 parts, each of which had 80 images. All the algorithms were run five times. Each time the algorithms were run, **four** of the five parts were used as training sets**,** and the other **one** part was used as the test set. The average verification results were reported over each of these 5 trials. For db5, 1840 images were divided into 5 parts, each of which had 368 images. All the algorithms were run five times. Each time, **one** of the five parts was used as training set, and the other **four** parts were used as test sets. The average verification results were reported over each of these 5 trials.

Experiment 1. We compared our approach with GMMA mentioned in Section 2. In GMMA, every two minutiae pairs extracted from test image and template image, as reference points, were chosen to get the corresponding transformation parameters (t_x, t_y, ρ, θ) and the number of matching minutiae. The maximum number of matching minutiae pairs was compared with a certain threshold λ to decide whether the two images are matched. Now we just set λ =13 manually.

Experiment 2. We compared our approach to our approach without the weighting method.

Experiment 3. We evaluated our approach's performance by choosing top l (l=10, 20, 30, n) minutiae from minutiae sequence p_1, p_2.....p_n with degressive weights.

Our matching approach was written in C++ and compiled by Microsoft Visual Studio 6.0 (VC++ 6.0). All the experiments were performed on the same computer. The configuration of the running computer is PIV1.0G, 256M DDR. The operating system in use was Microsoft Windows XP.

We used SVMlight[3] for the implementation of SVM [31], and took linear kernel in experiments.

6.3. Measures

The accuracy of the fingerprint verification algorithm was 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 in our experiments. Especially for database 5, we also measure the maximum memory size of our approach.

6.4. Experimental Results and discussions

We first performed five datasets of experiments, all of which examined the accuracy of our approach compared with GMMA. Table 3 and Table 4 show the FNMR and FMR results of db1 to db5.

Table 3
FNMR and FMR obtained from db1 to db4

	FNMR		FMR
	Our approach	GMMA	
1st database	2.03%	6.67%	~ 0

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

2nd database	12.78%	30.28%	~ 0
3rd database	7.78%	24.17%	~ 0
4th database	5.28%	13.05%	~ 0

Table 4
FNMR and FMR obtained from db5

FNMR		FMR	Max match memory (our approach)
Our approach	GMMA		
4.08%	13.84%	0.1%	6028 Kbytes

The FNMR and FMR results of db1 to db4 are shown in Table 3. We see that our approach achieves much better accuracy than GMMA for fingerprint verification. As shown in Table 2, sensors of different types capture fingerprints of these four datasets, so the images are of varying degree of quality. This strongly suggests that our feature extraction and SVM methods capture well the information needed for fingerprint verification, and are not significantly influenced by fingerprint image quality.

The FNMR and FMR results of db5 are shown in Table 4. 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, 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.

Then we turn to examine the accuracy of our approach, compared with our approach without weighting method. The experimental results are shown in Figure 7.

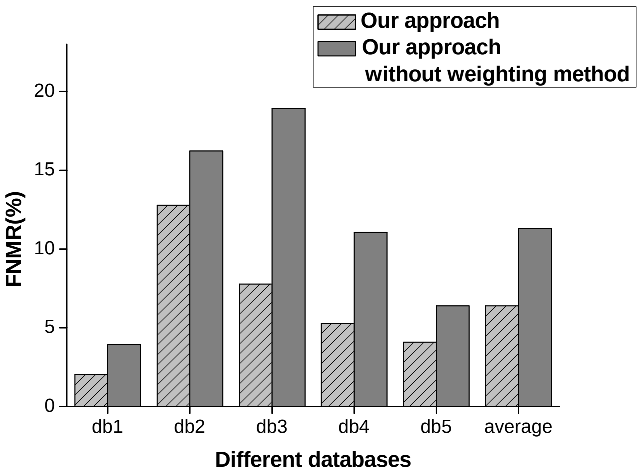


Fig. 7. Comparison the FNMR of our approach and GMMA in five different datasets.

We can see that our approach outperforms our approach without the weighting method in all the five databases. And the FNMR result is 4.92% better when using minutiae weighting method on average. It shows that the minutiae weighting method is very effective in decreasing the influence of fake minutiae.

We also evaluated our approach's performance by choosing different number l of minutiae from minutiae sequence: l=10, 20, 30 and n. Their performances on db1 and db2 are displayed in Table 5 and Fig.8.

Table 5
FNMR and FMR obtained from db1 and db2 choosing different minutiae number l from minutiae sequence p_1, p_2.....p_n with degressive weights.

	FNMR				FMR
	l_1=10	l_2=20	l_3=30	l_4=n	
1st database	6.75%	2.03%	3.31%	3.93%	~ 0
2nd database	29.05%	12.78%	15.17%	16.23%	~ 0

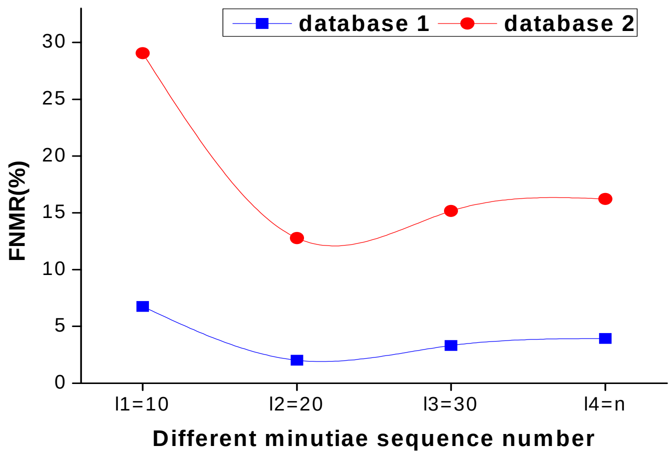


Fig. 8. Comparison the FNMR of our approach using different minutiae sequence number l in two different datasets.

It's shown that the result is the best when l is assigned to 20 for both datasets. When l is set more than 20, the FNMR will be a little higher. The reason is more fake minutiae are evolved in the matching process.

And when l is set to 10, the results are also not good because the minutiae selected from the two images are not matched even though they are usually all real minutiae.

Table 6
Average time of matching two fingerprints obtained from db1 to db5

	1st database	2nd database	3rd database	4th database	5th database
Our Approach	1.003 sec	0.603 sec	1.024 sec	0.831 sec	0.583sec
GMMA			>= 30 sec		

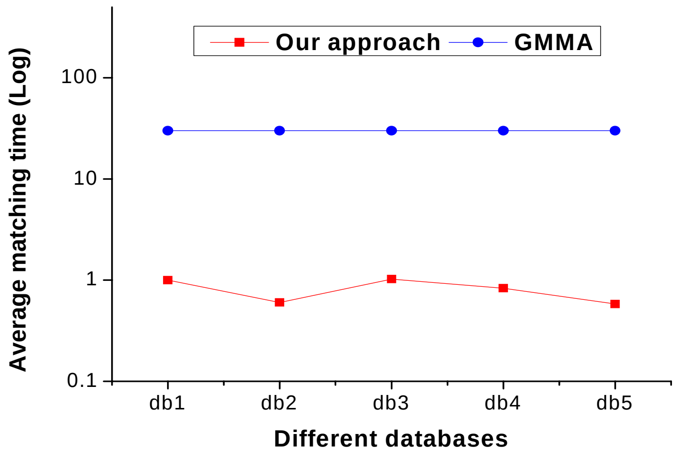


Fig. 9. Comparison the average matching time of our approach and GMMA in five different datasets.

The time performance obtained from db1 to db5 are shown in Fig.9. The average time for matching two fingerprints using our approach is only one thirtieth less than GMMA. This strongly suggests that our weighting method improves time performance significantly. In addition, the creation of the template of a pair of fingerprints (training stage) takes ~ 4 seconds on a PIV1.0G, 256M DDR computer, and the creation of the template of a pair of fingerprints (testing stage) takes ~ 0.5 seconds on the same computer.

It must be mentioned that our matching approach successfully uses weighting features and ranking strategy to avoid detecting core points in two fingerprints aligning process. Core point detection is mostly used to align the test image and the template image. But in some images the core point may be present close to the boundary of the image, or may not even be present [33], which will lead to an error in the final decision.

7. Conclusions

Our main contributions to fingerprint verification in this paper include: 1) A weighting method was proposed to supplement minutiae information for fingerprint images of poor quality; 2) The traditional minutiae-based matching task was translated to a classification task using a powerful SVM classifier.

Future work may include the examination of larger databases to verify the performance of our approach, and how to use transductive SVM (TSVM) [34] instead of SVM to improve the matching performance.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments and suggestions.

References

[1] E.Newham, The Biometric Report, SJB Services, New York (1995).

[2] Federal Bureau of Investigation (FBI), The Science of Fingerprints: Classification and Uses, US Government Printing Office, Washington, DC, 1984.

[3] L. Hong, Y.F. Wang, A. Jain, Fingerprint Image Enhancement: Algorithm and Performance Evaluation, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No.8 (1998).

[4] A. Jain, L. Hong, On-line fingerprint verification. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No.4, pp. 302-314 (1997).

[5] A. Jain, A. Ross, S. Prabhakar, Fingerprint Matching using Minutiae and Texture Features. In: International Conference on Image Processing (ICIP). Thessaloniki, Greece, pp.282-285 (2001).

[6] A.M. Bazen, G.T.B. Verwaaijen, Correlation based Fingerprint Verification System, Proceedings Program for Research on Integrated Systems and Circuits, Veldhoven, the Netherlands, 29 December-1 November, pp. 205－213 (2000).

[7] D. Maio, D. Maltoni, A.K. Jain, S. Prabhakar, Handbook of Fingerprint Recognition, Springer, New York, 2003.

[8] J. Hollingum, Automated Fngerprint Analysis Offers Fast Verification, Sensor Review, Vol.12, No.3, pp.12-15 (1992).

[9] B.M.Mehtre, N.N.Murthy, A Minutia Based Fingerprint Identification System, Proc. Second International Conference Advances in Pattern Recognition and Digital Techniques, Calcutta 1986.

[10] F. Pernus, S. Kovacic and L. Gyergyek, Minutiae-Based Fingerprint Recognition, Proc. Fifth International Conference Pattern Recognition, pp.1380-1382 (1980).

[11] J.H. Wegstein, An Automated Fingerprint Identification System, US Government Publication, Washington, 1982.

[12] S. Madhvanath, G. Kim, and V. Govindaraju, Chain Code Processing for Handwritten Word Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 21, pp.928-932 (1997).

[13] Z. Shi, V. Govindaraju, Segmentation and Recognition of Connected Handwritten Numeral Strings. Journal of Pattern Recognition, Pergamon Press, Vol. 30, No. 9, pp.1501- 1504 (1997).

[14] D.Simon-Zorita, J.Ortega-Garcia, S.Cruz-Llanas, J.L.Sanchez-Bote, J.Glez-Rodriguez, An Improved Image Enhancement Scheme for Fingerprint Minutiae Extraction in Biometric Identification, Proc. of 3rd International Conference on Audioand Video-Based Biometric Person Authentication AVBPA'01, J.Bigun and F.Smeraldi Eds, LNCS2091, pp. 217-222 (2001).
[15] L.O' Gorman, J.V. Nickerson, An approach to fingerprint filter design, Pattern Recognition 22 (1), pp. 29 – 38 (1989).
[16] B.G. Sherlock, D.M. Monro, K. Millard, Algorithm for enhancing fingerprint images, Electron. Lett. (18), pp.28 (1992).
[17] C.J. Lee, S.D. Wang, A Gabor filter-based approach to fingerprint recognition, in: IEEE Workshop on Signal Processing System, SiPS 99, pp. 371 – 378 (1999).
[18] D. Maio, Direct gray-scale minutiae detection in fingerprints, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1) January, pp.27 – 40 (1997).
[19] G. Agam and Its'hak Dinstein, Generalized Morphological Operators Applied to Map-Analysis, Proceedings of SSPE'96, pp. 60-69 (1996).
[20] P. Viola and M. Jones, Robust Real-time Face Dectection, International Journal of Computer Vision, Vol. 57(2), pp.137-154 (2004).
[21] S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz, Fusion of face and speech data for person identity verification, IEEE Trans. on Neural Networks 10(5), pp. 1065–1074 (1999).
[22] P. Verlinde, G. Chollet, and M. Acheroy, Multi-modal identity verification using expert fusion, Information Fusion 1(1), pp. 17–33 (2000).
[23] B. Gutschoven and P. Verlinde, Multi-modal identity verification using support vector machines (SVM), in Proc. of the International Conference on Information Fusion, FUSION, IEEE Press, pp. 3–8 (2000).
[24] J. Fierrez-Aguilar, J. Ortega-Garcia, D .Garcia-Romero, and J. Gonzalez-Rodriguez, A comparative evaluation of fusion strategies for multimodal biometric verification, in Proc. of IAPR International Conference on Audioand Video-based Person Authentication, AVBPA, Springer, pp. 830–837 (2003).
[25] A. Ross and A. Jain, Information fusion in biometrics, Pattern Recognition Letters 24(13), pp. 2115–2125 (2003).
[26] A. Jain, S. Prabhakar, L. Hong, S. Pankanti, Filterbank-based fingerprint matching, IEEE Trans.Image Process. 5 (9) , pp.846 – 859 (2000).
[27] R. Duda, P. Hart, D. Stork, Pattern Classification, Wiley, New York, 2000.
[28] C.W. Wang, J.W. Li, Fingerprint Identification Using Point Pattern Matching. Journal of Chongqing University (Natural Science Edition), Vol. 25, No.6 (2002).
[29] N.K. Ratha, S. Chen, A. Jain, Adaptive Flow Orientation-Based Feature Extraction in Fingerprint Images. Pattern Recognition, Vol.28, No.11 (1995).
[30] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines. Cambridge University Press. Cambridge, UK, 2000.
[31] Y. Yuan, F. Paolo, P. Massimiliano, Fingerprint Classification with Combinations of Support Vector Machines. Lecture Notes in Computer Science, Vol. 2091. Springer-Verlag, Berlin Heidelberg New York , pp. 253–258 (2001).
[32] V.N. Vapnik, The Nature of Statistical Learning Theory. Springer-Verlag, Berlin Heidelberg New York (2000).
[33] D. Maio, L. Nanni L, An effient fingerprint verification system using integrated gabor filters and Parzen Window Classifier. Neurocomputing 68, pp.208-216 (2005).
[34] T.Joachims, Transductive Inference for Text Classification using Support Vector Machines. In: Proceedings of the 16th International Conference on Machine Learning (ICML). Bled, Slovenia , pp.200-209 (1999).

Jia Jia received the B.E degree in computer science and technology from Tsinghua University, Beijing, China, in 2003. She is now a PHD candidate of the Key Laboratory of Pervasive Computing (Tsinghua University), Ministry of Education. Her research interests include biometrics, pattern recognition and image processing.

Lianhong Cai is now a Professor with the Department of Computer Science and Technology, Tsinghua University. She directs the Human-Computer Speech Interaction Group. Prof.Cai is a member of Multimedia Committee of Chinese Graphics and Image Society, and a member of Chinese Acoustics Society. She has been awarded Scientific Progress Prizes and the Invention Prizes from the Ministry of Mechanism and Electronics, and the Ministry of Education.

Pinyan lu received the B.E degree in computer science and technology from Tsinghua University, Beijing, China, in 2005. He is currently a first year graduate student at Tsinghua University in theoretical computer science. His research interests include algorithms and complexity theory.

Xuhui Liu received the B.E degree in computer science and technology from Tsinghua University, Beijing, China, in 2005. He is currently a first year graduate student at Tsinghua University in Database Laboratory, Institute of Software Department of Computer Science and Technology. His research interests include database and software engineering.