An Efficient Fingerprint Matching Algorithm for Integrated Circuit Cards

Jianwei Yang, Lifeng Liu, and Tianzi Jiang*, Member, IEEE

National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing 100080, P. R. China.

Abstract:

Fingerprint matching is a crucial step in fingerprint identification. Recently, a variety of algorithms for this issue have been developed. Each of them is application situation specific and has its advantages and disadvantages. It is highly desired to develop an efficient fingerprint verification technology for Integrated Circuit (IC) Cards or chips of high degree of security. IC cards have some special characteristics, such as very small storage space and slow processing speed, which hinder the use of most fingerprint matching algorithms in such situations. In order to address this issue, this paper presents an improved minutia-pattern (minutiae-based) matching algorithm by employing the orientation field of the fingerprint as a new feature. Our algorithm not only inherits the advantages of the general minutia pattern matching algorithms, but also overcomes their disadvantages. Experimental results show that the proposed algorithm can greatly improve the performance of fingerprint matching in both accuracy and efficiency, and it is very suitable for applications in IC cards.

Keywords: Fingerprint matching, minutia-pattern matching, ridge-based matching, IC cards, orientation field, FAR

1. Introduction

Fingerprint recognition as a reliable method among biometric feature recognition technology is widely applied in personal identification for the purposes of high degree of security. In most
application cases, some special requirements, for example, the high verification or recognition speed, small size of storage space for features, and enough accuracy have to be met. Consequently, it is impractical to store the total fingerprint as a feature into a template owing to the limitation of storage space and processing speed. As a result, we have to extract the most reliable features in the fingerprint.

What features are recorded determines the performance of fingerprint identification. Although some graph-based or structure-based fingerprint matching algorithms [1-3] have been proposed, those based on minutiae [4-6,10-12], special filters [13] or multiple matchers [14,18] are more valuable in research. Nowadays minutiae-based algorithms are extensively applied in fingerprint identification systems for civilian applications because minutiae, which are restricted to endpoints and bifurcations in this paper (see Figure 1), are believed to be the most reliable fingerprint features.

***Please insert Figure 1 about here***

The use of fingerprint verification for smart cards is a salient example among the various applications of fingerprint identification technology. Smart cards, a kind of IC cards, are being adopted in many application cases, such as mobile phones, credit cards, which require high degree of security. Fingerprint verification technology can meet this security requirement. However, smart cards are much inferior to personal computers or workstations in the capacity of storage space and processing speed because they inherently consist of the DSP (digital signal processor) or single chip microprocessor. Therefore, it is essential to save the storage space for features and speed up the processing on designing a fingerprint verification system that can be applied in smart cards.

For the special characteristics of IC cards, it is necessary to provide an appropriate matching algorithm for corresponding applications. So far, the popular fingerprint matching methods can be categorized into two classes: filter-based and minutiae-based. The former is developed by A. K. Jain et al [13] as an innovative approach to compete with the typical minutiae-based matching
algorithms. They detect a special point in a fingerprint as the reference point (core point is a special one, see Figure 1) and employ Gabor filters to acquire features different from minutiae enclosing the reference point, which is donated as FingerCode in [13]. However, Gabor filters have a high computational cost. And their performance is also much sensitive to their parameters, such as the density of ridges that is usually variant and difficult to be estimated in low quality fingerprints. Moreover, the reference point cannot be always correctly located due to poor quality or irregular texture of the fingerprint, which is fatal to this algorithm. Minutiae-based fingerprint matching is the approach of matching fingerprints through minutia patterns, which are extracted during the phase of minutiae extraction. There have existed some algorithms for point pattern matching without utilizing directions at the points [15]. Minutiae-based matching extends them by taking account of the direction of every minutia.

The fingerprint matching algorithms based on minutiae can also be focused on two prominent classes, whose differences lie in feature information stored in the template. One only stores the minutiae’s characteristics, including their coordinates, directions and types, acquired during the off-line minutiae extraction [4]. Then, the input fingerprint’s minutiae are detected by the same minutiae extraction method on-line and then compared with the minutiae stored in the fingerprint template. Finally, matching results are yielded through corresponding criteria. Comparatively, in addition to minutiae, the other also records associated ridges between minutiae as new information into the template [5]. It is an improved version of the former. Then, minutiae and associated ridges are extracted from the input fingerprint and matched with the combined features stored in the template based on both minutiae and ridges information. Obviously, the latter makes use of more feature information and hence achieve higher accuracy than the former. However, extra feature information may lead to burden the computation and overly increase the storage space. Associated ridges in [5] are represented as one-dimensional discrete signals. There is at least one ridge between two neighboring minutiae located in a same ridge and there are about 20~60 minutiae in a
normal fingerprint. Thus, associated ridge information is large in the total size to be stored for a small storage without compression, which much hinders the application of the ridge-based matching algorithm in those cases. Accordingly, we concentrate on the general minutiae-based algorithms for our IC cards’ application purpose. Unfortunately, the general minutia-pattern matching algorithms have some disadvantages, which corrupt the verification accuracy.

In order to retrieve fingerprints in large fingerprint databases, Ratha et al [4] proposed a general algorithm of minutia-pattern matching by means of Hough transform. To this end, the problem of the minutia pattern matching was converted to detect the highest peak in the Hough space of transformation parameters by discretizing the set of all possible transformation parameters and meanwhile computing the corresponding matching score for each transformation. Finally, they obtained the matching result by examining the number of matched minutiae. However, it is very difficult to accumulate enough evidence in the Hough transform space for a reliable matching when the number of minutiae is very small [5], which often encounter in practice. One of its direct difficulties is to precisely align two minutia patterns. The reason for the malfunction of the general minutia pattern matching can also be described as follows. Although minutiae are reliable features hiding in the fingerprints, they cannot always be correctly detected due to poor quality of fingerprint images or imprecise minutiae extraction algorithms. The number of matched minutiae is less than that of unmatched minutiae when the area of the overlapped region between two fingerprints is too small. So, it is challenging to achieve enough verification accuracy in that situation for this algorithm. Furthermore, their arbitrary scheme, i.e., the matching result is determined by only computing the number of matched minutiae disregarding that of unmatched minutiae, aggravates the situation. As a result, the general minutia-pattern matching algorithms will fail when the number of matched minutiae between two identical fingerprints (i.e. they are from a same finger) is too small.

This paper proposes a hybrid algorithm, which is originated from the minutiae-based and
ridge-based algorithms. The proposed method incorporates the advantages of two methods. Our algorithm, focusing on the fingerprint verification, not recognition, aims to overcome the shortcomings of the general minutia pattern matching algorithms and to improve the verification accuracy in IC cards applications without sacrificing the processing speed or greatly increasing the size of storage space. For this purpose, we employ the orientation field of the fingerprint as a new feature. Our method only increases tens of bytes of extra storage space for a 256 × 256 fingerprint image in contrast to the general minutia pattern matching algorithm described in [4], but it highly improves the performance of fingerprint verification in both efficiency and accuracy which are represented by the great descent of FAR (False Acceptance Rate).

The rest of this paper is organized as follows. In section 2, we will briefly introduce our feature extraction method, especially the step of computation of the orientation field. Section 3 is devoted to the methods of fingerprint matching and describing our more efficient algorithm. Experimental results are reported in section 4 and some conclusions are outlined in the last section.

2. Feature Extraction

Although this paper mainly discusses the fingerprint matching, we firstly give a brief introduction to our feature extraction method because its result greatly influences the performance of fingerprint matching. There are many features hiding in the fingerprints. Among them, minutiae are the most reliable features, invariant with aging and peculiar with others. However, in some cases, minutiae are insufficient to determine the matching situation when the amount of the overlap between two fingerprints is very small. In our algorithm, we employ another reliable feature, the orientation field underlying the fingerprint.

2.1 Computation of Orientation Field of Template Fingerprint

Orientation field is the set of directions of those pixels located in the ridges, which is stable feature information, independent of fingerprint capture equipments and distinct with respect to that of a different finger. Moreover, it can be easily detected and stored. In our algorithm, the
The orientation field of the fingerprint is not only significant to correctly extract minutiae from the fingerprint, but also stored as a feature into the template.

A. R. Rao in [9] has proposed an algorithm to estimate the orientation at every pixel of texture images. In order to compute the orientation field more exactly, A. K. Jain et al extends Rao’s approach to a new hierarchical one [5]. In our algorithm, we utilize the latter to estimate the orientation field.

The steps to estimate the orientation field in [5] are as follows:

1) Divide the input fingerprint image into blocks of size $W \times W$.

2) Compute the gradients $G_x$ and $G_y$ at each pixel in each block.

3) Estimate the local orientation of each block using the following formula:

$$
\theta(i, j) = \frac{1}{2} \tan^{-1}\left(\frac{\sum_{u=i-W/2}^{i+W/2} \sum_{v=j-W/2}^{j+W/2} 2G_x(u,v)G_y(u,v)}{\sum_{u=i-W/2}^{i+W/2} \sum_{v=j-W/2}^{j+W/2} (G_x^2(u,v) - G_y^2(u,v))}\right)
$$

where $W$ is the size of each block, and $G_x$ and $G_y$ are the gradient magnitudes in $x$ and $y$ directions respectively. Then, $\theta(i, j)$ is regularized into the degree range of $-90^\circ \sim +90^\circ$. Finally, we obtain the orientation field image consisting of orientations at every pixel.

To implement the formula (1) more efficiently, we utilize a sliding window technique to speed up the computation. Moreover, to exactly estimate the orientation field, precisely selecting the size of blocks is a key factor. So, A.K. Jain et al also calculated the consistency level of the orientation field in each block as an improved method of Rao’s. The consistency level is acquired through computation of the standard deviation of the orientation field in each block. The improved method can estimate the orientation field more exactly by both lowering the resolution level and re-estimating it when the consistency level is lower than a fixed threshold.

To save the storage space, we sample the orientation field centered in the fingerprint instead of
the whole one since the orientation field in the margin of the different fingerprint is often similar topologically. During the step, we divide the template fingerprint into $W_o \times W_o$ blocks and store the values of \textit{mean} and \textit{standard deviation} of the orientation field in each block centered in the fingerprint as new features into the template off-line (see Figure 2). To save storage space further, our algorithm stores the new features by their natural orders in the fingerprint instead of recording their positions. In our method, we set the size of each block as $32 \times 32$ for our tested fingerprint images, and record the values of \textit{mean} and \textit{standard deviation} of the orientation field in $4 \times 6$ blocks (see Figure 2).

***Please insert Figure 2 about here***

2.2 Minutiae Extraction

Minutiae extraction intends to detect minutiae and record their attributes including coordinates, directions and types, into the template. In our approach, we exclude minutiae’s type due to its imprecision and insignificance in fingerprint matching. D. Maio \textit{et al} [7] have proposed a minutiae extraction algorithm, which detects minutiae directly from gray-level fingerprint images without binarization or thinning. We have made some improvement on their algorithm and achieved better performance [8]. In this paper, we apply our improved method to extract minutiae and record them into the template. Refer to [7, 8] for more details about the phase of minutiae extraction.

After estimating the orientation field and extracting the minutiae, we obtain the combined features. A fingerprint and its corresponding orientation field image, minutia pattern are laid out in the Figure 3.

***Please insert Figure 3 about here***

3. Fingerprint Matching

The purpose of fingerprint matching is to determine whether two fingerprints are from the same finger. For this goal, we have firstly to align the input fingerprint with the template fingerprint that
is represented by its minutia pattern and retained orientation field, before computing the matching score of two fingerprints, because we do not have any prior knowledge about what is the relative position of two fingerprints and whether they are identical.

Referring to the matching phases described in [4, 5], we divide our algorithm into two stages as follows:

Step 1. Registration of feature patterns:
- Coarse registration;
- Computation of orientation field of input fingerprint;
- Fine registration.

Step 2. Computation of matching score.

3.1 Registration of Feature Patterns

In this step, we attempt to acquire the best registration of two fingerprints, assuming they are from the same finger. To this end, our method resorts to the similarity transformation and meanwhile obtains corresponding parameters and reference point. No matter whether the two fingerprints are identical ones or not, the following similarity transformation is performed:

$$F_{s,\Delta x,\Delta y}(x, y) = s\begin{pmatrix} \cos \Delta \theta & -\sin \Delta \theta \\ \sin \Delta \theta & \cos \Delta \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$  \hspace{1cm} (2)

where, \((s,\Delta \theta,\Delta x,\Delta y)\) denotes a set of similarity transformation parameters. In order to implement similarity transformation easily and tackle those problems of the general minutia pattern matching algorithms mentioned previously, we utilize a hierarchical method to align the positions of two minutia patterns through each minutia’s direction, instead of discretizing all the transformation parameters and detecting the highest peak in the Hough space as in [4]. In detail, the registration of feature patterns is decomposed into three phases: coarse registration, computation of orientation field of input fingerprint and fine registration.

3.1.1 Coarse Registration
Coarse registration in our algorithm is similar to the phase of the minutiae pairing in [4]. During the phase, the approximately correct similarity transformation parameters are obtained by considering the minutiae’s characteristics.

Our algorithm beforehand assumes that at least one common minutia can be correctly extracted from both the template fingerprint and the input fingerprint, and most of the input fingerprint’s minutiae can be transformed to the template fingerprint’s by regarding the common minutia as the reference point, if both of the fingerprints are from the same finger. This assumption is usually feasible in practice. In this step, we align the positions of two minutia patterns only considering their characteristics and disregarding the orientation field. Meanwhile, the scale of transformation parameters is set as a constant 1. To avoid detecting the peak in Hough space, we in turns set each minutia of the input fingerprint as the reference point of similarity transformation and record its corresponding transformation parameters, which maximize the number of matched minutiae of each transformation. For each minutia of the input fingerprint image, a set of transformation parameters \( (s, \Delta \theta, \Delta x, \Delta y) \) is acquired. So, we obtain \( m \) sets of transformation parameters, if the number of input fingerprint’s minutiae is \( m \). Finally, from the \( m \) sets of parameters, our method considers \( c \) sets of transformation parameters \( (\Delta \theta, \Delta x, \Delta y) \) and reference minutiae \( (c \) is set as 5 in our experiments), which correspond to the top \( c \) maxima in the number of matched minutiae, as candidates of correct transformation parameters and corresponding reference points.

This coarse registration method does not limit the position of input finger and hence supports the finger rotation scale to \( 2\pi \), which is a great improvement to the traditional algorithms employed in [4].

3.1.2. Computation of Orientation Field of Input Fingerprint

It is the first step in [4] to detect the core or delta point to classify the fingerprints in the database in order to reduce the retrieval space. It is unnecessary to classify the fingerprints for a one-to-one
fingerprint verification system. However, without the phase of classification to fingerprints, there will be more false accepted matchings for this general method. At the expense of additional storage and processing cost, we resort to the orientation field as new feature information to replace the classification of fingerprints. And the orientation field of the input fingerprint is also used to facilitate the fine registration and add a score in the phase of computation of matching score. Computation of the orientation field of the input fingerprint is a little different from that of the template fingerprint because the position of the input fingerprint has been aligned in the coarse registration.

After the coarse registration, \( c \) sets of candidates of best alignment of two fingerprints are obtained. Then, for each set of candidate, we compute the orientation field of the aligned input fingerprint with the similar steps of computation of the orientation field of the template fingerprint, instead of directly computing the orientation field of the original input fingerprint. Thus, it is ensured that the orientation field of each block in the aligned fingerprint corresponds well to that of the template fingerprint. To speed up the computation of mean and standard deviation of the orientation field, our algorithm accumulates the orientations at those pixels whose corresponding values in the ridge skeleton image are 0 (see Figure 3), i.e., our algorithm only computes the mean and standard deviation of the orientation field at pixels located in ridges instead of all the pixels of the original fingerprint image in order to alleviate the computational cost because pixels outside ridges are not useful in defining the dominant local direction. Meanwhile, only the values of mean and standard deviation of the orientation field of blocks located in the overlapped region are computed, which also contributes to the low computational cost.

To specify the similarity of two fingerprints in the orientation field, we define the matching score as follows:

\[
Os(p,q) = \frac{100N_{nb}}{M_{sb}}
\]  

(3)
where, $M_{sb}$ is the number of blocks in the superposition between two fingerprint images, $N_{mb}$ is the number of blocks in which orientation fields of two fingerprints are identical, and their relationship can be denoted by

$$N_{mb} = M_{sb} - NP_{mean} - NP_{deviation}$$

where $NP_{mean}$ is the number of blocks whose difference in the value of mean is larger than a threshold, and $NP_{deviation}$ is the number of blocks whose difference in the value of standard deviation is larger than another threshold.

Through formulas (3) and (4), we can measure the similarity of two fingerprints in the orientation field. If well aligned, two fingerprints from the same finger are much more similar in the orientation field topologically and $Os(p, q)$ gives a larger value. Therefore, $Os(p, q)$ reflects the result of alignment of two fingerprints.

### 3.1.3. Fine Registration

After the coarse registration, the approximate alignment of two fingerprints is obtained. During the fine registration, our algorithm employ the orientation field of the template fingerprint, recorded in the template, and that of the input fingerprint, acquired in the previous subsection, to achieve the more precise registration.

In most applications, the rotate angle $\Delta \theta$ is usually more significant with respect to the performance of registration than the parameters of $\Delta x$ and $\Delta y$. To more precisely aligning the two fingerprints from the result of coarse registration, we regulate the rotate angle $\Delta \theta$ and the scale $s$. Our method firstly defines the $c$ reference minutiae as the polar point to build a polar coordinate system respectively and converts the coordinates of the minutiae in both the template and aligned input minutiae into the polar coordinates. Then, we regulate the scale $s_i$ and rotate angle $\Delta \theta_i$ of each candidate ($i = 1, \ldots, c$) by little steps. Finally, we achieve the correct registration result from the $c$ candidates by comparing $c$ results of fine registration. The correct registration
result indicates that the number of matched minutiae between two fingerprints reaches the maximum and meanwhile the orientation fields of two fingerprints are best matched.

### 3.2 Computation of Matching Score

After coarse and fine registrations of two minutia patterns, the numbers of paired and matched minutiae can be obtained. If two minutiae fall into a same tolerance box after registration, they are defined as paired in this paper. If paired minutiae have equal directions (within some tolerance), they become matched. So, now each minutia in both the template fingerprint and the input fingerprint is grouped as paired, matched, paired but unmatched and unpaired (i.e., unmatched). How to establish the tolerance box is detailed in [4]. Here, our algorithm follows its scheme but we build it along each minutia’s direction instead of doing so along the coordinate axes because the deformation of fingerprints often appears along the directions of ridges.

The matching score in [4], is computed according to the following formula:

\[
Ms(p, q) = \frac{m' \times m'}{n_p \times n_q}
\]  

(5)

where, \( m' \), \( n_p \) and \( n_q \) represent the number of matched minutiae, the number of minutiae extracted from the input fingerprint, and the number of the minutiae recorded in the template, respectively. The matching score in [5] is defined as follows:

\[
Ms(p, q) = \frac{100m'}{\max(n_p, n_q)}
\]

(6)

which is different from (5). However, it is difficult to decide the threshold of matching score through (5) and (6) because both whether the input fingerprint and the template one are from a same finger and what is the size of the superposition between them are unknown. That is, (5) or (6) will mistake the identical fingerprints when the number of their matched minutiae is smaller than that of unmatched minutiae due to the small area in their overlapped region (see the second and third rows of the Figure 6). Therefore, it is insufficient to determine the result of fingerprint
matching only through the formula (5) or (6). To solve this problem, we also examine the percentage of matched minutiae in paired ones by the following formula:

\[
Ps(p, q) = \frac{100M_m}{N_p}
\]

(7)

where \( M_m \) and \( N_p \) are the numbers of matched minutiae and paired minutiae, respectively. When there is a small number of paired minutiae between two identical fingerprints, the number of the paired but unmatched minutiae is very small, often zero, resulting in a big value of \( Ps(p, q) \) (nearly 100). However, when there are some paired minutiae between two different fingerprints due to the occasion matching, there will exist more paired but unmatched minutiae, which is represented by a small value of \( Ps(p, q) \). As a result, we can still treat the fingerprints as possibly matched candidates when \( Ms(p, q) \) is small and \( Ps(p, q) \) is large.

Although the number and percentage of matched minutiae now can be acquired, it is still insufficient to claim the matching result because of disregarding unmatched minutiae but overly considering matched minutiae. However, how to utilize the information of unmatched minutiae is very difficult. To address the issue, we resort to the orientation field in computation of matching score.

Our computation of matching score consists of the formulas (3), (6) and (7). It is:

\[
\text{Re}(p, q) = \begin{cases} 
0 & \text{if} \quad Mn(p, q) < Tm1 \\
1 & \text{if} \quad Tm1 < Mn(p, q) < Tm2 \\
Ps(p, q) > Tp \\
Os(p, q) > To \\
0 & \text{else} \\
1 & \text{elseif} \quad Mn(p, q) > Tm2 
\end{cases}
\]

(8)

Where \( Mn(p, q) \) represents the number of matched minutiae in two fingerprints, \( T_m1, T_m2 \) represent two thresholds of matched minutiae, \( Os(p, q) \) and \( Ps(p, q) \) are obtained from the formula (3), (7), respectively, and \( Tp, To \) are the least percentage of matched fingerprints in matched minutiae and
matched orientation fields of blocks, respectively. The value of $\text{Re}(s(p, q))$ will take 1, if the two fingerprints are from a same finger.

From the formula (8), it can be concluded that minutiae are the prominent features when $\text{Mn}(p, q)$ are large enough ($Tm_2$ is set as 15 in our experiments). And the percentage of matched minutiae in paired ones, the percentage of matched orientation fields (instead of unmatched minutiae information) and the number of matched minutiae are together taken into account to determine the matching result of two fingerprints when $\text{Mn}(p, q)$ is between $Tm_1$ and $Tm_2$ ($Tm_1$ is set as 5 in our experiments).

4. Experimental Results

We test our algorithm in a fingerprint database at Università di Bologna, Italy. Live-scanned fingerprint images are not adopted in our experiment for the following reasons:
1) Fingerprint images captured from different equipments are different in quality.
2) Compared with the fingerprint database of the Università di Bologna, live-scanned fingerprint images usually have better quality and the general minutia-pattern matching algorithms are easy to achieve high verification accuracy (see Figure 4).

***Please insert Figure 4 about here***

Although the fingerprint database at Università di Bologna is not large that consists of $21 \times 8$ fingerprint images (containing 8 images per finger from 21 individuals), their qualities vary in a wide range, and even there are some series in which fingerprints from the same finger are not certain through visual inspection because of the small area of overlapped region between identical fingerprints or their very low quality. For integrality, we do not exclude any fingerprint regardless of their quality or the area of overlapped region. Moreover, during the phase of minutiae extraction, no filter technology is utilized to enhance the images. As a result, both poor quality and no-filter procedure may lower the verification accuracy of our system. However, we can still argue greater performance of our improved minutia-pattern matching algorithm by comparing ours with the
general minutia-pattern matching algorithms.

Each fingerprint is matched with other fingerprints except itself in our experiments. Totally, $168 \times 167 (28056)$ matchings are performed. The experimental results are shown as follows.

Receivers operating characteristic (ROC) curves for the comparison between our algorithm and the general minutiae based matching method are shown in Figure 5.

***Please insert Figure 5 about here***

Table 1 and 2 exhibit the performances of two algorithms in different given thresholds.

***Please insert Table 1 and 2 about here***

The results of our experiments reveal that our algorithm is very efficient in preventing the error of false matching. That is to say, our algorithm greatly reduces the FAR. Except that we reduce the constraint of the orientation field in computing matching score, our algorithm cannot overly increase the FAR (see Figure 5, solid curve suspends when FAR is less than $10^0$), which demonstrates the advantage of our algorithm. When two identical fingerprints are best aligned by the similarity transformation through their minutiae’s characteristics, their orientation fields will match well (see Figure 6, the first row). If only considering the number of matched minutiae and disregarding the orientation field or other feature information as in the literature [4], we maybe claim two identical fingerprints in the second row of the figure 6 are not from the same finger because of the small number or percentage of matched minutiae between the two fingerprints. However, in fact, they are identical. So, with the orientation field information, our approach can add some evidences into the matching score if their orientation fields are much similar topologically. Furthermore, in order to improve the verification accuracy, we can lower the threshold of the least number and lowest percentage of matched fingerprints in the whole minutiae. In other word, we intentionally choose more possibly matched fingerprints as candidates. If no new features are employed, the strategy will bring a number of imposters in above situation, which greatly increases the FAR. Since fingerprints of different classes could have some occasionally
identical minutiae after registration, it is still a difficult task to select a proper threshold of matching score for the general minutia pattern matching algorithms. However, through the orientation field features our algorithm can remove those imposters, because their orientation fields are obviously different.

***Please insert Figure 6 about here***

We investigate our experimental results and conclude that the matching errors mostly result from incorrect minutiae extraction and inexact registration, which, to a large extend, result from the poor quality of our tested database and lack of filtering to them. The third row in the figure 6 displays the result of poor quality fingerprint matching. The (1") image and (2") image are from the same finger and have small area in overlapped region, or even different texture. Whether they are determined as identical fingerprints or not depends on the threshold of matching score defined in the formula (8), though our algorithm can approximately align them. If we excessively lower the threshold, some similar fingerprints in both minutiae distribution and orientation field may unexpectedly be claimed as the identical one, such as the situation of the fourth row in figure 6. Most of the false accepted matchings in our experiments result from overly lowering the threshold of matching score.

5. Conclusions

In this paper, we have proposes an improved minutia-based pattern matching algorithm, which achieves good performance in both efficiency and accuracy neither sacrificing the processing speed nor greatly increasing the size of storage space for features.

The primary advantages of the proposed method by adding the orientation field are its small size in storing extra features and easy implementation. In our system, the value of the mean of the orientation field in each block is recorded by one signed byte (-125~+126), and the standard deviation by another unsigned byte (0~255). Both the value of mean and that of standard deviation are represented by the degree, not radian. Consequently, the size of the additional storage space for
new features of a 256×256 fingerprint is $4 \times 6 \times 2$ bytes. Storage space for features is reported by A. K. Jain et al in [13]. In their experiments, 640 bytes of storage are required for a 508×480 fingerprint image and 896 for an 832×768 fingerprint image. For those images, our algorithm will regulate the blocks to the size of 64×64 due to their large resolution 500 dpi (dots per inch). Consequently, our algorithm will require about 200~300 bytes for the features of minutiae and orientation fields (assuming there are 20~60 minutiae in a fingerprint), which is smaller than that of the method in [13] or [5]. So, our algorithm is very appropriate for applications in IC cards or chips.

Another advantage of our algorithm is its more precise registration scheme, which supports the finger rotation scale to $2\pi$. Through the orientation field and computation of $P_s(p,q)$, we can, to some extent, tackle the trouble that the number of matched minutiae between two identical fingerprints is comparable with that of occasionally matched minutiae between two different fingerprints. Therefore, our method can achieve good performance without classification to fingerprints, which also benefits to applications in IC cards.

**Acknowledgements**

We acknowledge that the experiments in this research were conducted on the fingerprint database at the Università di Bologna of Italy. We would also like to give thanks to our colleagues in Medical Imaging and Computing Group for their stimulated discussions and comments on our work.

**References**


About the author—JIANGWEI YANG received his B.Sc. degree in automation from Tianjin University, P. R. China, in 2000. He is currently a M.Sc. candidate in the National Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences. His research interests include biometrics, image processing, pattern recognition, and computer vision.

About the author—LIFENG LIU received his B.Sc. degree in telecommunications from Chang Chun Institute of Post and Telecommunications, P. R. China, in 2000. He is currently a M.Sc. candidate in the National Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences. His research interests include biometrics, image processing, pattern recognition, and computer vision.

About the author—TIANZI JIANG received his BSc degree in computational mathematics from
Lanzhou University, China, in 1984, and the MSc degree in applied mathematics and PhD degree in computational mathematics from Hangzhou University, China, in 1992 and 1994, respectively. From 1984-1989, he was an assistant professor at Suzhou Silk Engineering Institute. From 1994-1997, he was first a postdoctoral research fellow and then an associate research professor at National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (04/1994-11/1999). I was with School of Mathematics at the University of New South Wales, Sydney, Australia (06/1997-06/1999), Signal and Image Processing Group at Max-Planck Institute of Cognitive Neuroscience, Leipzig, Germany (06/1999-03/2000), and School of Computer Science at the Queen's University of Belfast (04/2000-04/2001). Now he is a full professor and the leader of Medical Imaging and Computing at National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences. His research interests include modeling in functional brain imaging, medical image analysis, computer vision, computer graphics and visualization, and applied mathematics. He is a member of the editorial board of International Journal of Computer Mathematics, and served for some international conferences in medical imaging and analysis as a program committee member. He is also a member of the IEEE, the IEEE Computer Society, and the IEEE Engineering in Medicine and Biology Society.
Figure 1. Fingerprint image and corresponding minutiae. Circles represent bifurcation, rectangles representing endpoints, and triangle representing core point.

Figure 2. Fingerprint images and corresponding sampled orientation field maps. The size of each block is $32 \times 32$, and red solid lines represent the directions of the value of $mean$, black dotted lines representing the directions of the value of $mean$ plus or subtract $standard deviation$. 
Figure 3. A result of feature extraction. From up-left to bottom-right, the original image, corresponding orientation field image, auxiliary image of the ridge skeleton, and minutia pattern.

Figure 4. Two fingerprints captured from two different equipments. The former is captured by the equipment from Veridicom Company, and the latter is captured by the equipment from Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academic of Sciences. Both are better in quality than our tested fingerprints.
Figure 5. ROC curves of our experimental results
Figure 6. The experimental results of fingerprint verification obtained with our algorithm. The first two fingerprints of each row are the template image and input image respectively. The third one consists of the overlapped fingerprints. The fourth one is the orientation field feature of the template image and input image, where black lines represent the orientation field of template image, red lines, input image. The last column is the results of registration of minutiae patterns.
Table 1. Some experimental results of our algorithm with different threshold Tm1 (given Tm2=15, Ps=85, Os=90)

<table>
<thead>
<tr>
<th>Threshold Value</th>
<th>FAR (%)</th>
<th>FRR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tm1=5</td>
<td>0.149</td>
<td>14.88</td>
</tr>
<tr>
<td>Tm1=6</td>
<td>0.0186</td>
<td>18.71</td>
</tr>
<tr>
<td>Tm1=7</td>
<td>0.0112</td>
<td>21.85</td>
</tr>
</tbody>
</table>

Table 2. Some experimental results of the general minutiae based algorithm with different threshold Tm1 (given Tm2=15, Ps=85, Os=90)

<table>
<thead>
<tr>
<th>Threshold Value</th>
<th>FAR (%)</th>
<th>FRR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tm1=5</td>
<td>2.98</td>
<td>12.59</td>
</tr>
<tr>
<td>Tm1=6</td>
<td>0.841</td>
<td>17.09</td>
</tr>
<tr>
<td>Tm1=7</td>
<td>0.1897</td>
<td>20.92</td>
</tr>
</tbody>
</table>