Journal of Signal and Information Processing, **2011, 2, 79-87**
doi:10.4236/jsip.2011.22011 Published Online May 2011 (http://www.SciRP.org/journal/jsip)

Online Fingerprint Verification Algorithm and Distributed System

Ping Zhang[1], Xi Guo[1], Jyotirmay Gadedadikar[2]

[1]Department of Mathematics and Computer Science, Alcorn State University, Lorman, USA; [2]Department of Advanced Technologies, Alcorn State University, Lorman, USA.
Email: {pzhang, jyo}@alcorn.edu, guoxi3000@yahoo.com

Received January 30th, 2011; revised March 5th, 2011, accepted March 7th, 2011.

ABSTRACT

In this paper, a novel online fingerprint verification algorithm and distribution system is proposed. In the beginning, fingerprint acquisition, image preprocessing, and feature extraction are conducted on workstations. Then, the extracted feature is transmitted over the internet. Finally, fingerprint verification is processed on a server through web-based database query. For the fingerprint feature extraction, a template is imposed on the fingerprint image to calculate the type and direction of minutiae. A data structure of the feature set is designed in order to accurately match minutiae features between the testing fingerprint and the references in the database. An elastically structural feature matching algorithm is employed for feature verification. The proposed fingerprint matching algorithm is insensitive to fingerprint image distortion, scale, and rotation. Experimental results demonstrated that the matching algorithm is robust even on poor quality fingerprint images. Clients can remotely use ADO.NET on their workstations to verify the testing fingerprint and manipulate fingerprint feature database on the server through the internet. The proposed system performed well on benchmark fingerprint dataset.

Keywords*: *Online Fingerprint Verification, Fingerprint Feature Extraction, Web Database Query

1. Introduction

Biometrics refers to the automatic identification of a person based on his/her physiological or behavioral characteristics. Identification based on biometrics is preferred over traditional methods with the advantage that biometrics identification techniques obviate the need to remember a PIN/password which may be forgotten, or the need to carry the tokens like passports and driver's licenses which may be forged, stolen, or lost. For example, in the e-commerce application, the biggest concern is security problem. Currently, the common security method for online transaction consists of using one's password or PIN. However, PIN is cumbersome and insecure as people are afraid of passwords being stolen and the transaction process being invaded by adept hackers. With the increasing use of computers as vehicles of information, it is necessary to restrict access to sensitive/personal data**.** The biometric techniques can potentially prevent unauthorized access to or the fraudulent use of ATM, cellular phones, smart cards, desktop PCs, workstations, and computer networks. As a result, online biometric identification is enjoying a renewed interest.

Since manual fingerprint identification is extremely tedious and time consuming, automatic and reliable fingerprint identification systems are in great demand. Many research achievements on fingerprint recognition and identification have been published in literature recently [1-4]. For example, fingerprint recognition and verification techniques include point set matching [5,6], graph matching [7], simulated annealing and genetic algorithms [8], relaxation [9], neural network based method for classification [10], and structural method for fingerprint recognition [11], etc.

To capture the texture information of fingerprint image, Gabor filter with eight orientations is proposed by Jain *et al*. [12]. Patil *et al.* [13] used four orientations of Gabor filters for extracting fingerprint features from gray scale fingerprint image. The image is cropped to the size of 128 × 128 pixels using its core point as the center. Huang and Aviyente [14] and Khan and Javed [15] employed wavelet based analysis in order to provide rich discriminatory texture structure for fingerprint verification. To capture global transformation between two fingerprint images, genetic algorithm is adopted by Tan and Bhannu

[16].

The combination of fingerprint minutiae and texture information performs the better recognition rate than individual methods. In the reference paper [17], the global structure (orientation field) and the local features (minutiae) are combined so that the global orientation field is beneficial to the alignment of the fingerprints which are either incomplete or poor quality.

Ross *et al.* [18] proposed a new technique to estimate the nonlinear distortion in fingerprint pairs based on ridge curve correspondences. The nonlinear distortion, called thin plate spline (TPS) function, is used to estimate the "average" deformation model for a specific finger when several impressions of that finger are available. Experimental results demonstrated that incorporating their proposed model resulted in an improvement in the matching performance.

Although texture-like feature can improve fingerprint verification performance, minutiae pattern is the most important feature. Generally speaking, minutiae extraction from distorted images is not reliable and often gives erroneous results in matching.

In this paper, we will focus on online fingerprint verification over the internet. First, the basic concept of fingerprint verification is reviewed in the introduction section. Then, an online fingerprint verification distributed system is drawn in Section II. Fingerprint acquisition and image preprocessing are discussed in Section III. Both minutiae feature extraction and dynamic matching algorithm are proposed in Section IV while experiments are conducted in Section V. A conclusion ends this paper.

2. Online Fingerprint Verification Distributed System

Feature extraction plays an important role in the fingerprint verification system. Reliable and significant features are extracted from fingerprint image. The local features (ridge ending and ridge bifurcation) and global features (core point) of fingerprint are defined as follows:

1) Ridge Ending: the point where a ridge ends abruptly.

2) Ridge Bifurcation: the point where a ridge forks or diverges into two branch ridges.

3) Core: The maximum curvature point. Some fingerprint has two cores.

4) Minutiae: ridge ending, bifurcation, and core are called as the minutiae points.

The minutiae are shown in **Figure 1**. Advanced features like loops, islands, and delta (triangular portion) on the fingerprint image can be formed by combining all of the above minutiae points.

In the proposed system, live-scan fingerprint devices are used as fingerprint acquisition. A live-scan fingerprint is directly obtained from the finger without the intermediate use of paper. For example, the Thompson-CFS chip-based sensor works through thermal sensor to detect temperature difference across the ridges and valleys. Siemens sensors are based on differential capacitance.

The captured fingerprint image is a grayscale image with 256 levels and different sizes (512 × 512, 256 × 256, or 128 × 128) depending on application requirement. A benchmark fingerprint database can be obtained from National Institute of Standards and Technology (NIST), USA. The NIST Special Fingerprint Database 4 [19] includes 2000 fingerprint image pairs, which was used to test the proposed fingerprint algorithm.

In the proposed system, fingerprint image preprocessing and feature extraction are conducted on computer workstations. In order to effectively manage fingerprint database on a server through the internet, the most advanced ActiveX Data Objects for .NET (ADO.NET) are employed to conduct fingerprint database query. For example, the operations of fingerprint enrollment, insertion and deletion of a record, etc. are carried out on the workstation sides and the changes in the database stored on the server side. The fingerprint verification is processed over the internet in such a way that the workload of server is decreased. Multiple workstations share one database on the server. The fingerprint distributed system is shown in **Figure 2**.

3. Fingerprint Image Preprocessing

Image filtering, image enhancement, and thinning are indispensable to fingerprint image processing. In this section, we will address these issues.

Figure 1. Fingerprint image with minutiae points.

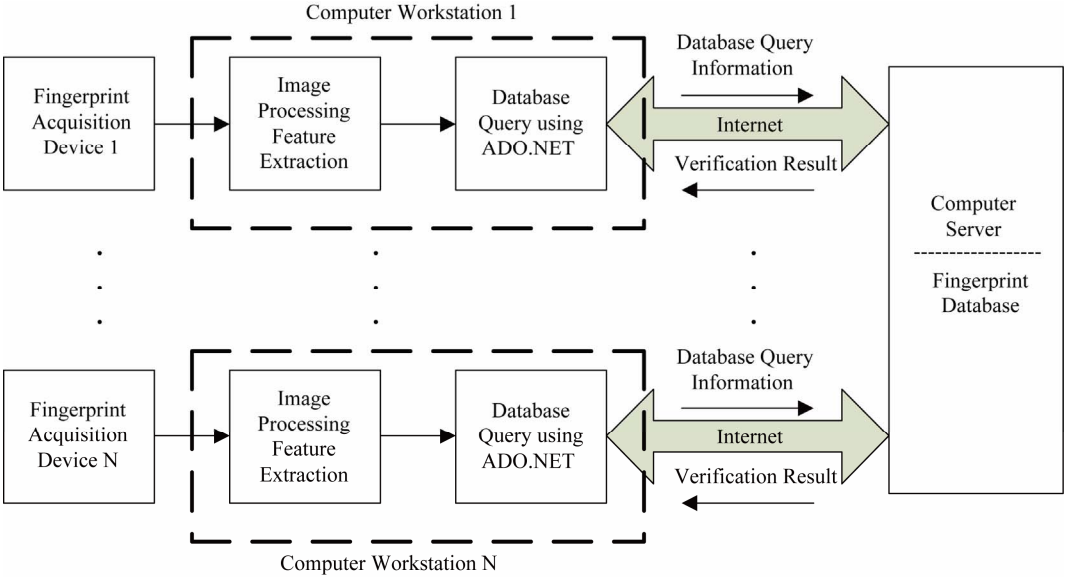


Figure 2. Online fingerprint verification distributed system.

3.1. Image Noise Removal and Enhancement

Median filter [20] is used to remove salt-and-pepper noises or spot-like noises. If a pixel is accidentally changed to an extreme value caused by various reasons, then the result of filter can achieve excellent result. The advantage of median filter is to keep the edge of the image and to remove salt-and-pepper noises in the images.

Histogram Equalization [20] is applied to enhance the ridges of the fingerprint images. Histogram Equalization reassigns a new value of a pixel based on the image histogram. The algorithm works by dividing the image into overlapping subimages with size of $L \times L$; where L is the length and width of the subimage. Histogram information is obtained from local area of the fingerprint image.

3.2. Thinning Fingerprint Image

Fingerprint images consist of ridges. The position and direction of the ridges and the relationship among ridges convey unique information. The extra pixels usually comprise the thickness of the lines that need to be removed in order to accurately extract minutiae points.

According to Zhang and Suen's thinning algorithm [4], a thinning process is much like erosion. The pixels to be removed are marked in the first instance and then removed in a second pass over the image. This process is repeated until there are no more redundant pixels left. The remaining pixels are those belonging to the skeleton of the ridges and no minutiae points have been removed. This is called thinning by successive deletion. The skeleton must remain intact and must have a few basic properties listed below:

1) It should consist of thin regions, one pixel wide;

2) The pixels comprising the skeleton should lie in the centre of the ridges;

3) Skeleton pixel must be connected to each other to form the same number of regions as existed in the original image.

Figure 3 shows the processed result of one original fingerprint image. **Figure 3(a)** is an original low-quality fingerprint image scanned by an optical senor, **Figure 3(b)** shows the result image after median filter, histogram equalization, thinning procedure.

3.3. Spur Detection and Refinement Procedure

After thinning, there exist either some spurs in the thinning image or broken lines in the ridges. A ridge smoothing algorithm is employed to delete the spurs and to link the broken lines with following criteria:

- If a branch in a ridge map is roughly orthogonal to the local ridge direction and its length is less than a specified threshold, then it will be removed.
- If a break line in a ridge is short enough and no other ridges pass through it, then it will be connected.
- If several minutiae form a cluster in a small region, then remove all of them except for the one nearest to the cluster center.
- If two minutiae are located close enough, facing each other, but no ridge lie between them, then all of them are removed.

4. Fingerprint Classification

4.1. Coarse Feature Classification

A fingerprint image can be classified as one of five cate-

(a)

(b)

Figure 3. Original low-quality fingerprint image and the preprocessed images. (a) Original fingerprint image1; (b) Preprocessed image 1.

gories [2]:

- Arch
- Left loop
- Right Loop
- Tented Arch
- Whorl

A human expert can easily perform coarse finger classification. For an automatic system, the problem becomes much more difficult due to poor image quality and impression distortion. Two hierarchical classifications are applied in our recognition system, namely, one coarse classification and one fine classification.

The first step is to employ coarse classification method [2] to classify fingerprint image into one of the five categories.

For the fine classification, further feature extraction method needs to be investigated. Two most stable and easily extracted features are ending points and bifurcation, which are used as fine features in this paper.

4.2. Fine Feature Extraction

After image pre-processing and the coarse feature extraction, a thinned fingerprint image will be applied to following process:

Minutiae extraction: In order to extract minutiae in the thinned fingerprint image, a 5 × 5 pattern template is imposed on the image as shown in **Figure 4**.

In the minutiae extraction, the thinned image is scanned twice to locate two kinds of points (ridge-ending and bifurcation).

Assuming Point M is a detecting point, the set of B(0), B(1), ⋯, B(7) are its 3 × 3 neighboring points in a clockwise direction beginning at top-left position; whereas the set of A(0), A(1), A(2), ⋯, A(15) are its 5 × 5 neighboring points, which directions are shown in **Figure 4**.

If M is a ridge ending, the following criteria must be satisfied:

Criterion 1:

$$C_1 = \sum_{k=0}^{7} \left| \langle B(k+1),8\rangle - \langle B(k),8\rangle \right| = 2$$

and (1)

$$C_2 = \sum_{k=0}^{15} \left| \langle A(k+1),16\rangle - \langle A(k),16\rangle \right| = 2$$

Criterion2:

if $\left(B(j) \neq 0\right)$

then

$$\sum_{k=-1}^{1} \left(\langle A(2j-k),16\rangle \right) = 1, \ \text{if} \ \ j = 1,3,5,7$$

and (2)

$$\sum_{k=-2}^{2} \left(\langle A(2j-k),16\rangle \right) = 1, \ \text{if} \ \ j = 0,2,4,6$$

where $\langle B(j),8\rangle$ presents the modulo operation of $B(j)$ with 8, where 8 is the total number of pixels in the 3 × 3 neighboring ring; whereas $\langle A(j),16\rangle$ denotes the modulo operation of $A(j)$ with 16, where 16 is the total number of the 5 × 5 neighboring ring.

The direction of the ending point M is assigned as one of the directions $A(0) \sim A(15)$, in which the direction of the correspondence $A(i)$ is non-zero. The position and the direction are considered as ending point feature set.

For the bifurcation extraction, criterion 3 is applied.

*A*0	*A*1	*A*2	*A*3	*A*4
*A*15	*B*0	*B*1	*B*2	*A*5
*A*14	*B*7	*M*	*B*3	*A*6
*A*13	*B*6	*B*5	*B*4	*A*7
*A*12	*A*11	*A*10	*A*9	*A*8

Figure 4. A minutiae extraction mask.

Criterion3:

$$C_3 = \sum_{k=0}^{7} \left| \langle B(k+1),8 \rangle - \langle B(k),8 \rangle \right| = 6$$
$$\text{and} \qquad (3)$$
$$C_4 = \sum_{k=0}^{15} \left| \langle A(k+1),16 \rangle - \langle A(k),16 \rangle \right| = 6$$

We can modify the criterion 2 accordingly.

The direction of the bifurcation is assigned as one ridge direction that has the maximum distance to the other two ridges. It can be expressed by Criterion 4 below:

If $\left(A(j)=1\right)$

Then r_1 *and* r_2 are calculated based on the following equation:

$$D_l = r_1 + r_2 \left| \left(\sum_{i=1}^{r_1} \langle A(j+i),16 \rangle = 0 \right) \right. \qquad (4)$$
$$\left. \text{and} \left(\sum_{i=0}^{r_2} \langle A(j-i),16 \rangle = 0 \right) \right.$$

$l \in (1,2,3),\ 0 \le j \le 15,$
$DIR = D_{\max}\ iff.\ D_{\max} = \max\{D_1, D_2, D_3\}$

where r_1 is the number of the binary pixels, which values are continuously equal to 0 on the 5 × 5 neighboring ring counting in the clockwise direction, and starting from the point $A(j)$; whereas r_2 has the same meaning in the anticlockwise direction.

l represents one of three ridges on the bifurcation area. *DIR* is the direction of the bifurcation.

For example, in **Figure 5**, the central point is point *M*, which is a bifurcation point with three ridges: Line *M*-*A*14, Line *M*-*A*4, and Line *M*-*A*8. The number of 0's counted by Equation(4) in the its 5 × 5 neighboring pixels for Line *M*-*A*14 is 10; the number of 0's for Line *M*-*A*4 is 8; and the number of 0's for Line *M*-*A*8 is 8. So the direction of the bifurcation is the direction of Line *M*-*A*14.

If there are two lines that have the same number of 0's in the 5 × 5 neighboring pixels, then first ridge line direction counted from top-left in the clockwise direction will be considered as bifurcation direction.

*A*0	*A*1			*A*4
*A*14		*M*		
*A*12				*A*8

Figure 5. Bifurcation point in the 5×5 neighboring area.

4.3. Fine Feature Match

An efficient and effective algorithm for feature matching is the central theme of an automatic fingerprint identification system. As its name implies, the matching process deals with two fingerprint impressions captured at different times and in different environments. It will verify whether or not the two fingerprints are identical. The matching of two fingerprint impressions has proved to be difficult. Three areas of concerns might be that: 1) the minutiae of fingerprint impressions captured might have different coordinates and relative angles; 2) the shape of two fingerprint impressions taken at different places might not be the same due to the effect of stretching; 3) same fingerprint might have different impression images as the different parts of fingerprint enrolled or as noises introduced while enrolling.

An ideal automatic fingerprint system must satisfy the following criteria: 1) the size of feature file must be small in order to minimize search time and space; 2) matching algorithm must be fast as well as accurate; 3) matching algorithm must cope with the problems of rotation, distortion, false minutiae, and omitting minutiae of matching pairs; 4) matching algorithm must be robust if two matching fingerprints only have partial images.

Based on the criteria mentioned above, a novel hybrid matching approach is proposed in this paper. The algorithm is divided two steps:

- All minutiae are ordered according to the distance from the fingerprint's core.
- The distance is represented by the number of ridges between two feature points in the fingerprint image.

All the minutiae are listed according to the distance to the core: $\{P_0, P_1, P_2, \cdots, P_{m-1}\}$. Here m is the number of minutiae.

Each feature set includes the following information:

Information of each minutia in the fingerprint image must be recorded. The details of encoding information are described as data structures below:

```
Struct Fingerprint Feature
{
unsigned char Type; // type of calculating minutia: ending point (1) or bifurcation (2)
int DIR; // direction of the calculating minutia
struct Neighborhood_Minutiea_Feature P[6];
POINTS Coordinates_xy; // coordinates of testing minutia
}
Struct Neighborhood_Minutiea_Feature
{
unsigned char Type; // type of neighboring
```

minutiae: *ending point* (1) *or bifurcation* (2)
int ***LDIS***;
// distance between the neighboring minutia and the calculating minutia
int ***LDIR***;
// direction difference between the neighboring minutia and the calculating minutia
}

For each minutia P_i, five neighboring minutiae which are nearest to the minutia P_i are sorted. The five sorted minutiae are represented by P_{i1}, P_{i2}, P_{i3}, P_{i4}, P_{i5} respectively.

The distance between the P_i and one of the neighborhoods P_{ij} is defined as:

$LDIS[i,j]$ = the number of ridges between P_i and P_{ij}; $j = 1, 2, \cdots, 5$.

The local direction difference between P_i and P_{ij} is calculated as follows:

$$LDIR[i,j] = DIR[i] - DIR[j]$$

where $DIR[i]$ is the direction of calculating point P_i. $LDIR[i,j]$ is the directional difference between the calculating minutia P_i and its neighboring minutia P_{ij}.$LDIS[i,j]$ and $LDIR[i,j]$ are saved into the data structure: *Neighborhood_Minutiea_Feature*.

Based on our definition, the extracted features (distance and direction) are insensitive to fingerprint image distortion and rotation.

Let the testing fingerprint $\overline{F}$ be represented as a set of *m* minutiae features as follows:

$$\overline{F} = \{F_1, F_2, \cdots, F_m\} \tag{5}$$

Note that each of the elements in the feature set is a feature data structure containing the following components:

$$\begin{aligned} F_i = \{&Type,\ DIR,\ Type.P_{i1},\ P_{i1}.LDIS,\ P_{i1}.LDIR, Type.\\ &P_{i2}, P_{i2}.LDIS,\ P_{i2}.LDIR, Type.P_{i3},\ P_{i3}.LDIR,\\ &Type.P_{i4},\ P_{i4}.LDIS,\ P_{i4}.LDIR,\ Type.P_{i5},\\ &P_{i5}.LDIS,\ P_{i5}.LDIR,\ \text{Coordinates\_xy}\} \end{aligned} \tag{6}$$

Similarly, let the feature set of *kth* fingerprint image in fingerprint database be represented as a set of *n* minutiae points. Each minutiae point is represented by two data structures defined in the Struct Fingerprint Feature and Struct Neighborhood_Minutiea_Feature.

$$\overline{R_k} = \{R_{k1}, R_{k2}, \cdots, R_{kn}\} \tag{7}$$

It is assumed that the two fingerprint images are roughly aligned. The matching algorithm seeks to find the number of matched minutiae between the testing fingerprint and the reference fingerprint. The proposed matching algorithm is elaborated as follows:

The distance between the *ith* minutia of testing fingerprint F_i and the *jth* minutia of the reference fingerprint R_{kj} in the database can be calculated according to the distance integral norm:

$$\begin{aligned} D(i,j) = &abs\left(F_i.Type - R_{kj}.Type\right)\\ &+w_1 * \left(F_i.DIR - R_{kj}.DIR\right)\\ &+\sum_{l=1}^{5}\{w_2 * abs\left(F_i -> P_{il} \cdot LDIS - R_{kj} -> P_{jl} \cdot LDIS\right)\\ &+abs\left(F_i -> P_{il} \cdot LDIR - R_{kj} -> P_{jl} \cdot LDIR\right) \end{aligned} \tag{8}$$

The symbol "–>" represents the data structure pointer operation; parameters w_1 and w_2 are weight factors which are empirically determined to give the maximum similarity matching. In our experiments, w_1 is set to 1.0 and w_2 is set to 0.5.

Input: A set of structural features of the testing fingerprint $\overline{F} = \{F_1, F_2, \cdots, F_m\}$ and a set of structural features in the *kth* reference fingerprint $\overline{R_k} = \{R_{k1}, R_{k2}, \cdots, R_{kn}\}$.

Output: the ratio of the matched pair of two fingerprint images to the average minutiae number of two fingerprint images (*MF*).

Count = 0;
*Loop*1: *FOR* (i = 1; $i \le m$; ++i) {
*Loop*2: *FOR* (j = 1; $j \le$ n; ++j) {
If ((*Type of* F_i ==*Type of* R_{kj}). *and.* $|F_i.DIR\text{-}R_{kj}.DIR| < \phi$)
{
Compute $D[i,j]$;
$D[i]min = min(D(i,j-1), D(i,j))$;
// Write down the value j of reference fingerprint, which has minimum distance with minutiae i in testing fingerprint;
}
} *// end of Loop* 2
If ($D[i]min < \sigma$)
Count = *Count* + 1;
// increment the matched minutiae numbers; *A paired testing minutia will not be paired again;*
} *// end of Loop* 1
MF = 2**Count*/(*m***n*);
End

Here ϕ and σ are two parameters depending on the image resolutions and application conditions. In our experiments, ϕ is set to 0.75 and σ is set to 5.0. *MF* is a parameter to judge the matching degree. The higher the *MF* is set, the higher possibility the two fingerprints will get matched.

4.4. Discussion of Proposed Algorithm

As the distance between two feature points on a finger-

print image is measured by the number of ridge/valley, the distance feature is therefore insensitive to scale and distortion. In addition, direction feature is the difference between the direction of testing minutia and that of its neighboring minutia, which is image-rotation insensitive. As a result, the proposed fingerprint feature extraction method is insensitive to image distortion and rotation.

Another advantage is that an elastically structural feature matching method proves to be a fast and reliable solution for fingerprint verification and identification.

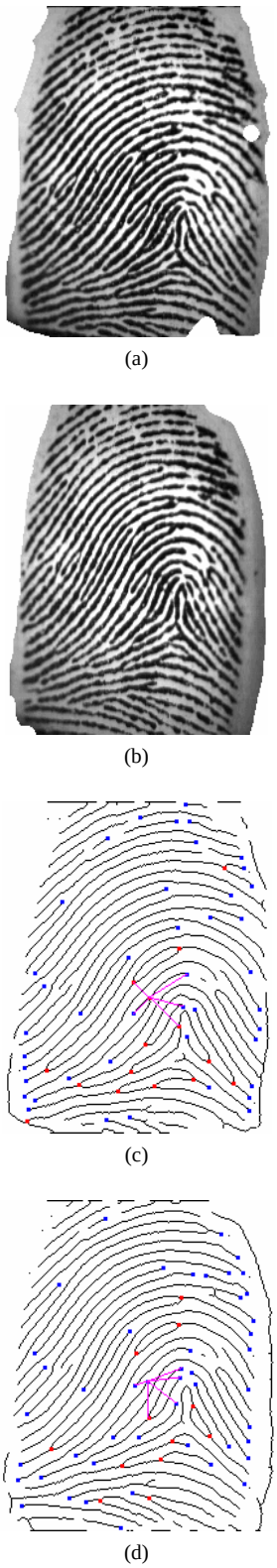
5. Experiments

NIST Special Fingerprint Database IV [19] is used in the experiment. The database contains 2000 fingerprint image pairs with 8-bit grayscale images. Some poor quality fingerprints images are severely deformed due to rolling and/or inking problems of the distorted ridges.

In the first experiment, 200 fingerprints from NIST Special Database 4 Fingerprint Database were chosen to test the proposed system. By using PC computer with 4 MB of internal memory and 2.50 GHz, The system shows that the image preprocessing which includes image filtering, enhancement, and thinning for a 256 × 256 grayscale fingerprint image, needs about 150 ms. The minutia feature extraction and encoding takes 100 ms. It takes about 200 ms to match a testing fingerprint with 200 enrolled fingerprints in the database. As shown in **Figure 2**, image preprocessing and feature extraction were conducted on individual workstation computers. The fingerprint match (verification) is through the internet and conducted on a computer server, where enrolled fingerprint features were saved into database. ActiveX Data Objects for .NET was used for web-based database query. Clients can use ADO.NET on their workstation computer to remotely control the fingerprint feature database on the server, such as fingerprint feature enrollment, deletion, fast sorting, query, etc.

Figure 6 gives two examples of original fingerprint images and image preprocessed (noise removal, image enhancement and thinning) images. In the **Figures 6(c) and 6(d)**, the blue points highlight the ending points and the bifurcation points are marked in red. In each thinned image, one feature point and its five nearest neighboring points are linked for visualizing feature extraction data structure.

As for the deteriorated fingerprint image pairs in the data set, it is very difficult to match all the five neighboring feature points using Equation (8) due to missing minutiae and false minutiae occurring in the processed image. A flexible scheme is proposed in the applications. The different numbers of neighboring feature points are chosen to conduct fingerprint verification in order to achieve the highest recognition rate. **Table 1** shows the relationship of the False Acceptance Rate (FAR), False



(a) (b) (c) (d)

Figure 6. Original fingerprint images and feature extracted images. (a) original fingerprint image1; (b) original fingerprint image 2; (c) feature extracted image 1; (d) feature extracted image 2.

Table 1. FAR and FRR with different number of neighboring feature points in the fingerprint matching.

No. of Neighboring Feature Points	False Acceptance Rate (FAR)	False Reject Rate (FRR)
1	20.00%	0.50%
2	15.50%	1.00%
3	4.50%	2.50%
4	1.50%	5.00%
5	0.5%	6.00%

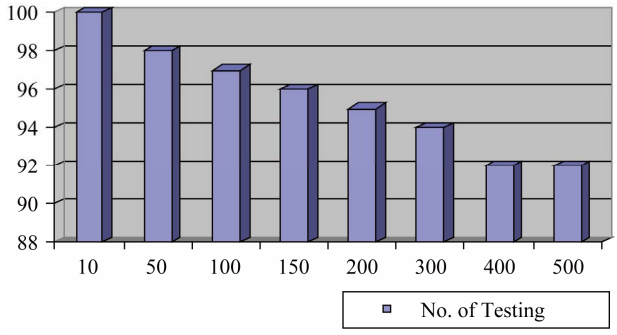


Figure 7. Fingerprint verification rates on different enrollment number of images.

Reject Rate (FRR) with the different number of neighboring feature points in the fingerprint image matching.

In the second experiment, a comparative experiment on the different number of fingerprints was conducted. The recognition performance with different enrollment numbers of fingerprint is shown in **Figure 7**.

For 200 fingerprint images, the verification rate (correct passing rate) can be as high as 95%.

6. Conclusions

Fingerprint verification over the internet is a new research topic as it will deal with web-based database query and fingerprint feature transmission over internet. A web-based fingerprint verification algorithm and its distributed system is presented in this paper. Based on internet transmission, no extra hardware system is needed. A new fingerprint feature extraction method and a flexible structural feature matching algorithm are proposed in the system. Experiments show that the proposed fingerprint verification algorithm is insensitive to fingerprint image distortion, scale, and rotation. The web-based online distributed system performs well on the benchmark fingerprint dataset and practical applications.

Further research will be conducted on how the network security can be applied to fingerprint feature transmission over the internet. The feature data should be encrypted before transmission in order to protect the system from being invaded by adept hackers.

REFERENCES

[1] Z. Miklos and K. Vajna, "A Fingerprint Verification System Based on Triangular Matching and Dynamic Time Warping," *IEEE Transaction on Pattern Recognition and Machine Intelligence*, Vol. 22, No. 11, November 2000, pp. 1266-1276. doi:10.1109/34.888711

[2] N. K. Ratha, K. Karu, S. Chen and A. K. Jain, "A Real-Time Matching System for Large Fingerprint Database," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 8, August 1996, pp. 799-813. doi:10.1109/34.531800

[3] A. K. Jain, L. Hong and R. Bolle, "On-line Fingerprint Verification," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 4, April 1997, pp. 302-314. doi:10.1109/34.587996

[4] T. Y. Zhang and C. Y. Suen, "A Fast Parallel Algorithm for Thinning Digital Patterns," *Communication of the ACM*, Vol. 27, No. 3, March 1984, pp. 236-239. doi:10.1145/357994.358023

[5] J. H. Wegstein, "An Automated Fingerprint Identification System," Technical Report 500-89, National Bureau of Standards, Bethesda, 1982.

[6] S. Umeyama, "Parametterized Point Pattern Matching and its Application to Recognition of Object Families," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 2, February 1993, pp. 136-144. doi:10.1109/34.192485

[7] D. K. Isenor and S. G. Zaky, "Fingerprint Identification Using Graph Matching," *Pattern Recognition*, Vol. 19, No. 2, 1986, pp.113-122.

[8] N. Ansari, M. H. Chen and E. S. H. Hou, "A Genetic Algorithm for Point Pattern Mattering, Dynamic, Genetic, and Chaotic Programming," John Wiley and Sons, New York, 1992, pp. 353-371.

[9] J. Ton and A. K. Jain, "Registering Landsat Images by Point Matching," *IEEE Transaction on Geoscience and Remote Sensing*, Vol. 27, No. 5, September 1989, pp. 642-651. doi:10.1109/TGRS.1989.35948

[10] V. S. Srinivasan and N. N. Murthy, "Detection of Singular Points in Fingerprint Images," *Pattern Recognition*, Vol. 25, No. 2, February 1992, pp. 139-153. doi:10.1016/0031-3203(92)90096-2

[11] A. K. Hrechak and J. A. Mchugh, "Automated Fingerprint Recognition Uing Structural Matching," *Pattern Recognition*, Vol. 23, No. 8, 1990, pp. 893-904. doi:10.1016/0031-3203(90)90134-7

[12] A. K. Jain, S. Prabhakar and S. Pankanti, "Filterbank-based Fingerprint Mtching," *IEEE Transactions on Image Processing*, Vol. 9, No. 5, 2000, pp. 846-859. doi:10.1109/83.841531

[13] P. M. Patil, R. S. Suralkar and H. K. Abhyankar, "Fingerprint Verification Based on Fixed Length Square Finer Code," *Proceeding of IEEE Conference on Tools with*

Artificial Intelligence, Hong Kong, 16 November 2005, pp. 657-662.

[14] K. Huang and S. Aviyente, "Choosing Best Basis in Wavelet Packets for Fingerprint Matching," *Proceedings of IEEE Conference on Image Processing*, East Lansing, 24-27 October 2004, pp. 1249-1252.

[15] N. Y. Khan and M. Y. Javed, "Efficient Fingerprint Matching Technique Using Wavelet Based Features," *Proceedings of Digital Image Computing Techniques and Applications*, 2007, pp. 253-259.

[16] X. Tan and B. Bhanu, "Fingerprint Verification Using Genetic Algorithm," *Proceedings of 6th IEEE Workshop on Applications of Computer Vision*, 2002, pp. 79-83.

[17] J. Gu, J. Zhou and X. Tang, "Fingerprint Recognition by Combining Global Structure and Local Cues," *IEEE Transactions on Image Processing*, Vol. 15, No. 7, 2006, pp. 1951-1964.

[18] A. Ross, S. C. Dass and A. K. Jain, "Fingerprint Warping Using Ridge Curve Correspondences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 1, 2006, pp. 19-30. doi:10.1109/TPAMI.2006.11

[19] C. J. Watson and C. L. Wilson, "NIST Special Database 4 Fingerprint Database," National Institute of Standards and Technology Advanced Systems Division Image Recognition Group, 17 March 1992.

[20] R. C. Gonzalez and R. E. Woods, "Digital Image Processing," 3rd Version, Pearson Prentice Hall, 2008.

