A randomized approach with geometric constraints to fingerprint verification

Kuo Chin Fan\textsuperscript{a, *}, Cheng Wen Liu\textsuperscript{a}, Yuan Kai Wang\textsuperscript{b}

\textsuperscript{a}Institute of Computer Science and Information Engineering, National Central University, Chung-Li 32054, Taiwan, ROC
\textsuperscript{b}Department of Electronic Engineering, Fu-Jen Catholic University, Hsin-Chuang, Taipei 24205, Taiwan, ROC

Received 26 August 1998; accepted 9 August 1999

Abstract

In this paper, a fuzzy bipartite weighted graph model is proposed to solve fingerprint verification problem. A fingerprint image is preprocessed first to form clusters of feature points, which are called feature point clusters. Twenty-four attributes are extracted for each feature point cluster. The attributes are characterized by fuzzy values. Attributes of an input image to be verified are considered as the set of left nodes in a fuzzy bipartite weighted graph, and the attributes of claimed template fingerprint image are considered as the set of right nodes in the graph. The fingerprint verification problem is thus converted into a fuzzy bipartite weighted graph matching problem. A matching algorithm is proposed for the fuzzy bipartite weighted graph model to find an optimal matching with a goodness score. Experimental results reveal the feasibility of the proposed approach in fingerprint verification. © 2000 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Fingerprint verification; Bipartite weighted graph; Fuzzy set; Clustering; Greedy algorithm

1. Introduction

Fingerprint is a distinguishing physiological characteristic for characterizing people. Hence, it plays a significant role in biometrics. In the past decades, the research of automatic fingerprint verification techniques was rising due to the fact that fingerprint verification is a non-intrusive, user-friendly, and relatively inexpensive biometrics technique. Many approaches have been proposed for solving fingerprint verification problem. Grammar matching has been used for fingerprint in earlier works [1]. A fingerprint image is first preprocessed and converted into 1-dimensional symbol string or 2-dimensional tree. Symbol strings/trees are then parsed by string or tree grammars to determine the fingerprint type. Its advantage is that the method focuses on using roughly estimated textural direction of interested image area which is capable of tolerating interference of small noise. The methodology is suitable for coarse classification but is not adequate for detail matching. The drawback lies in its incapability of processing variations of details.

Feature point matching method is a prevalent approach for fingerprint verification [2]. In this approach, a fingerprint image is usually processed by a thinning algorithm to obtain feature data. Feature data of an input image forms a point pattern. The feature data contains ridge endings and ridge bifurcations. The methodology can process detail matching, but is not suitable for coarse classification. Detail matching using this methodology is greatly influenced by the interference of noise and distortion in image processing.

Graph matching method has been proposed by Isenor and Zaky [3] for matching fingerprint images by representing a fingerprint image as an attributed graph. To use relational description of structure, complex algorithms, such as graph isomorphism, have to be adopted in the
2. Fingerprint image processing

2.1. Preprocessing and clustering

In this study, minutiae, i.e. the endpoints and junctions on the ridges, are extracted to serve as feature points. To extract minutiae, ridge ends and bifurcations within a skeletal image have to be identified first. A skeletal fingerprint image is the thinning results obtained from a bi-level fingerprint image. These minutiae are then treated as feature points and fed to a clustering module to form clusters. The adopted clustering technique is the maximin algorithm [4] which is applied to extract minutiae so that close minutiae will be grouped together to form a cluster. The generated cluster is named feature point cluster. After the clustering process, feature points (minutiae) within a fingerprint image are divided into several clusters with each cluster being represented by a rectangular bounding box, which is uniquely determined by the minimum bounding rectangular box enclosing all of the feature points in the cluster. Shown in Fig. 1 is the clustering process of feature points in a fingerprint image.

2.2. Feature extraction

A feature point cluster $c_i$ contains a set of close minutiae, which will be measured and represented by 24 attributes. The attributes are described as follows:

- $NOB(c_i)$ is the number of bifurcations in $c_i$.
- $NOE(c_i)$ is the number of endpoints in $c_i$.
- $WH(c_i)$ is the ratio of width over height of the rectangular bounding box of $c_i$. That is, $WH(c_i) = \frac{\text{Width}(c_i)}{\text{Height}(c_i)}$.
- $OT(c_i)$ is the orientation angle of all ridge pixels in $c_i$. It is estimated by applying a proposed orientation estimation algorithm. The angle ranges between $-180^\circ$ and $180^\circ$. A proposed orientation algorithm was developed. The steps of this algorithm are listed in Table 1.

- $S_k(c_i)$ are twenty measurements obtained by applying $5 \times 5$ masks $\Psi_k$ on $c_i$, where $k \in \{1, \ldots, 20\}$. Shown in Fig. 2 are these 20 masks. The 20 masks $\Psi_k$ are divided into four groups: 45° slant-upward masks (RSL), 135° slant-upward masks (LSL), vertically slant masks (VSL), and horizontally slant masks (HSL). There are three possible values for each pixel of $\Psi_k$: white pixel representing background is denoted as 0, black pixel representing foreground is denoted as 1, and grey pixel representing a do not-care pixel is denoted as $\times$. $S_k(c_i)$ can be obtained by Eq. (1):

$$S_k(c_i) = \left[ \sum_x \sum_y \left( 1 - (I'(x, y) \oplus \Psi_k)/f_k(\Psi_k) \right) \right] f_k(1)'$$ (1)

Fig. 1. The clustering process of feature points in a fingerprint image.
2.3. Fuzzy representation of features

All attributes are converted into the representation of fuzzy sets. Each attribute has several fuzzy sets to fuzzify the attribute value. For instance, the four attributes NOB, NOE, WH, and OT have three fuzzy sets: SMALL, MEDIUM, and LARGE, or have five fuzzy sets: SMALL, SLIGHTLY SMALL, MEDIUM, SLIGHTLY LARGE, and LARGE to represent the fuzzy value of these attributes. Each fuzzy set has a membership function for describing the degree of belonging. Take NOE as an example. If it is measured 6 and is converted into its fuzzy membership values of SMALL, MEDIUM, and LARGE whose membership function is shown in Fig. 3. The fuzzification of the NOE gives three membership values:

\[ \mu_{\text{small}}(6) = 0.14, \mu_{\text{medium}}(6) = 0.67, \mu_{\text{large}}(6) = 0.00. \]

To those people who need to deal with the imprecision and uncertainty, fuzzy set theory is attractive. Abundant of papers relating to fuzzy set theory and its applications have been presented. Many researchers have applied fuzzy set theory to solve complex real-world problems and reasoning [5-7]. In many applications, it is satisfied that membership functions of fuzzy sets are determined based on subjective perceptions rather than on other objective entities or involved data. Generally speaking, construction of membership functions is vital in the applications of fuzzy set theory. Since the determination of membership functions is fundamental in many real-world applications, it is important to notice where membership functions come from and how they are derived. Focuses on the way in obtaining more exact membership functions would be better for a fuzzy system. However, performance using the subjective membership function is satisfied in our experiments. The types of membership functions used in Refs. [8,9] is triangular and sinusoidal membership functions, which can be tuned globally.

3. Fuzzy bipartite weighted graph model

3.1. Bipartite weighted graph

Formally, a bipartite weighted graph \( B = (N, E) \) is a graph, where \( N = N_L \cup N_R \) is a node set containing two mutually independent sets \( N_L \) and \( N_R \), and \( E \) is an edge set connecting \( N_L \) and \( N_R \). A matching of a bipartite weighted graph \( B \) means the production of an edge set \( E \) such that each edge \( e = (l,r), l \in N_L, r \in N_R \) connecting two nodes. Every edge has a positive weight value \( w_{ij} \) to represent the cost of the edge. Each node in \( N_L \) can connect only to one node in \( N_R \), and no two edges connect to the same node. A bipartite weighted graph matching problem is to find an optimal matching with the lowest sum of weight. If there are nodes that are not
Table 2
Modified XOR operation

<table>
<thead>
<tr>
<th>(\psi_k(i,j))</th>
<th>0</th>
<th>1</th>
<th>(X)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Gamma(x + i - 2, y + j - 2))</td>
<td>0</td>
<td>1</td>
<td>X</td>
</tr>
</tbody>
</table>

Fig. 2. The 20 masks \(\psi_k\).

Fig. 3. Membership functions for The NOE feature.

connected in a matching, the matching will be punished by penalty costs in regard to the unmatched nodes. Let \(N_L = \{l_1, l_2, \ldots, l_m\}\), \(N_R = \{r_1, r_2, \ldots, r_n\}\), and \(w_{ij}\) be the weight of the edge \((l_i, r_j)\). Assume \(s_i\) is the penalty of unmatched node \(l_i\), and \(t_j\) is the penalty of unmatched node \(r_j\). The problem of bipartite weighted graph matching with penalty [10] is given by finding a matching matrix \(X = [x_{ij}]\) to minimize Eq. (3):

\[
\sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij}x_{ij} + \sum_{i=1}^{m} \left(1 - \sum_{j=1}^{n} x_{ij}\right)s_i + \sum_{j=1}^{n} \left(1 - \sum_{i=1}^{m} x_{ij}\right)t_j
\]

subject to \(\sum_{i=1}^{m} x_{ij} = 0 \text{ or } 1 \text{ for all } 1 \leq j \leq n\),

\(\sum_{j=1}^{n} x_{ij} = 0 \text{ or } 1 \text{ for all } 1 \leq i \leq m\),

\(x_{ij} \in \{0,1\}\).

Each element \(x_{ij}\) in the matching matrix \(X\) represents an edge in a bipartite weighted graph. That is, \(x_{ij}\) equals 1 if the node \(l_i\) connects to the node \(r_j\). In the following discussion, the nodes within the set \(N_L\) are called left nodes, and the ones within the set \(N_R\) are right nodes. If \(l_i\) is an unmatched node, then \(\sum_{j=1}^{n} x_{ij} = 0\). Likewise, if \(r_j\) is an unmatched node, then \(\sum_{i=1}^{m} x_{ij} = 0\).

3.2. Fuzzy bipartite weighted graph

This subsection gives definitions of our proposed fuzzy bipartite weighted graph model. A fuzzy bipartite weighted graph is a bipartite weighted graph [10] that uses fuzzy sets to characterize the attributed values of nodes. Our model also makes an extension to bipartite weighted graph. Our graph model allows multiple attributes for a node. The following is a fundamental definition of fuzzy bipartite weighted graph.
Definition 1 (Fuzzy bipartite weighted graph). A fuzzy bipartite weighted graph, denoted as $\bar{G} = (N, E)$, consists of a finite set of nodes and a finite set of edges, where

- $N = N_L \cup N_R$ is a finite set of fuzzy-attributed nodes,
- $N_L = \{\bar{t}_1, \bar{t}_2, \ldots, \bar{t}_m\}$ is the set of left fuzzy attributed nodes,
- $N_R = \{\bar{r}_1, \bar{r}_2, \ldots, \bar{r}_n\}$ is the set of right fuzzy attributed nodes,
- $\bar{t}_i \in N$ is a fuzzy attributed node that has a label $v_i$ and $p$ attributes $a_{1}, a_{2}, \ldots, a_{p}$. Each attribute $a_i$ is characterized by $q$ fuzzy sets, where $\mu_{i1}, \mu_{i2}, \ldots, \mu_{iq}$ are fuzzy membership functions of the attribute $a_i$,
- $E = N_L \times N_R = \{x_{ij}|x_{ij} = (\bar{t}_i, \bar{r}_j), \bar{t}_i \in N_L, \bar{r}_j \in N_R\}$ is a finite set of edges.

There are many matchings or edge sets $E$ in a fuzzy bipartite weighted graph. In finding an optimal matching $E$, the cost $w_{ij}$ of each edge $x_{ij}$ is obtained by computing the compatibility between $\bar{t}_i$ and $\bar{r}_j$.

Definition 2 (Compatibility). A compatibility $\alpha$ is a measure number of similarity between two fuzzy attributed nodes $\bar{t}$ and $\bar{r}$. Its value is between 0 and 1. That is, $0 \leq \alpha(\bar{t}, \bar{r}) \leq 1$. Let $\mu_{ij}(\bar{t})$ be the degree of membership of the $j$ fuzzy set for the attribute $i$ of the node $\bar{t}$. The compatibility $\alpha$ is given by

$$
\alpha(\bar{t}, \bar{r}) = \bigwedge_{i=1}^{p} \bigvee_{j=1}^{q} \left( \mu_{ij}(\bar{t}) \land \mu_{ij}(\bar{r}) \right),
$$

where $\land$ and $\lor$ are t-norm and t-conorm operators, respectively.

It is easy to see that the cost $w_{ij}$ of an edge $x_{ij}$ is equal to $\alpha(\bar{t}_i, \bar{r}_j)$. A matching $E$ consists of a number of pairing of fuzzy-attributed nodes between two node sets. Suppose that there is a matching $E = \{x_{ij}\}$, we will define feasibility of a matching that is derived from the compatibility of all pairings of fuzzy attributed nodes in the matching.

Definition 3 (Feasibility). A feasibility $f_E$ is the cost of a matching $E$. It is obtained by calculating the t-norm of all compatibility in the matching, which can be formulated as follows:

$$
f_E = \bigwedge_{(\bar{t}_i, \bar{r}_j) \in E} \alpha(\bar{t}_i, \bar{r}_j).
$$

For each fuzzy bipartite weighted graph, there are many matchings. Each matching has a feasibility. The fuzzy bipartite weighted graph matching problem is an optimization problem that is to find a matching with the optimal feasibility. To punish the unmatched nodes, the fuzzy bipartite weighted graph with penalty problem is elucidated in the next definition.

Definition 4 (Fuzzy bipartite weighted graph matching with penalty problem). A fuzzy bipartite weighted graph matching problem is to find a matching that minimizes the equation:

$$
\sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij}x_{ij} + \sum_{i=1}^{m} \left(1 - \sum_{j=1}^{n} x_{ij}\right)s_{i} + \sum_{j=1}^{n} \left(1 - \sum_{i=1}^{m} x_{ij}\right)t_{j}
$$

subject to $\sum_{i=1}^{m} x_{ij} = 0$ or 1 for all $1 \leq j \leq n$,

$$
\sum_{j=1}^{n} x_{ij} = 0$ or 1 for all $1 \leq i \leq m$,

$x_{ij} \in \{0, 1\}$,

$w_{ij} = \alpha(\bar{t}_i, \bar{r}_j)$.

3.3. Approximate optimization algorithm

To find the optimum of a fuzzy bipartite weighted graph, two algorithms are proposed in this paper. One is an augmentation algorithm that augments $W = [w_{ij}]$ in Eq. (3) to be a square matrix $W^* = [w_{ij}]$. If there are $n$ nodes in $N_i$ and $m$ nodes in $N_j$, then $W$ is an $n \times m$ matrix, and $W^*$ is an $(m + n) \times (m + n)$ matrix. The augmentation algorithm is described in Table 3.

To construct $W^*$ from $W$, the guideline for setting $s_i$ and $t_j$ is given. Considering the condition of $w_{ij} > s_i + t_j$, i.e. the two possible matching nodes are not likely to be selected for an optimal pairing, the setting of $(s_i, t_j)$ is a prerequisite to ensure possible pairings fall within the status to be selected. However, the arrangement of selection for an optimal pairing is not for guaranteed. Thereby, the punishment of a bad pairing between two corresponding pairs $l_i$ and $r_j$ is defined as follows. $w_{ij} = w(l_i, r_j) = |1 - \alpha(l_i, r_j)| \times K_s$, where $\alpha(l_i, r_j)$ is the compatibility defined in Definition 1, and

<table>
<thead>
<tr>
<th>Table 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Augmentation algorithm</strong></td>
</tr>
</tbody>
</table>

**Input**: A weighted matrix $W$, and two penalty vectors $S$ and $T$.  
**Output**: An augmented matrix $W^*$

**Step 1**: Construct a matrix $S'$ from $S$ by setting the diagonal elements $s'_{ij} = s_i$, and all other elements to $\infty$. Create a matrix $T'$ from $T$ in the same way.

**Step 2**: Construct $W^*$ from $W, S', T'$ by

$$
W^* = (w_{ij}) = \frac{W \cdot S'}{T' - \infty}
$$

where $\infty$ is a matrix with each entry being an infinite value.
\(K_\alpha\) is a scaling factor. The value of \(s_i\) and \(t_j\) are empirically set by trial and error. The block size of \(l_i\) and \(r_j\) are considered for the setting of \(s_i\) and \(t_j\), respectively.

The approximate optimization algorithm for solving the fuzzy bipartite weighted graph matching with penalty problem is a greedy algorithm. The greedy algorithm is an iterative method that accepts an augmented matrix as its input, and adopts Hungarian algorithm to find an optimal solution for current iteration.

Since geometric constraints generally represent important relational descriptions of surface structure, a geometric constraint function is also applied in the finding of solution in the greedy algorithm. Suppose \(g(r_i, r_i, l_j, l_j)\) is a testing function to check the following two conditions: (1) whether two node pairs \((r_i, r_i)\) and \((l_j, l_j)\) are near within the distance of \(\Gamma\) pixels, and (2) whether the difference between \(\theta_1 = |OT(l_j) - OT(l_j)|\) and \(\theta_2 = |OT(r_i) - OT(r_i)|\) is within a preset threshold \(\delta\). The geometric constraint function can be formulated as follows:

\[
g(r_i, r_i, l_j, l_j) = \begin{cases} 
0, & \text{if } \min(|\theta_1 - \theta_2|, 180 - |\theta_1 - \theta_2|) \leq \delta \text{ and } \\
1, & \text{otherwise.}
\end{cases}
\]

where \(\theta_1\) is the difference of the orientation angle between nodes \(r_i\), and \(r_i\), and \(\theta_2\) is the difference of the orientation angle between nodes \(l_j\) and \(l_j\).

The greedy algorithm is described in Table 4. The solution of the proposed greedy algorithm can converge to a local optimum satisfying Eq. (6) and the geometric constraint in Eq. (7). The time complexity of the greedy algorithm is analyzed as follows. Since the time complexity of the Hungarian method is \(O(n^3)\), the step 2 in the proposed greedy algorithm can be solved in \(O((n + m)^3)\) time.

### 4. Fingerprint verification as a fuzzy bipartite weighted graph matching problem

In our work, the fingerprint verification task is modeled as a fuzzy bipartite weighted graph matching problem. The clusters in the testing input can be considered as the left nodes of a fuzzy bipartite weighted graph, whereas the clusters of the database template as the right nodes of the fuzzy bipartite weighted graph. The optimal solution to be found reflects the minimum of the total cost between the corresponding nodes in a fuzzy bipartite weighted graph.

Here we will give an example to demonstrate how to model fingerprint verification problem as fuzzy bipartite weighted graph matching problem, and obtain the feasibility between two fingerprint images \(I_1\) and \(I_2\). If we adopt only two attributes \(OT\) and \(WH\) in this example, a node \(I_1 \in N_R\), which represents the first feature point cluster of \(I_1\), has attributes: \(\hat{I}_1 = \{\text{Horizontal, Slant 135}\}\). The first feature point cluster of \(I_1\) is the node \(\hat{I}_1 \in N_R\) that has attributes \(\{\text{Vertical, Short}\}\). The second feature point cluster of \(I_2\) is the node \(\hat{I}_2 \in N_R\) that has attributes \(\{\text{Vertical, Short}\}\). The fourth feature point cluster of \(I_4\) is the node \(\hat{I}_4 \in N_R\) that has attributes \(\{\text{Vertical, Medium}\}\).

### Table 4

The proposed randomized greedy algorithm

<table>
<thead>
<tr>
<th>Input: An augmented matrix (W).</th>
<th>Output: A matching matrix (X = [x_{ij}]).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1: Set (k = 0) and (W_0 = W).</td>
<td>Step 2: Set (k = k + 1) and (W_k = W_{k-1}). Apply Hungarian algorithm to (W_k). Let (X_k) be the obtained optimal solution.</td>
</tr>
<tr>
<td>Step 3: Randomly generate a number (z \in [0, 1]).</td>
<td>Step 4: If (z &gt; \eta), randomly select two distinct elements (x_{ij}) and (x_{kl}) in (W_k) such that: (a) (x_{ij} = x_{kl} = 1) and (b) (w_{ij} \neq 0, w_{kl} \neq 0). Set (w_{ij}) and (w_{kl}) by the following equation:</td>
</tr>
</tbody>
</table>
| \[ w_{ij} = \begin{cases} 
0, & \text{if } g(l_i, l_i, r_j, r_j) = 0, \\
\{w_{ij}\}, & \text{otherwise.}
\end{cases} \] | \[ w_{ij} = \begin{cases} 
0, & \text{if } g(l_i, l_i, r_j, r_j) = 0, \\
\{w_{ij}\}, & \text{otherwise.}
\end{cases} \] |
| Step 5: If \(z \leq \eta\), select the element \(x_{ij}\) that has the maximum compatibility. Set \(w_{ij} = 0\). | Step 6: If \(X_k = X_{k-1}\) or \(k \geq \) maximum iteration number, then stop the algorithm; Otherwise, goto step 2. |
for compatibility between \([5,0.5] \), \([0.25,0.25] \), and \([0.23,0.23] \). The compatibility between \(T_1\) and \(T_2\), \(w_{ij} = a[T_1(T_2)]\) is \([5,0.5] \), \([0.25,0.25] \), and \([0.23,0.23] \).

If \(a\) is the standard AND operator in fuzzy logic and \(\vee\) is the standard OR operator, then the obtained numerical value of \(w_{ij}, w_{12}, w_{21}, \) and \(w_{22}\) are 0.35, 0.41, 0.29, 0.42, respectively. The feasibility \(f_k = \vee (I, (T_i, T_j))\) = \((\vee (w_{11}, w_{22})) or (\vee (w_{21}, w_{12}))\). This is due to the fact that in a bipartite weighted graph, if a matching \(w_{11}\), \(w_{22}\) is optimal, then the other matching, \(w_{12}, w_{21}\), is not, and vice versa.

5. Noisy fingerprint image generation

Fingerprint images are always contaminated by noise. Some portion of the fingerprint image may be lost due to unevenly distributed pressure. One or more cut may appear throughout the fingerprint, or noise may be distributed throughout the image. To simulate these situations, four methods are used for generating noisy fingerprint images.

5.1. Random distribution of constant noise

Perturbation is made randomly at \(p\%\) of \(N_x \times N_y\) pixels for each image with a predefined percentage. Suppose a pixel \(p\) with the gray value \(G_p\) is randomly chosen to be perturbed. Let \(N_g\) be the maximum gray value of pixel. Thereby,

\[
G_p = \begin{cases} 
G_p + nt & \text{if } G_p \leq N_g - nt, \\
N_g & \text{otherwise.} 
\end{cases}
\]

for \(p = 1, 2, \ldots, p \times N_x \times N_y / 100\), where \(nt\) represents the magnitude of added noise.

5.2. Random distribution of random noise

In this kind of noise, a predefined percentage is also randomly determined and random noise is injected in the corresponding gray value. Suppose the magnitude of added noise is represented by \(X = x\), where \(X\) is normally distributed. That is, \(X \sim N(m, \sigma)\), where \(m\) is the mean of the normal distribution and \(\sigma\) is the standard deviation of the normal distribution. If a pixel \(p\) with gray value \(G_p\) is selected randomly, its new gray value will be computed as follows:

\[
G_p = \begin{cases} 
G_p + x & \text{if } 0 \leq G_p \leq N_g - x, \\
N_g & \text{otherwise.} 
\end{cases}
\]

5.3. Cut mark

In this kind of noise, any two points in the fingerprint image are selected randomly and the pixels lying on the line of width \(b_w\) joining these two points are set to be the largest gray value, \(N_g\). In other words, \(G_p = N_g\) for all pixels lying along the line of width \(b_w\) in order to model a cut mark on the fingerprint image. These cut marks are produced in two distinct orientations: along the left and right diagonals through the image such that they are 90° apart. They are named in the forward and reverse directions, respectively.

5.4. Missing information

Missing information means the loss of a certain portion of a fingerprint image. To simulate it, a fixed portion of fingerprint image is selected randomly. Then set all the pixels within this portion to be the largest gray value, \(N_g\), or the smallest gray value, 0, randomly. Therefore, \(G_p = N_g\) (0) for all pixels \(p\) lying within the randomly selected portion of the image. Note that setting \(G_p = N_g\) simulates the situation for insufficient inking of the fingerprint in the specified region, whereas setting \(G_p = 0\) simulates the condition of excess inking or blotches.

6. Experimental results

In order to testify the proposed approach, the fingerprint verification system is implemented using C language under LINUX environments. Our system consists of three main modules: preprocessing module, feature extraction module, and matching module. Preprocessing module eliminates noisy effect and redundant information in a fingerprint image. The following feature extraction module extracts 24 attributes. The matching module will then select a fingerprint image in fingerprint template database. An initialized compatibility matrix \(W\) between the two fingerprint images is produced by the augmentation algorithm. The randomized greedy algorithm iteratively searches the near optimal solution. Fig. 4 illustrates the system flowchart of the proposed approach.

Table 5 tabulates the number of iterations required by the proposed greedy algorithm. The average number of iteration is generally fewer if the feature point clusters of an input fingerprint image is the same as that of the prototype fingerprint image. Totally, there are 100 fingerprint images from 10 persons. These images are scanned from the inkless fingerprint scanner FC-100 of Startek. Images are sized in 570 \times 570 pixels. Shown in Fig. 5 are some example samples. The prototype database includes 1100 fingerprint images from 10 persons. Each image of a fingerprint images of a person produces 6 noise-contaminated images and 4 slightly clined fingerprint
images. 10°-left-clined and 10°-right-clined fingerprint images were used for training the threshold of acceptance (or rejection). Besides, uncontaminated, 15°-left-clined and 15°-right-clined fingerprint images were used for measuring the accuracy of the type-I error rate. There are 300 training fingerprint images and 300 testing fingerprint images for measuring the type-I error rate. Images of the random noise, Gaussian noise, cut make forward,
cut mark reversed, information loss black, and information loss white, were used for measuring the type-II error rate. For each fingerprint image, 6 imitated fingerprint images made for measuring type-II error rate were collected. So there are 600 forgery fingerprint images (from the noise-contaminated images) collected in the data base to evaluate the performance of the type-II error rate.

The creation of the reference fingerprint image is discussed as follows. In the learning stage, due to the limited amount of training data, each of the training fingerprint of a person was selected as a reference fingerprint and compared with his (or her) every other training data (so they were treated as the testing fingerprints). The similarities of those pairedwise fingerprint images are captured by adopting the compatibility function \( z \). The difference of fuzzy numbers are captured for the reference fingerprint. The fingerprint that has the least sum of the difference is selected as the best reference for a person. Let \( m_{ij} \) and \( d_{ij} \) denote the mean and standard deviation of the compatibility differences calculated above for the image of the \( i \)th persons \( j \)th fingerprint. The acceptance threshold \( \hat{\lambda}_{ij} \) is defined by

\[
\hat{\lambda}_{ij} = m_{ij} + Kd_{ij},
\]

where \( K \) is a positive constant.

In the testing stage, if the difference between the reference and the testing fingerprint of the \( i \)th fingerprint is less than or equal to the threshold \( \hat{\lambda}_{ij} \), the latter is accepted as a genuine fingerprint; otherwise, it is rejected. The type-I and type-II error rates for the experiments with different \( K \) values ranging from 0 to 4 at intervals of 0.3 are summarized in Table 6. Fig. 6 shows the type-I and type-II error rate diagram. In considering both the type-I and type-II error rate, \( K = 2 \) may be a good choice.

Verification errors in the experiments may occur mainly due to the following reasons. (1) Geometrical distribution diagram of the feature point clusters of fingerprint images are very similar. (2) One feature point cluster in a fingerprint image is splitted into more than one feature point cluster in another fingerprint image after preprocessing. (3) Variations of orientation is drastically large.

### 7. Conclusion

In this paper, we propose a new approach for fingerprint verification. The crucial point to the success of this new model is the proper choice of feasibility criterion \( z \). Feasibility criteria \( z \) is composed of combination of t-norms and t-conorms. In finding the criterion of feasibility \( z \), the tested t-norms and t-conorms \([12]\) are logical product, Hamacher product, algebraic product, Einstein product, bounded product, logical sum, Hamacher sum, algebraic sum, Einstein sum and bounded sum. These logical operators are recommended for use in the order of the logical product and logical sum, bounded product and bounded sum, Einstein product and Einstein sum, algebraic product and algebraic sum, and Hamacher product and Hamacher sum. The logical product and logical sum gives the same testing pairing the largest difference measure, and then the bounded product and bounded sum. Finally, it is algebraic product and algebraic sum.

After performing the module of preprocessing and feature point clustering, a number of feature point clusters forms. The similarity measuring input fingerprint images is defined by the fuzzy numbers of the \( \text{NOB, NOE, WH, OT, SL}_k \). It is possible that some ridge lines may not appear in the prototype fingerprint image, and vice versa. Therefore, some feature point clusters are lost after clustering. As a result, the Hungarian method or an augmenting path algorithm cannot be applied directly. To overcome this problem, the cost of unmatched node is introduced. Our matching goal becomes finding a matching such that the sum of weights of compatibility is minimum. The matching result produced by the Hungarian method or the augmenting path algorithm depends on the given costs (weights) and penalty between two associated nodes. For a good matching, the distance between feature point clusters and the penalty of unmatched feature point clusters is carefully given. Next, a new greedy algorithm which is proposed based on Hungarian method or the augmenting path algorithm is devised to restrict the optimal matching satisfying the constraints of the geometric relation. After applying the proposed algorithm, we can find a stable matching that

---

**Table 5**
The average number of iterations required by the proposed algorithm

<table>
<thead>
<tr>
<th>( N_L ) Node #</th>
<th>( N_R ) Node #</th>
<th>Iter. #</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>16</td>
<td>3</td>
</tr>
<tr>
<td>18</td>
<td>17</td>
<td>4</td>
</tr>
<tr>
<td>18</td>
<td>18</td>
<td>3</td>
</tr>
<tr>
<td>18</td>
<td>19</td>
<td>5</td>
</tr>
<tr>
<td>18</td>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>19</td>
<td>17</td>
<td>3</td>
</tr>
<tr>
<td>19</td>
<td>18</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>19</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>19</td>
<td>21</td>
<td>5</td>
</tr>
<tr>
<td>20</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>19</td>
<td>2</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>21</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>22</td>
<td>2</td>
</tr>
</tbody>
</table>

---
Fig. 5. The original fingerprint samples of a person.

Fig. 6. Type-I and type-II error rates in experiments.

preserves the geometric relations. In experiments, 100 fingerprint images were selected as the prototype fingerprint images. Verification results decay with a higher setting threshold. There are some suggestions for further improvement on the performance: (1) Focus on a good feature extraction module. (2) Improve the implementation of Hungarian method or the augmenting path algorithm, such as decomposing the complete graph into many smaller graphs. (3) Add more structural restrictions to the matching, such as Up–Down, Left–Right relations.
Table 6
A summarized table of Type-I and Type-II error rates for experiments

<table>
<thead>
<tr>
<th>K</th>
<th>Type-I error rate (%)</th>
<th>Type-II error rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>42.67</td>
<td>0.83</td>
</tr>
<tr>
<td>0.3</td>
<td>21.00</td>
<td>1.17</td>
</tr>
<tr>
<td>0.6</td>
<td>13.33</td>
<td>1.50</td>
</tr>
<tr>
<td>0.9</td>
<td>11.67</td>
<td>1.67</td>
</tr>
<tr>
<td>1.0</td>
<td>10.67</td>
<td>2.00</td>
</tr>
<tr>
<td>1.3</td>
<td>10.33</td>
<td>2.50</td>
</tr>
<tr>
<td>1.6</td>
<td>8.33</td>
<td>3.67</td>
</tr>
<tr>
<td>1.9</td>
<td>3.67</td>
<td>4.50</td>
</tr>
<tr>
<td>2.0</td>
<td>3.00</td>
<td>4.83</td>
</tr>
<tr>
<td>3.0</td>
<td>2.00</td>
<td>24.17</td>
</tr>
<tr>
<td>4.0</td>
<td>1.67</td>
<td>54.17</td>
</tr>
</tbody>
</table>

References


About the Author—KUO-CHIN FAN was born in Hsinchu, Taiwan, on 21 June 1959. He received his B.S. degree in Electrical Engineering from National Tsing-Hua University, Taiwan, in 1981. In 1983 he worked for the Electronic Research and Service Organization (ERSO), Taiwan, as a Computer Engineer. He started his graduate studies in Electrical Engineering at the University of Florida in 1984 and received the M.S. and Ph.D. degrees in 1985 and 1989, respectively. From 1984 to 1989 he was a Research Assistant in the Center for Information Research at University of Florida. In 1989, he joined the Institute of Computer Science and Information Engineering at National Central University where he became professor in 1994. He was the chairman of the department during 1994–1997. Currently, he is the director of Software Research Center at National Central University. Professor Fan is a member of IEEE, and a member of SPIE. His current research interests include image analysis, pattern recognition, and computer vision.

About the Author—CHENG-WEN LIU was born in Taijong, Taiwan, on 7 April 1970. He received his B.S. degree in Electrical Engineering from National Central University in 1992. He started his graduate studies in Computer Science at the National Central University in 1992. He is currently a Ph.D. student with research interests in fingerprint recognition, image processing, and pattern recognition.

About the Author—YUAN-KAI WANG was born in Nantou, Taiwan, Republic of China, on October 5, 1968. He received the B.S. degree in electrical engineering in 1990 and Ph.D. degree in computer science in 1995, respectively, all from National Central University, Chung-Li, Taiwan. From 1995 to 1999 he was a postdoctoral fellow and project leader in the Institute of Information Science, Academia Sinica, Taiwan. Currently, he is an associate professor in the department of electronic engineering at Fu-Jen Catholic University. Dr. Wang is a member of IEEE and a member of the Chinese Image Processing and Pattern Recognition Society. He was a winner of the Best Paper Award in both 1994 and 1998 IPPR conference on computer vision, graphics, and image processing. His current research interests include pattern recognition, genetic algorithms, image processing, and natural language understanding.