Unsupervised Fingerprint Recognition

Wei-Ho TSAI¹, Jun-Wei LIN¹, Nonmembers, and Der-Chang TSENG†(a), Member

SUMMARY This study extends conventional fingerprint recognition from a supervised to an unsupervised framework. Instead of enrolling fingerprints from known persons to identify unknown fingerprints, our aim is to partition a collection of unknown fingerprints into clusters, so that each cluster consists of fingerprints from the same finger and the number of generated clusters equals the number of distinct fingers involved in the collection. Such an unsupervised framework is helpful to handle the situation where a collection of captured fingerprints are not from the enrolled people. The task of fingerprint clustering is formulated as a problem of minimizing the clustering errors characterized by the Rand index. We estimate the Rand index by computing the similarities between fingerprints and then apply a genetic algorithm to minimize the Rand index. Experiments conducted using the FVC2002 database show that the proposed fingerprint clustering method outperforms an intuitive method based on hierarchical agglomerative clustering. The experiments also show that the number of clusters determined by our system is close to the true number of distinct fingers involved in the collection.

key words: clustering, fingerprint, Rand index, unsupervised

1. Introduction

The fingerprint has long been an important biometric feature for personal identification. With the increasing demand on various security and privacy applications nowadays, development of automatic techniques for fingerprint recognition [1]–[4] is gaining in importance. In general, fingerprint recognition can be divided into two categories: verification and identification, where the former aims to determine whether a captured fingerprint belongs to a particular person, whereas the latter aims to determine which of the persons a captured fingerprint belongs to. Fingerprint identification is usually considered tougher than fingerprint verification, since its difficulty increases drastically with the increase in the number of enrolled persons and fingerprints in the database. To improve the accuracy and efficiency of a large-scale fingerprint identification system, a number of methods, such as pre-classification [5], [6], indexing [7], [8] and retrieval [9], [10], have been extensively studied. The common strategy is to avoid comparing a captured fingerprint with the entire fingerprint database, but with a subset of the database instead. Good reviews of fingerprint recognition can be found in previous works [11], [12].

However, although the techniques for large-scale fingerprint identification have become more and more effective, they are awkward for handling a situation where the captured fingerprints do not belong to any of the persons enrolled in the database. For example, suppose there are 100 fingerprints collected in a criminal scene, and unfortunately, all the fingerprints come from the people not enrolled in the database. Then, even if all the fingerprints are correctly judged as outliers (from unenrolled people) after 100 runs of identification, it is of little help to infer the identities of these fingerprints. Under this circumstance, further analyses by human examiners, e.g., comparing the fingerprints with those collected in other crime scenes, are often required. However, this takes time and energy.

Often, multiple fingerprints captured in a criminal scene may come from the same people. And, different but related criminal scenes might also contain fingerprints from the same people. Thus, if a collection of unknown fingerprints can be partitioned into clusters, where each cluster contains only one person’s fingerprint(s), then the human efforts required to examine the fingerprints can be greatly reduced to examining each cluster only. This motivates our research into developing automatic techniques for clustering unknown fingerprints based on their identities. We refer to this task as fingerprint clustering hereafter.

Given a set of $N$ unknown fingerprints, each comes from one of the $P$ distinct fingers, where $N \geq P$, and $P$ is also unknown, the specific aim of the fingerprint clustering is to produce a partitioning of the $N$ fingerprints into $M$ clusters such that $M \leq P$, and each cluster consists exclusively of fingerprints from only one finger. In contrast to the conventional supervised fingerprint recognition, which requires an enrollment process to establish templates of each known person, fingerprint clustering is considered a type of an unsupervised recognition, which is thus not limited to handling the fingerprints of the persons enrolled in the database.

To the best of our knowledge, no prior work has been done to investigate the problem of fingerprint clustering in question. As far as the methodology is concerned, the most closely related research is the biometric clustering in the field of speaker recognition, e.g. [14] and our previous work in [18], which aims to cluster speech utterances based on speaker voice characteristics. This study is inspired by our previous work in [18], which formulates the task of fingerprint clustering as a problem of minimizing the clustering errors characterized by the Rand index. Here, the index functions by a property that its minimum value occurs only when each cluster consists exclusively of fingerprints from
only one finger and the number of clusters equals the number of fingers involved in the fingerprint collection. Although the basic strategy of our clustering method stems from [18], the detail of estimating the Rand index in this work are different from that in [18], in order to better accommodate the clustering method to the characteristics of fingerprint. In addition, to minimize the estimated Rand index, we develop two optimization schemes, respectively, based on a genetic algorithm (GA) and tabu search (TS). Compared to [18], in which only GA-based optimization is performed, we introduce another type of optimization based on tabu search (TS). The notations used in this paper are defined as follows.

2. Problem Formulation

The notations used in this paper are defined as follows.

- \( \mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N \): \( N \) fingerprints to be clustered;
- \( s_1, s_2, \ldots, s_P \): \( P \) distinct fingers involved in the \( N \) fingerprints, where \( P \) is also unknown;
- \( c_1, c_2, \ldots, c_M \): \( M \) clusters to be generated;
- \( \phi_i \): the index of the finger producing \( \mathbf{X}_i \);
- \( h_i(M) \): the index of the cluster within \( M \) clusters to which \( \mathbf{X}_i \) is assigned;
- \( n_{ms} \): the number of fingerprints in \( c_m \);
- \( n_{sp} \): the number of fingerprints from \( s_p \);
- \( n_{mp} \): the number of fingerprints that are from \( s_p \) and clustered into \( c_m \).

Fingerprint clustering can be viewed as a problem of determining a set of indices \( \mathbf{H}(M) = \{h_1(M), h_2(M), \ldots, h_N(M)\} \) that satisfy \( h_i(M) = h_j(M) \) for any \( \mathbf{X}_i \) and \( \mathbf{X}_j \) from the same finger, and \( h_i(M) \neq h_j(M) \) for any \( \mathbf{X}_i \) and \( \mathbf{X}_j \) from different fingers.

Depending on the application, there are a number of ways to evaluate the performance of fingerprint clustering. In this study, we use two metrics: purity [14], [15] and the Rand index [16], [17]. Purity comes in two types, cluster purity and finger purity. Both types indicate the degree of correct clustering. Cluster purity represents the probability that if we pick any fingerprint from a cluster twice at random, with replacement, both of the selected fingerprints are from the same finger. Specifically, the purity for cluster \( c_m \) is computed by

\[
\rho_m(\mathbf{H}(M)) = \frac{1}{P} \sum_{p=1}^{P} \left( \frac{n_{mp}}{n_{ms}} \right)^2.
\]  \hspace{1cm} (1)

From Eq. (1), it follows that \( n_{ms}^{-1} \leq \rho_m(\mathbf{H}(M)) \leq 1 \), in which the upper bound and lower bound reflect that all the within-cluster fingerprints are from the same finger and completely different fingers, respectively. To evaluate the overall performance of \( M \)-clustering, we compute average cluster purity

\[
U^C(\mathbf{H}(M)) = \frac{1}{N} \sum_{m=1}^{M} n_{ms} \rho_m(\mathbf{H}(M)).
\]  \hspace{1cm} (2)

Obviously, a perfect clustering should satisfy \( U^C(\mathbf{H}(M)) = 1 \). However, this does not work both ways. The value of \( U^C(\mathbf{H}(M)) \) generally increases as the value of \( M \) increases, since the metric does not consider errors that place fingerprints from the same fingers in different clusters. To reflect such errors, another type of purity, finger purity, is defined as the probability that if we pick any fingerprint from a different finger at random, with replacement, both of the selected fingerprints are clustered into the same cluster. Specifically, the purity for finger \( s_p \) is computed by

\[
\phi_p(\mathbf{H}(M)) = \frac{1}{P} \sum_{m=1}^{P} \left( \frac{n_{mp}}{n_{sp}} \right)^2.
\]  \hspace{1cm} (3)

From Eq. (3), it follows that \( n_{sp}^{-1} \leq \phi_p(\mathbf{H}(M)) \leq 1 \), in which the upper bound and lower bound reflect that all the fingerprints from finger \( s_p \) are clustered into the same cluster and different clusters, respectively. Considering \( P \) fingers, we compute average finger purity

\[
U^F(\mathbf{H}(M)) = \frac{1}{N} \sum_{p=1}^{P} n_{sp} \phi_p(\mathbf{H}(M)).
\]  \hspace{1cm} (4)

As opposite to \( U^C(\mathbf{H}(M)) \), the value of \( U^F(\mathbf{H}(M)) \) generally decreases as the value of \( M \) increases. Thus, to compare the performances of different clustering approaches generating different number of clusters, we combine cluster purity and finger purity into a single metric:

\[
U(\mathbf{H}(M)) = \sqrt{U^C(\mathbf{H}(M)) \cdot U^F(\mathbf{H}(M))}.
\]  \hspace{1cm} (5)

In general, the value of \( U(\mathbf{H}(M)) \) increases with the increase in the number of clusters initially, but decreases gradually when too many clusters are generated. The largest value of \( U(\mathbf{H}(M)) \) should occur when \( M = P \).

On the other hand, the Rand index indicates the degree of incorrect clustering. It is defined as the number of fingerprint pairs from the same finger that are placed in different clusters, or fingerprint pairs from different fingers that are placed in the same cluster, i.e.,

\[
R(\mathbf{H}(M)) = \sum_{m=1}^{M} n_{ms}^2 + \sum_{p=1}^{P} n_{sp}^2 - 2 \sum_{m=1}^{M} \sum_{p=1}^{P} n_{mp}^2.
\]  \hspace{1cm} (6)

Obviously, the smaller the value of \( R(\mathbf{H}(M)) \), the better the
clustering performance will be. The value of $R(H(M))$ usually decreases with an increase in the value of $M$ initially, and reaches the minimum at $M = P$. When $M > P$, the value of $R(H(M))$ starts to increase as the value of $M$ increases. The Rand index can be alternatively represented as a mis-clustering rate:

$$E(H(M)) = \frac{\sum_{m=1}^{M} n_{m}^2 + \sum_{p=1}^{P} n_{mp}^2 - 2 \sum_{m=1}^{M} \sum_{p=1}^{P} n_{mp}^2}{\sum_{m=1}^{M} n_{m}^2 + \sum_{p=1}^{P} n_{mp}^2}. \tag{7}$$

3. Methodology

Our basic strategy is to determine a set of indices $H(M) = \{h_1(M), h_2(M), \ldots, h_N(M)\}$ for the $N$ fingerprints to be clustered, such that the resulting purity is maximized or the resulting Rand index is minimized, where $h(M), 1 \leq i \leq N$, is an integer between 1 and $M$, and the value of $M$ is to be determined. This can be achieved by first representing the purity or Rand index as a function of the indices, and then minimizing or maximizing it with respect to the indices. As we found it difficult to represent the purity as a function of the indices, only Rand index is considered as a criterion to find the optimal indices. Adapted from our previous work reported in [18], this study derives an approximate Rand index in consideration of the similarities between fingerprints.

Inspect Eq. (6). It can be found that the terms

$$\sum_{m=1}^{M} n_{m}^2 = \sum_{m=1}^{M} \sum_{i=1}^{N} \delta(h_i(M), m)^2$$

$$= \sum_{m=1}^{M} \sum_{i=1}^{N} \delta(h_i(M), m)\left[\sum_{j=1}^{N} \delta(h_j(M), m)\right]$$

$$= \sum_{m=1}^{M} \sum_{j=1}^{N} \delta(h_i(M), m)\delta(h_j(M), m)$$

$$= \sum_{j=1}^{N} \sum_{i=1}^{N} \delta(h_i(M), h_j(M)),$$

$$\sum_{p=1}^{P} n_{mp}^2 = \sum_{p=1}^{P} \sum_{i=1}^{N} \delta(o_i, p)^2$$

$$= \sum_{p=1}^{P} \sum_{i=1}^{N} \delta(o_i, p)\left[\sum_{j=1}^{N} \delta(o_j, p)\right]$$

$$= \sum_{p=1}^{P} \sum_{j=1}^{N} \delta(o_i, p)\delta(o_j, p)$$

$$= \sum_{j=1}^{N} \sum_{i=1}^{N} \delta(o_i, o_j),$$

\[\dagger\]

$$\sum_{m=1}^{M} \sum_{p=1}^{P} n_{mp}^2 = \sum_{m=1}^{M} \sum_{p=1}^{P} \sum_{i=1}^{N} \delta(h_i(M), m)\delta(o_i, p)^2$$

$$= \sum_{m=1}^{M} \sum_{p=1}^{P} \sum_{i=1}^{N} \delta(h_i(M), m)\delta(o_i, p)\left[\sum_{j=1}^{N} \delta(h_j(M), m)\delta(o_j, p)\right]$$

$$= \sum_{m=1}^{M} \sum_{p=1}^{P} \sum_{i=1}^{N} \delta(h_i(M), m)\delta(o_i, p)\delta(h_j(M), m)\delta(o_j, p).$$

\[\dagger\]

where $\delta(\cdot)$ in Eqs. (8)–(10) is a Kronecker Delta function. Thus, the optimal set of cluster indices can be determined by

$$H^* = \arg \min_{H(M), 1 \leq M \leq N} R(H(M)) \tag{11}$$

and

$$R(H(M)) = \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(h_i(M), h_j(M)) + \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(o_i, o_j) - 2 \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(h_i(M), h_j(M))\delta(o_i, o_j). \tag{12}$$

However, as the computation of $\delta(o_i, o_j)$ in Eq. (12) requires that the groundtruth of each fingerprint be known in advance, it is impossible to find $H^*$ directly from Eqs. (11) and (12). To solve this problem, we estimate $\delta(o_i, o_j)$ based on the similarity between $X_i$ and $X_j$. Specifically,

$$\hat{\delta}(o_i, o_j) \approx \delta(o_i, o_j)$$

$$= \begin{cases} 1, & \text{if } i = j \\ S(X_i, X_j)/S_{\text{max}}, & \text{if } i \neq j \text{ and } S_{\text{max}} > 0, \\ 0, & \text{if } i \neq j \text{ and } S_{\text{max}} = 0 \end{cases} \tag{13}$$

where $S(X_i, X_j)$ denotes a certain similarity measure between $X_i$ and $X_j$ that is positive, and $S_{\text{max}}$ is the largest of the similarity measures $S(X_i, X_j), \forall i \neq j$. Hence, the approximation of $\delta(o_i, o_j)$ in Eq. (13) is a value between 0 and 1. In addition, recognizing the inevitable difference between $\delta(o_i, o_j)$ and $\hat{\delta}(o_i, o_j)$, we introduce a factor $\alpha$ with value around 1 in the estimation of the Rand index, which modifies Eq. (12) by

$$R(H(M)) \approx \hat{R}(H(M))$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(h_i(M), h_j(M)) + \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(o_i, o_j)$$

$$- 2\alpha \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(h_i(M), h_j(M))\delta(o_i, o_j). \tag{14}$$

Accordingly, our aim is to find $H^*$ such that $\hat{R}(H(M))$ is

\[\dagger\] In [18], the term $\sum_{p=1}^{P} n_{mp}^2$ is assumed to be a constant, since it is not changed by any clustering results. However, considering the range of its value can be large and difficult to predict, this study expresses this term explicitly as the relations between fingerprints, in order to approximate the Rand index more accurately.
minimized. This task involves a combinatorial optimization problem. We investigate two potential algorithms, namely the Genetic Algorithm (GA) and Tabu Search (TS) in finding \( \mathbf{H}^* \). It should be noted that other optimization algorithms, such as Simulated Annealing (SA) or combined GA/SAMTS, can also be applied here. The purpose of this study is to provide feasible implementations using different optimization algorithms rather than compare the performances between different optimization algorithms.

3.1 Genetic Algorithm (GA)

The GA [19] explores a given search space in parallel by iteratively modifying a population of chromosomes. Each chromosome, encoded as a string of alphabets or real numbers called genes, represents a potential solution to a given problem. In our task, a chromosome is exactly a legitimate chromosome, encoded as a string of alphabets or real numbers, and a gene corresponds to a cluster index associated with a fingerprint.

As shown in Fig. 1, the GA optimization starts by randomly generating chromosomes \( \mathbf{H}^{(1)}(M^{(1)}), \mathbf{H}^{(2)}(M^{(2)}), \ldots, \mathbf{H}^{(Z)}(M^{(Z)}) \) according to a pre-defined population size, \( Z \), and the number of generated clusters, \( M^{(i)}, 1 \leq z \leq Z \). For example, to cluster seven fingerprints, we can generate chromosomes like \([1\ 1\ 1\ 2\ 2\ 3\ 3\ 3\ 4\ 4\ 4],\ [1\ 2\ 2\ 2\ 2\ 2\ 2\ 3\ 3\ 4\ 4\ 4],\ [1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 3\ 3\ 4\ 4\ 4] \), in which the number of generated clusters is 3, 4, and 2, respectively. Then, the fitness of all chromosomes is evaluated via the inverse of the estimated Rand index, i.e., \( F(\mathbf{H}^{(z)}(M^{(z)})) = 1/R(\mathbf{H}^{(z)}(M^{(z)})), 1 \leq z \leq Z \). Based on this evaluation, a particular group of chromosomes is selected from the population to generate offspring by subsequent recombination. To prevent premature convergence of the population, the selection operation is performed with the linear ranking scheme [20], which sorts chromosomes in decreasing order of fitness, and then assigns the expected number of offspring according to their relative ranking. Chromosomes with large fitness values produce several copies, while those with very small fitness values may be eliminated; hence, the total chromosome population size does not change.

Next, crossover between the selected chromosomes is performed by exchanging the substrings of two chromosomes at two randomly selected crossover points. For example, the crossover for chromosomes \([1\ 1\ 1\ 2\ 2\ 3\ 3] \) and \([1\ 2\ 3\ 1\ 2\ 3\ 2] \) generates \([1\ 1\ 3\ 1\ 2\ 3\ 3] \) and \([1\ 2\ 1\ 2\ 2\ 3\ 2] \) respectively, if the selected crossover points are 2 and 6 (indicated by the underlined parts). After the crossover operation, a mutation operator is used to introduce random variations into the genetic structure of the chromosomes. To do this, we generate a random number and then replace one gene of an existing chromosome with a mutation probability. The resulting chromosomes that do not conform to the baseform representations are converted into their baseform counterparts. Then, the fitness evaluation, selection, crossover, and mutation steps are repeated continuously, in the hope that the overall fitness of the population will increase from generation to generation. When a pre-set maximum number of generations, say \( Q \), is reached, the best chromosome in the final population is taken as the solution, \( \mathbf{H}^* \). The optimal set of cluster indices also indicates the optimal number of generated clusters, which corresponds to the estimated number of fingers in the fingerprint collection.

However, since the index of one cluster can be interchanged with that of another cluster, multiple chromosomes may yield an identical clustering result. For example, the chromosomes \([1\ 1\ 1\ 2\ 2\ 3\ 3] \), \([1\ 1\ 1\ 3\ 3\ 2\ 2] \), \([2\ 2\ 2\ 1\ 1\ 3\ 3] \), and \([1\ 1\ 5\ 5\ 4\ 4] \) represent the same clustering result derived by grouping seven fingerprints into three clusters. Such a non-unique representation of the solution would significantly increase the GA search space, and may lead to an inferior clustering result. To avoid this problem, we limit the inventory of chromosomes to conform to a baseform representation defined as follows.

Let \( I(c_m) \) be the lowest index of the fingerprint in cluster \( c_m \). Then, a chromosome is a baseform

\[
\text{iff } \forall \ c_m, c_l \neq \{\emptyset\}, \text{if } m < l, \text{then } I(c_m) < I(c_l), \tag{15}
\]

where \( \{\emptyset\} \) indicates that a cluster does not contain any fingerprint. Among the above chromosomes, \([1\ 1\ 1\ 2\ 2\ 3\ 3] \) is a baseform, since the lowest index of the fingerprint in clusters \( c_1, c_2, \) and \( c_3 \) is 1, 4, and 6, respectively, which satisfies Eq. (15). In contrast, chromosomes \([1\ 1\ 1\ 3\ 3\ 2\ 2] \) and \([2\ 2\ 2\ 1\ 1\ 3\ 3] \) are not baseforms, since the lowest index of the fingerprint in clusters \( c_1, c_2, \) and \( c_3 \) is 1, 6, and 4, respectively, which does not satisfy Eq. (15). Meanwhile, chromosome \([1\ 1\ 1\ 5\ 5\ 4\ 4] \) implies that clusters \( c_2 \) and \( c_3 \) do not contain any fingerprint; hence it is not a baseform either. Fortunately, it is conceivable that all the non-baseform chromosomes can be converted into a unique baseform rep-
3.2 Tabu Search (TS)

Tabu search (TS) is an iterative improvement algorithm that begins with some trial solution and attempts to find a better solution based on a greatest-descent principle. In contrast to the general descent algorithms, which often suffer from the trap of local optimums, TS has a good ability to escape local optimums by accepting occasional inferior solutions during the search, in the hope of reaching the best solution in the end. As accepting inferior solutions introduces a risk of returning to the solutions that have already been considered, TS uses a list of prohibited moves, known as a tabu list, to reduce revisiting recent solutions recorded in the list and thus save computational time.

The convergence of the TS often depends on the size of the tabu list. Long tabu list takes more time to verify the candidate solution, whereas short tabu list could cause frequent and wasteful revisits to previous solutions and thus be less effective in escaping local optima. Previous studies [22] have shown the appropriate tabu list size in many applications is between five and 12, with seven being the most popularly used. Therefore the tabu list size is chosen to be seven in this work.

In addition, if a solution under consideration is good enough but found to be tabu, TS may still allow the solution to be accepted. This mechanism is called the aspiration. In essence, the aspiration overrides the tabu list, in order to avoid missing the potential solutions. There are a number of different aspiration criteria used in the literature [21], [23]. The one we used in this study is to override the tabu list if the resulting solution yields lower value in Eq. (12) than the one obtained earlier for the same move.

The TS procedure for minimizing Eq. (12) is detailed as follows and Fig. 2.

- **Step 1**: Generate randomly a solution in analogous to the legitimate chromosome described in Sect. 3.1 as the current solution, and compute its resulting Rand index in Eq. (12). Set tabu list as empty and iteration counter \( k = 0 \).
- **Step 2**: Perform the move. Generate a set of trial solutions by randomly replacing one element of the current solution with random numbers between 1 and \( M \). Then, select the best trial solution that yields the smallest Rand index in Eq. (12).
- **Step 3**: Perform candidate evaluation. If the Rand index yielded by the best trial solution is smaller than that of the current solution, go to Step 4, else go back to Step 2.
- **Step 4**: Perform the tabu test. If the best trial solution is not in the tabu list, then accept it as a current solution, update the tabu list, and go to Step 6, else go to Step 5.
- **Step 5**: Perform the aspiration criterion test. If satisfied, then override the tabu state and go to Step 6, else go back to Step 2.
- **Step 6**: Perform the termination test. If \( k \) reaches the predefined maximum number of iterations, then stop, else set \( k = k + 1 \) and go back to Step 2.

4. Inter-Fingerprint Similarity Computation

To estimate the Rand index in Eq. (12), the prerequisite is to compute the similarities between fingerprints. As a professional fingerprint examiner relies largely on minute details of ridge structures, our approach for inter-fingerprint similarity computation is based on the topological structural matching of minutiae. Figure 3 shows the procedure of the inter-fingerprint similarity computation. It consists of preprocessing, feature extraction, and feature matching. The preprocessing component aims to remove non-fingerprint parts while distill those helpful for fingerprint recognition from each fingerprint image. The feature extraction component characterizes each fingerprint as a set of parameters. Finally, the feature matching component outputs the similarities between fingerprints, by comparing the parameters of fingerprints in a pairwise manner. Note that since there have been a copious of studies investigating the problem of measuring the similarities between fingerprints, this work does not intend to develop a brand new solution to this problem, but only to provide an example from numerous possibilities. All the three components for the inter-fingerprint similarity computation are derived from some prior work [24]–[30].
4.1 Preprocessing

Figure 4 shows the overall flowchart of the preprocessing for each fingerprint. It begins by computing the orientation field to help the subsequent processes locate the region of interest from a fingerprint image. We apply the gradient algorithm reported in [24] to the computation of the orientation field. In addition, it is assumed that each fingerprint image is represented by 8-bit gray scale and there is only one fingerprint present in the image. After the orientation field is computed, the fingerprint image is enhanced using the method in [25], so that fingerprints ridges can be more clear and continuous. Then, Otsu thresholding method [26] is used to separate fingerprint from its background. However, as a fingerprint image resulting from the gray-level thresholding may inevitably contain noises like stains, pores, or scratches, we use average filtering to reduce the noises. Furthermore, to improve the contrast between ridges and valleys, which may be degradated by average filtering, the fingerprint enhancement method in [25] is used then again. Next, we reduce the 8-bit grayscale image to 2-bit black and white image based again on Otsu thresholding method [26]. Finally, the fingerprint ridge is thinned into a single pixel width using the method in [27], so that it is easier to find the minutiae defined as ridge bifurcations and endpoints. Fig. 4 also shows an example of fingerprint after the preprocessing is performed.

4.2 Feature Extraction

This component begins by detecting each fingerprints minutiae and singular points if available. A singular point is defined as the core and delta, where the ridge orientation vanished or discontinued. It can be used to align two fingerprints when measuring their similarity. In this work, we apply the Poincare index-based method in [28] to detect the singular prints when measuring their similarity. In this work, we apply the Poincare index-based method in [28] to detect the singular points. As an example shown in Fig. 5, each minutia is denoted by a triplet of parameters: the x- and y-coordinates and the angle (θ) between the tangent to the ridge line at the minutia position and the horizontal axis.

4.3 Feature Matching

Let $\mathbf{F}^{(i)} = \{f_1^{(i)}, f_2^{(i)}, \ldots, f_{K^{(i)}}^{(i)}\}$ and $\mathbf{F}^{(j)} = \{f_1^{(j)}, f_2^{(j)}, \ldots, f_{K^{(j)}}^{(j)}\}$ be the feature representations of i-th and j-th fingerprints, where each feature, say $f_k^{(i)}$, consists of a triplet $(x_k^{(i)}, y_k^{(i)}, \phi_k^{(i)})$, $1 \leq k \leq K^{(i)}$, and $K^{(i)}$ and $K^{(j)}$ are the numbers of features in $\mathbf{F}^{(i)}$ and $\mathbf{F}^{(j)}$, respectively. The similarity between the two fingerprints is measured by counting the number of features that are “consistent” between them. Here, the consistency holds if the spatial distance ($SD$) between two aligned features is smaller than a given tolerance $\phi_s$ and the direction difference (DD) between the features is smaller than a given tolerance $\phi_d$:

$$SD(f_k^{(i)}, f_l^{(j)}) = \sqrt{(x_k^{(i)} - x_l^{(j)})^2 + (y_k^{(i)} - y_l^{(j)})^2} \leq \phi_s,$$

(16)
In this study, the features between two fingerprints are aligned using the method in [30]. Accordingly, the number of consistent features between $F_i$ and $F_j$ can be measured using

$$L = \sum_{k=1}^{K_i} \sum_{l=1}^{K_j} \delta'(f_i^k, f_j^l),$$

where

$$\delta'(f_i^k, f_j^l) = \begin{cases} 1, & \text{if } SD(f_i^k, f_j^l) \leq \phi_s \text{ and } DD(f_i^k, f_j^l) \leq \phi_d, \\ 0, & \text{otherwise.} \end{cases}$$

Then, the similarity between fingerprints $X_i$ and $X_j$ can be computed by

$$S(X_i, X_j) = \frac{2L}{K_i + K_j}.$$

Figure 6 shows an example of measuring the similarity between two fingerprints. The numbers of features in the left-sided and right-sided fingerprints of Fig. 6 are 19 and 32, respectively; and the number of consistent features between the two fingerprints is 2. Thus, their similarity is $2 \cdot 2 / (19 + 32) = 0.0784$.

5. Experiments

5.1 Database

The fingerprint database used in this study stems from the DB3 of FVC 2002 [13]. It consists of 800 fingerprint images collected from 100 distinct fingers using capacitive sensors, 8 impressions each. As the proposed clustering method contains tunable parameters, such as $\alpha$ in Eq. (14), there is a need of a development data set for tuning the parameters. To this end, we divided the database into two subsets, denoted by DB3-A and DB3-B, where the former consists of the first four impressions of each finger, and the latter consists of the last four impressions of each finger. Thus, each subset contains 400 fingerprint images from 100 distinct fingers. In Sect. 5.2.3, DB3-A was used as a test data set, while DB3-B was used as a development data set.
5.2 Experiment Results

5.2.1 Examining the Validity of the Inter-Fingerprint Similarity Computation

Before evaluating the performance of the proposed fingerprint-clustering systems, our first experiment was conducted to validate the inter-fingerprint similarity computation. The parameters $\phi_s$ and $\phi_d$ in Eqs. (16) and (17) were set to be $2\sqrt{2}$ and $2^\circ$ empirically. Through the use of DET plot [31], we can estimate the false acceptance rate (FAR) and false rejection rate (FRR) of a fingerprint-verification system based on our inter-fingerprint similarity computation, where FAR reflects the likelihood of grouping fingerprints from different fingers into the same cluster, and FRR reflects the likelihood of grouping fingerprints from the same fingers into the different clusters. Figure 7 shows the DET curves with respect to DB3-A and DB3-B. The equal error rate (EER), which FAR = FRR, falls between 4% and 8%. Although the performance of our inter-fingerprint similarity computation is not comparable with that of the state-of-the-art method reported in [32], it is our aim to examine the feasibility of fingerprint clustering, given the imperfect inter-fingerprint similarity computation.

5.2.2 Evaluating the Clustering Performance, Given that the Number of Fingers is Known

Next, we evaluated the performance of the proposed fingerprint-clustering systems. In using GA, the parameter values used for the maximum number of generations, the chromosome population size, the crossover probability, and the mutation probability were empirically determined to be 5000, 4000, 0.5, and 0.1, respectively. In using TA, the maximum number of iterations was set to be 4000. In addition, for the sake of performance comparison, we also implemented a fingerprint clustering system based on the agglomerative hierarchical clustering (HAC) method [16], which could be one of the most popular data clustering methods. The HAC-based system starts with each fingerprint in its own cluster and successively merging the most similar pairs of clusters until one cluster contains all the fingerprints. The similarity between two clusters, say $c_i$ and $c_j$, can be measured in the following heuristic measures:

(i) complete linkage $S_c(c_i, c_j) = \min_{X_n \in c_i, X_k \in c_j} S(X_n, X_k)$,

(ii) single linkage $S_s(c_i, c_j) = \max_{X_n \in c_i, X_k \in c_j} S(X_n, X_k)$,

(iii) average linkage $S_a(c_i, c_j) = \frac{\sum_{X_n \in c_i, X_k \in c_j} S(X_n, X_k)}{\#(X_n \in c_i, X_k \in c_j)}$,

where $\#(X_n \in c_i, X_k \in c_j)$ denotes the number of fingerprint pairs involved in the summation. However, HAC can only construct a hierarchy of clusterings with the number of clusters ranging from 1 to N. It does not contain a mechanism to determining how many clusters should be generated. Thus, this experiment compares the performances of the proposed system and HAC-based system by fixing the number of generated clusters.

Table 1 shows the results of partitioning 400 fingerprint images in each subset into 100 clusters. We can see

| Method                    | Purity | Mis-clustering Rate |
|---------------------------|--------|---------------------|
| HAC with complete linkage | 0.92   | 0.11                |
| HAC with average linkage  | 0.90   | 0.16                |
| HAC with single linkage   | 0.84   | 0.29                |
| Proposed method using GA | 0.95   | 0.09                |
| Proposed method using TS | 0.93   | 0.12                |

Table 1 Results of partitioning 400 fingerprints into 100 clusters.
(a) 400 fingerprints in DB3-A

| Method                    | Purity | Mis-clustering Rate |
|---------------------------|--------|---------------------|
| HAC with complete linkage | 0.92   | 0.11                |
| HAC with average linkage  | 0.81   | 0.37                |
| HAC with single linkage   | 0.87   | 0.25                |
| Proposed method using GA | 0.98   | 0.02                |
| Proposed method using TS | 0.95   | 0.07                |

(b) 400 fingerprints in DB3-B
from Table 1 that the performance of an HAC-based system varies significantly with the selection of different inter-cluster similarity measurements. In the case of clustering the fingerprints in DB3-A, average linkage performs better than the other two linkages, whereas in the case of clustering the fingerprints in DB3-B, average linkage performs worst among others. This indicates that the HAC-based systems are highly heuristic and lack stability. By contrast, we can see from Table 1 that the proposed systems consistently yield higher purities and lower mis-clustering rates than HAC-based systems using various linkages. This indicates the superiority of the proposed systems over the HAC-based systems. In addition, Table 1 also shows that the proposed system using GA performs slightly better than that using TS. However, it should be noted that the purpose of this work is not put on comparing the performances between GA and TS but on providing feasible implementations using different optimization algorithms. As there are many other sophisticated versions of GA or TS being proposed continually, we do not intend to conclude which one is better between GA and TS in this case.

Figure 8 shows the purity and mis-clustering rate as a function of the number of clusters generated by the proposed system using GA. We can also see from Fig. 8 that the purity increases with the increase in the number of clusters in the beginning, but declines gradually after an excess of around 100 clusters is created. On the contrary, the value of the mis-clustering rate decreases with the increase of the number of clusters initially, but increases gradually after an excess of around 100 clusters is created. Apparently, the maximum of purity and minimum of mis-clustering rate appear near the number of fingers involved. This indicates again that the optimal number of clusters is equal to the number of fingers involved, which is however, usually unknown and needed to be estimated.

5.2.3 Evaluating the Clustering Performance, Given that the Number of Fingers is Unknown

We then examined our system under the condition that the true number of involved fingers is unknown and must be estimated. Each subset was divided into five overlapping groups involving different numbers of fingers. The relationships between groups are shown in Fig. 9. We conducted clustering experiment for each groups separately, in order to examine if the optimal numbers of clusters determined using Eq. (11) could be close to the true number of fingers in each group. The value of $\alpha$ was determined to be 1.21,
Table 2 Results of estimating the number of fingers involved for each group in DB3-A.

| Group | # Fingprints | True # Fingers | # Fingers Estimated with the GA-based System | # Fingers Estimated with the GA-based System |
|-------|--------------|---------------|--------------------------------------------|--------------------------------------------|
| 1     | 40           | 16            | 12                                         | 0                                          |
| 2     | 80           | 20            | 22                                         | 23                                         |
| 3     | 200          | 50            | 57                                         | 44                                         |
| 4     | 320          | 80            | 89                                         | 86                                         |
| 5     | 400          | 100           | 112                                        | 115                                        |

Table 3 Performances of clustering the fingerprints of the five groups in DB3-A.

(a) GA-based system

| Group | # Clusters | True # Fingers | # Clusters Estimated | Purity | Mis-clustering Rate |
|-------|------------|----------------|----------------------|--------|---------------------|
| 1     | 1.0        | 1.0            | 1.0                  | 0.99   | 0.01                |
| 2     | 1.0        | 0.0            | 0.98                 | 0.01   | 0.96                |
| 3     | 0.95       | 0.96           | 0.96                 | 0.07   | 0.08                |
| 4     | 0.95       | 0.95           | 0.94                 | 0.08   | 0.08                |
| 5     | 0.95       | 0.95           | 0.93                 | 0.10   | 0.10                |

(b) TS-based system

| Group | # Clusters | True # Fingers | # Clusters Estimated | Purity | Mis-clustering Rate |
|-------|------------|----------------|----------------------|--------|---------------------|
| 1     | 1.0        | 1.0            | 1.0                  | 0.99   | 0.02                |
| 2     | 0.95       | 0.95           | 0.97                 | 0.04   | 0.04                |
| 3     | 0.98       | 0.98           | 0.94                 | 0.11   | 0.11                |
| 4     | 0.95       | 0.95           | 0.93                 | 0.17   | 0.17                |
| 5     | 0.95       | 0.95           | 0.90                 | 0.15   | 0.15                |

Based on the observations on clustering the five groups in subset DB3-B.

Table 2 shows the results of estimating the number of fingers involved for each group in DB3-A, according to $\alpha = 1.21$. It can be seen from Table 2 that the number of fingers estimated with either our GA-based system or TS-based system is around the true number of fingers involved in each group. Table 3 shows the performances of clustering each group in DB3-A. For the ease of comparison, we also list the clustering results obtained by setting the number of generated clusters to be the true number of fingers, i.e., column “# Clusters = True # Fingers”. We can see from Table 3 that the purities and mis-clustering rates achieved with the estimated numbers of fingers were close to those achieved with known numbers of fingers. Although the clustering results were not perfect, most of the fingerprints from the same fingers were grouped into the same clusters. This confirms the feasibility of using unsupervised clustering to index unknown fingerprints.

6. Conclusion

We have extended the conventional fingerprint recognition problem from supervised classification to an unsupervised clustering paradigm. Instead of enrolling fingerprints from known persons to identify unknown fingerprints, the proposed system aims to partition unknown fingerprints into clusters, such that fingerprints from the same fingers can be identified. This system is beneficial to deal with the fingerprints from the persons not enrolled in the database. We have proposed a clustering method based on the minimal Rand Index to determine which fingerprints should be clustered and how many clusters should be generated. Experiments show that the proposed fingerprint clustering method achieves higher purity and lower Rand Index than that based on hierarchical agglomerative clustering. Experiments also show that the number of clusters determined by our system is close to the true number of fingers involved in the fingerprint data, indicating that our system is capable of identifying fingerprints from the same unknown persons.

Despite the potential, the methods proposed in this study can only be regarded as a preliminary investigation in realistic fingerprint clustering applications. To be of more practical use, more work is needed to study the effectiveness and efficiency of clustering a large-scale fingerprint collection. In particular, it may be necessary to deal with the images containing multiple overlapping and non-overlapping fingerprints. Hence, techniques for segmenting and separating multiple fingerprints from an image would need to be further investigated.

References

[1] J. Gu, J. Zhou, and C. Yang, “Fingerprint recognition by combining global structure and local cues,” IEEE Trans. Image Process., vol.15, no.7, pp.1952–1964, 2006.
[2] H. Choi, K. Choi, and J. Kim, “Fingerprint matching incorporating ridge features with minutiae,” IEEE Trans. Information Forensics and Security, vol.6, no.2, pp.338–345, 2011.
[3] Y. Wang and J. Hu, “Global ridge orientation modeling for partial fingerprint identification,” IEEE Trans. Pattern Anal. Mach. Intell., vol.33, no.1, pp.72–87, 2011.
[4] D. Zhang, F. Liu, Q. Zhao, G. Lu, and N. Luo, “Selecting a reference high resolution for fingerprint recognition using minutiae and pores,” IEEE Trans. Instrum. Meas., vol.60, no.3, pp.863–871, 2011.
[5] X. Tan, B. Bhanu, and Y. Lin, “Fingerprint classification based on learned features,” IEEE Trans. Syst., Man Cybern., C, Appli. Rev., vol.35, no.3, pp.287–300, 2005.
[6] H.W. Sun, K.Y. Law, D. Gollmann, S.L. Chung, J.B. Li, and J.G. Sun, “Efficient fingercode classification,” IEICE Trans. Inf. & Syst., vol.E91-D, no.5, pp.1252–1260, May 2008.
[7] R. Cappelli, M. Ferrara, and D. Maltoni, “Fingerprint indexing based on minutia cylinder-code,” IEEE Trans. Pattern Anal. Mach. Intell., vol.33, no.5, pp.1051–1057, 2011.
[8] R. Cappelli, M. Ferrara, and D. Maio, “Candidate list reduction based on the analysis of fingerprint indexing scores,” IEEE Trans. Information Forensics and Security, vol.6, no.3, pp.1160–1164, 2011.
[9] K.C. and C.H. Leung, “Improvement of fingerprint retrieval by a statistical classifier,” IEEE Trans. Information Forensics and Security, vol.6, no.1, pp.59–69, 2011.
[10] X. Jiang, M. Liu, and A.C. Kot, “Fingerprint retrieval for identification,” IEEE Trans. Information Forensics and Security, vol.1, no.4, pp.532–542, 2006.
[11] D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar, Handbook of fingerprint recognition, 3rd ed., Springer, 2003.
[12] R. Cappelli, D. Maio, D. Maltoni, J.L. Wayman, and A.K. Jain, “Performance evaluation of fingerprint verification systems,” IEEE Trans. Pattern Anal. Mach. Intell., vol.28, no.1, pp.3–17, Jan. 2006.
[13] http://bias.csr.unibo.it/fvc2002/
[14] A. Solomonoff, A. Mielke, M. Schmidt, and H. Gish, “Clustering persons by their voices,” Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1998.
[15] L. Hubert and P. Arabie, “Comparing partitions,” J. Classification
[16] L. Kaufman and P.J. Rousseuw, Finding groups in data: An introduction to cluster analysis, Wiley, New York, 1990.

[17] W.M. Rand, “Objective criteria for the evaluation of clustering methods,” J. Amer. Stat. Assoc. vol.66, pp.846–850, 1971.

[18] W.H. Tsai and H.M. Wang, “Evolutionary Minimization of the Rand Index for Speaker Clustering,” J. Computer Speech and Language, vol.23, no.2, pp.165–175, 2009.

[19] D.E. Goldberg, Genetic algorithm in search, optimization and machine learning, Addison-Wesley, New York, 1989.

[20] J.E. Baker, “Adaptive selection methods for genetic algorithm,” Proc. International Conference on Genetic Algorithms and Their Applications, pp.101–111, 1985.

[21] F. Glover, “Future paths for integer programming and links to artificial intelligence,” Comput. Oper. Res., vol.13, no.5, pp.533–549, 1986.

[22] F. Glover, “Artificial intelligence, heuristic frameworks and tabu search,” Managerial Decis. Econ., vol.11, pp.365–375, 1990.

[23] J.A. Bland, and G.P. Dawson, “Tabu search and design optimization,” Comput. Aided Des., vol.23, no.3, pp.195–201, 1991.

[24] C. Dahai, J. Xiang, F. Feng, Z. Jun, G. Lei, and M. Weihua, “Comparative analysis of fingerprint orientation field algorithms,” Proc. International Conference on Image and Graphics, pp.796–801, 2009.

[25] S. Hatami, H. Reshad, K. Mahmoud, and A. Hossein, “Wavelet based fingerprint image enhancement,” Proc. IEEE Symposium on Circuits and Systems, vol.5, pp.4610–4613, 2005.

[26] N. Otsu, “A threshold selection method from gray level histogram,” IEEE Trans. Systems, Man, and Cybernetics, SMC-9, pp.62–66, 1979.

[27] T.Y. Zhang and C.Y. Suen, “A fast parallel algorithm for thinning digital patterns,” Commun. ACM, vol.27, no.3, pp.236–239, 1984.

[28] J. Li, W.Y. Yau, and H. Wang, “Singular points detection using interactive mechanism in fingerprint images,” Proc. International Conference on Control, Automation, Robotics and Vision, 2006.

[29] Y. Al-Najjar and A. Sheta, “Minutiae extraction for fingerprint recognition,” Proc. 5th International Multi-Conference on Systems, Signals and Devices, 2008.

[30] S.H. Chang, F.H. Cheng, W.H. Hsu, and G.Z. Wu, “Fast algorithm for point pattern matching: Invariant to translations, rotations and scale changes,” Pattern Recognit., vol.30, pp.311V-320, 1997.

[31] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, “The DET curve in assessment of detection task performance,” Proc. Eurospeech, vol.4, pp.1899–1903, 1997.

[32] http://bias.csr.unibo.it/fvc2002/results/resultsAvg.asp

Wei-Ho Tsai received his B.S. degree in electrical engineering from National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C., in 1995. He received his M.S. and Ph.D. degrees in communication engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 1997 and 2001, respectively. From 2001 to 2003, he was with Philips Research East Asia, Taipei, Taiwan, where he worked on speech processing problems in embedded systems. From 2003 to 2005, he served as a Postdoctoral Fellow at the Institute of Information Science, Academia Sinica, Taipei, Taiwan. He is currently a Professor at the Department of Electronic Engineering & Graduate Institute of Computer and Communication Engineering, National Taipei University of Technology, Taiwan. His research interests include spoken language processing and music information retrieval.