A Fingerprint Recognizer Using Interminutiae Binary Constraint Graph

F. Benhammadi, H. Hentous, and K. Bey Baghdad

Laboratory of Computer Science, Military Polytechnic School, BP 17, Bordj-El-Bahri, Algiers 16111, Algeria

Correspondence should be addressed to F. Benhammadi, benhammadif@yahoo.fr

Received 30 April 2007; Revised 26 August 2007; Accepted 4 December 2007

Recommended by Arun Ross

The main aim of this paper is to describe an inter-minutiae binary network-based global matching algorithm. It exploits absolute minutiae features represented by a binary constraint graph. A dynamic tolerance area in the neighborhood of the minutiae combined with the search-rearrangement method on the binary constraint graph is used to perform the global matching. Tests on the public fingerprint databases DB1 and DB2 FVC2002 showed that our approach outperforms those based on the forward tracking search only. The most relevant aspect of this algorithm is that it does not require implicit alignment of the two fingerprint images due to the strong dependencies between minutiae described by the constraint graph.

1. INTRODUCTION

During the last two decades, several automatic biometric systems have been designed to meet new requirements in civilian applications. Recall that fingerprints have been used for many centuries as a means for identifying people and many fingerprint matching approaches were reported in literature. They can be broadly classified into two categories: structural feature-based matching [1–4], which is the most popular and widely used, and image-based matching [5–7]. Both approaches are based on two major fingerprint matching principles: global and local minutiae matchings, which use the implicit or the explicit alignment step simultaneously with the classical features as absolute position, direction, ridge counts, and type of minutiae. For the prealignment phase, some authors avoid fingerprint alignment [8] by introducing an intrinsic coordinate system according to a portioned regular region defined by the orientation field. However, this approach has some limits such as reliably partitioning the fingerprint into regular regions and unambiguously defining intrinsic coordinate axes in poor-quality fingerprint images [9]. This matching uses two main approaches: absolute pre-alignment (explicit alignment) to find the point correspondences and the optimal minutiae alignment [10, 11] and relative pre-alignment (implicit alignment) which consists in explicitly aligning one or more corresponding points (minutiae or singular points) generally based on a generalized Hough transform [12]. On the other hand, the second matching technique attempts to match minutiae occurring within a small local area [13–18]. These approaches are robust to nonlinear distortion due to graph-based matching. The approach described by Chikkerur et al. [17] uses what they called K-plet representation to encode local neighborhood of a minutiae. Their method is based on dynamic programming to obtain candidate star pairs followed by a consolidation of the local matches based on a novel coupled breadth first search (BFS) algorithm. It propagates the local matches simultaneously in both fingerprints to ensure a reliably matching of all minutiae. The accuracy of this method depends critically on the way this local matching is started; while the approach proposed in [18] matches two fingerprints by solving a minimum cost flow problem.

These local matching algorithms have a high accuracy and efficiency to nonlinear deformation compared to global matching ones. For the latter, there has been several attempts to improve the matching process by using the elastic distortions model [13, 19–21]. Kovács-Vajna [13] have used a
triangular matching approach for the large deformations and demonstrated that small local distortions can result in large global deformations. The method proposed in [19] uses the thin plate spline (TPS) model to align each pair from different fingerprint impressions. Ross et al. [20] also introduce the average deformation using the TPS model which is based on minutiae pair alignment for several impressions of the same finger to improve the alignment minutiae points. They have mentioned that the incorporation of the finger-specific deformation model in the matching process shows the good performance compared to the conventional approaches. On the other hand, Jea and Govindaraju [21] propose the tolerance range errors (distance, orientation) according to the interminutiae distance values. These tolerance areas are defined by thresholds that vary with the length of the interminutiae line segment. They asserted that these tolerances provide robustness against spatial distortions.

In this paper, we address a new minutiae-link-based matching algorithm for fingerprint recognition without need for a prealignment step. This algorithm rests upon a new representation which is described through an interminutiae relational network using a binary constraint graph. The interminutiae constraint network is constructed from the fingerprint enrollment via the number of ridges crossed by the straight line segment connecting two minutiae and the two angles formed by their orientations and this line, respectively. Hence, with this representation, the minutiae features become invariant to geometric transformations, which allows to avoid prealignment in our matching algorithm. The global matching uses the search-rearrangement heuristic to check constraints between the matched minutiae by applying dynamic tolerance area. Furthermore, we have used two types of tolerance errors: the fixed and adaptive thresholds. Some experimental results are presented to demonstrate significant improvements in fingerprint matching accuracy on the two public fingerprint databases DB1 and DB2 [9].

The breakdown of the paper is as follows. We describe fingerprint features extraction in Section 2. Section 3 provides our fingerprint constraint graph representation. The matching process via minutiae correspondence computation and the binary constraints checking is presented in Section 4. Experiments carried out in this paper are reported in Section 5. Finally, a conclusion is drawn in Section 6.

2. FINGERPRINT FEATURES

For convenience, we use the minutia orientation redefinition introduced in our paper [10] and extract the conventional interminutiae ridge counts [14, 15]. Besides, we introduce two new features that we called relative interminutiae angles (noted as $\alpha_{ij}$ and $\alpha_{ji}$). These angles are formed by the straight line segment connecting two minutiae ($m_i, m_j$) and the two angles formed by this line and their orientations, respectively. As shown on Figure 1(a), each relative interminutiae angle is obtained considering the anticlockwise sense. It is worth to note that this representation is completely different from that described in [18] since they use the angle formed by the straight line segment connecting two minutiae ($m_i, m_j$) and the x-axis.

Figure 1(b) illustrates the main difference between the two representations. It is easily noticeable that for [18] we can obtain the same angle $(\phi_{ij})$ between $m_i$ and $m_j$ for two different fingerprints, but in our representation, relative interminutiae angles $\alpha_{ij}$ and $\alpha_{ji}$ are completely different in two fingerprint images.

3. INTERMINUTIAE BINARY CONSTRAINT GRAPH

In our matching algorithm, the fingerprint features are represented by the interminutiae constraint network. The main idea of our approach is to connect the minutiae features throughout the links, which will be governed by constraints. Thus, we define a new representation for fingerprint features using constraint network between all minutiae (binary constraint graph) to represent these links.

The binary interminutiae constraint graph consists of a set of minutiae (variables) $m_1, \ldots, m_n$ with respective features (domains) and a set of binary constraints. The binary constraint graph $G = (V, E)$ for interminutiae relations is the undirected graph, where $V$ is the set of variables which are represented by the minutiae and $E$ is the set of edges, where each edge corresponds to a binary constraint $C$ which describes the link between two minutiae (a minutia pair).

Let $T = \{m_i, i = 1, \ldots, n\}$ be the minutiae list extracted from the fingerprint image. Formally, the interminutiae constraint network is defined as follows:

1. a set of $n$ minutiae (variables): $m_i, i = 1, \ldots, n$,
2. for each minutia $m_i$, a domain $d_i = \{r_{ij}, a_{ij}\} | j = 1, \ldots, n, j \neq i$, exists, where $(r_{ij}, a_{ij})$ are previously computed by the preceding features (ridge counts and the relative angles) between the minutia $m_i$ and the other minutiae $m_j$, for $j \neq i$,
3. for each pair of the minutiae $\{m_i, m_j\}$, $i \neq j$, a binary constraint $C_{i,j}$, which is a subset of $d_i \times d_j$, exists. In our representation, every constraint takes the two features at the same time. On the other hand, each binary constraint is as follows:

$$C_{i,j} = \{(r_{ij}, r_{ji}), (a_{ij}, a_{ji})\}, \quad (1)$$

4. for each minutia $m_i$, the constraint relation $C_i$ is defined to describe the relation between minutia $m_i$ and the other ones as follows:

$$C_i = \bigcup_{j \neq i} C_{i,j}. \quad (2)$$

Figure 1: Relative interminutiae angles.
In the enrollment phase, the relational network for each minutia \( m_i \) is computed from the features (Euclidean distance, ridge counts, and relative interminutiae angles) between this minutia and other minutiae \( m_j \), for \( j \neq i \) and the fingerprint template is generated by the global interminutiae constraint graph which represents the minutiae relational network as shown in Figure 2. Let us note that for each minutiae pair \( (m_i, m_j) \), we have \( r_{ij} = r_{ji} \).

4. Binary Constraint Graph Matching

In this section, we introduce our global constraint matching algorithm which is used to find corresponding minutiae pairs of two fingerprints. There are two steps to match the interminutiae binary constraint graph. The first one is to compute minutiae pairing (the set of split matched minutiae). The second one is to compute the global binary constraints checking between matched minutiae from the split sets.

4.1. The Minutiae Pairing

The minutiae pairing allows identifying corresponding minutiae pairs from two constraint graphs. Let \( T = \{m_i, i = 1, \ldots, m\} \) and \( I = \{m'_j, j = 1, \ldots, n\} \) be the minutiae lists extracted from template and input fingerprints and let \( G_T = (V_T, E_T) \) and \( G_I = (V_I, E_I) \) be their constraint graphs, respectively. The minutiae pairing for two constraint graphs is described into the two following stages. First, we match vertices (minutiae belonging to the template and the input fingerprints). The matching function considered here is multivalent, that is, each minutia of the input fingerprint constraint graph is matched with a possibly empty set of minutiae of the template constraint graph. More formally, a matching of two constraint graphs \( G_I \) and \( G_T \) is a set \( M \subseteq V_T \times V_I \) which contains every pair \( (m_i, m'_j) \in V_T \times V_I \) such that the minutiae \( m_i \) is matched with the minutiae \( m'_j \). We now describe how two minutiae are matched.

Let \( d_i = \{(r_{ik}, \alpha_{ik}) \mid k = 1, \ldots, n, k \neq i\} \) and \( d_j = \{(r'_{jl}, \alpha'_{jl}) \mid l = 1, \ldots, m, l \neq j\} \) be the minutiae \( m_i \) and \( m'_j \) domains. For the relative interminutiae angles tolerance, we define the adjustable (adaptive) angles according to the relative interminutiae angle formed by the three straight line segments connecting two minutiae \( (m_i, m'_j) \) and the angle errors formed by these lines and their orientations, respectively, as follows:

\[
T_{ao}^- = \min \left( (D_1, D_3)[2\pi], (D_1, D_4)[2\pi], (D_2, D_3)[2\pi], (D_2, D_4)[2\pi] \right),
\]

\[
T_{ao}^+ = \max \left( (D_1, D_3)[2\pi], (D_1, D_4)[2\pi], (D_2, D_3)[2\pi], (D_2, D_4)[2\pi] \right).
\]

This adjustment overcomes the small non-linear deformation present in fingerprint image pairs, minutia orientation and localization error as illustrated in Figure 3.

Then the minutia \( m_i \) is matched with minutia \( m'_j \) if \( \exists \{r_{ik}, \alpha_{ik}\} \in d_i \) and \( \exists \{r'_{jl}, \alpha'_{jl}\} \in d_j \) that respect the following conditions:

\[
|r_{ik} - r'_{jl}| = 0,
\]

\[
T_{ao}^- \leq \alpha'_{jl} \leq T_{ao}^+.
\]

Given a minutia \( m_i \) segment, we compute the possible interminutiae angles at localization errors \( (A_i \text{ and } B_i) \) of each minutia according to minutia orientation errors \( (\theta_{i}^- \text{ and } \theta_{i}^+) \). \( (D_1, D_3)[2\pi] \) indicates the relative interminutiae angle between the lines \( (D_1) \) and \( (D_3) \). In our approach, the minutia localization error is ±15 pixels and the orientation error is ±15 degrees.
However, the large distortions (Figure 4) influence not only the minutiae ridge counts angles but also the ridge counts. Using the ridge count or distance in this case, the matching result is usually negative. Some authors have made recourse to the average deformation model as described in [20] to overcome these large distortions. This model permits to predistort minutiae points before matching process but the model required registration stage. Hence, in our approach, we have proposed strong constraint (see (4)) to compare ridge counts values. This latter allows matching each two minutiae according to the other nearest neighbor minutiae and the distortion influences this constraint for the long distance between minutiae.

In the second stage, we also have to identify the set of split matched minutiae, that is, the set of minutiae that are matched to more than one minutia. So, each minutia \( m' \) from the input fingerprints is associated with the set \( S_{m'} \) of its matched minutiae from the template fingerprint:

\[
\text{Split}(M) = \left\{ (m', S_{m'}) \mid m' \in V_T \text{ and } S_{m'} = \{ m' \in V_T \mid (m', m'_j) \in M \} \right\}
\]

\[(6)\]

4.2. The binary constraint checking

Once a set of split minutiae is computed, the next phase is to check the global constraints consistency for the matched minutiae. A state of the constraint graph matching problem is defined by an assignment of template minutiae to some or all of the input minutiae belonging to the set of split matched minutiae. An assignment that does not violate any constraint is called a consistent or legal consistent assignment. A complete assignment is one in which every minutia belonging to the input fingerprint is mentioned. Therefore, an assignment \( A_M = \{m'_1 - m'^1_k, m'_2 - m'^2_l, \ldots, m'^{\text{split}(M)}_j - m'^{\text{split}(M)}_i\} \) exists, where \( m'^1_k \in S_{m'_1}, m'^2_l \in S_{m'_2}, \ldots, m'^{\text{split}(M)}_j \in S_{m'^{\text{split}(M)}_i} \), is consistent if the constraint \( C_{ij} = \{(a_{ij}, a_{ij}), (r_{ij}, r_{ij})\} \) for each matched minutia pair \((m'_i, m'_j)\) is satisfied.

The backtracking search is used for the constraints checking that chooses a minutia \( m'_i \) from the set \( S_{m'_i} \) for one minutia \( m'_j \) at a time and backtracks when the minutia \( m'_j \) has no constraint consistency to the other minutiae belonging to \( S_{m'_j} \) for \( i \neq j \). We have chosen the backtracking search because the variable ordering and the value selection heuristics help significantly and the search may be systematic. Moreover, in some cases, this algorithm with search-rearrangement heuristics is more efficient solving due to specific solvers such as prolog that we have exploited in our approach. Part of the search tree for the global constraint checking is shown in Figure 5, where we have assigned the matched minutiae from input fingerprint in the order \( m'_1, m'_2, \ldots, m'^{\text{split}(M)}_j \), where \( |\text{split}(M)| \) represents the set of split matched minutiae cardinality.

The backtracking search algorithm is described in the following way:

We now proceed to give the complexity of our global matching algorithm. Assume that the domain size for each variable (each minutia \( m'_i \) from the set \( S_{m'_i} \) computed from the template fingerprint) is \( d \). The overall time-complexity of our backtracking search algorithm is widely considered to be \( O^* (d^n) \) such that \( n \) represents the matched minutiae from the input fingerprint for the interminutiae constraint graph. So, we express the time-complexity of our

**Figure 3:** The relative interminutiae angle error range.

**Figure 4:** Due to the nonlinear distortions, the minutiae pairing is not accurate: ridge counts errors.

**Figure 5:** Part of the search tree for the global constraint checking.
backtracking search algorithm by the number of partial solutions generated by the algorithm. This is a valid representation because the time complexity can be represented as the number of partial solutions multiplied by a polynomial factor which is ignored by the $O^*$ notation.

4.3. The score matching

A matching score for fingerprint features establishes a correspondence between the input constraint graph and all the graphs of the template database set. So, after the global constraints checking has been verified, a match score is computed based on the number of corresponding minutiae in the assignment sets. Otherwise, the match score represents the depth of the tree constructed from backtracking search which is the maximal cardinality of assignment sets. Let $A_M^j$ for $j = 1, \ldots, j(\text{Split}(M))$ be the assignment sets computed from the backtracking search algorithm. Then, the match score is

$$Ms = \max \left( \left| A_{M}^1 \right|, \left| A_{M}^2 \right|, \ldots, \left| A_{M}^{j(\text{Split}(M))} \right| \right), \quad (7)$$

where $\left| A_{M}^1 \right|, \left| A_{M}^2 \right|, \ldots, \left| A_{M}^{j(\text{Split}(M))} \right|$ are the cardinality of the assignment sets.

Our backtracking search does not specify explicitly the order of the minutiae to be assigned. Experiments and analysis by several researchers show that the ordering in which variables are chosen for instantiation can have substantial impact on the complexity of backtracking search [22]. This job is done by ordering heuristics. Several heuristics have been developed and analyzed for selecting variable ordering. The simplest and the most successful ordering heuristic is called search-rearrangement method or Fail-Fist [24]. Therefore, we choose the minutiae with the fewest possible remaining alternatives (smallest set of split matched minutiae) and they are selected for instantiation at each step in the tree search. Thus, the order of minutia instantiation is, in general, different in different branches of the tree and is determined dynamically, that is, the implementation requires maintaining split set of each minutia which is updated dynamically when matched minutiae are removed or restored. The optimizing problem for matched minutiae consistency checking is to determine an order in which to perform the tests, which minimizes the expected number of tests performed. So, the selection of the matched minutiae (instruction: matched_minutia $\rightarrow m_j^i$) in the backtracking algorithm will be replaced by the selection of the minimal element among the split sets by choosing the minutia $m_j^i$ with the fewest values (the smallest set $S_{m_j^i}$ of its matched minutiae from the template fingerprint). The second choice is to try the values (matched minutia of $m_j^i$) according to the minimal ridge counts values. This last order allows to favor a local matching by choosing at each step the closest neighbors of the minutia $m_j^i$.

5. EXPERIMENTAL RESULTS

The tests were performed by our proposed fingerprint matching algorithm. Therefore, we show the experimental results in both verification and identification modes. The experimental fingerprints databases came from the publicly available DB1-a and DB2-a extracted from the Fingerprint Verification Competition (FVC2002) [9]. This gave a total of 100x8 distinct fingerprint images for each database. Besides, all experiments discussed in this section are conducted on a Pentium IV 3.2 GHz.

When verification or identification process is necessary, the binary constraint graph is extracted from the user’s presented finger, and it is compared to the binary constraint graphs (templates) in the database, as illustrated by the system flowchart in Figure 6.

5.1. Verification mode

For the evaluation of our fingerprint verification algorithm, we adopt the equal error rate (EER) that is an important indicator rate when the false acceptance rate (FAR) and the false rejection rate (FRR) are identical. For FAR, the total number of tests is $(100 \times 99)/2 = 4950$ and for FRR, the total number of tests is $(8 \times 7)/2 \times 100 = 2800$. Then, the EER
reported from the test are 2.31% on DB1-a and 1.42% on DB2-a as illustrated by Figure 7. So, we have a false rejection rate, FRR = 8.61% at false acceptance rate FAR = 0%.

Compared with the approach proposed in [18] on the database DB2-a, we can see that when 7% ≤ FAR ≤ 10%, our system achieves a lower FRR, while their approach achieves a lower FRR for FAR ≤ 7%. Moreover, the two methods all have high speed matching with small EERs 1.06% and 1.31% using our minutiae extraction module as illustrated in Figure 8. In addition, their algorithms have been developed by implementing the exact graph matching using the minimum cost flow method and the NN approach. In contrary, we have used inexact matching in our approach. Thus, the two approaches are suitable for common applications with lower security requirements. Finally, the weakness of our approach lies in the failure of the strong constraint (rigid tolerance) on ridge count comparison particularly in the case when we have the ridge count number with long distance between minutiae and for the deformable images when we have large distortions. However, our method depends on many implementation parameters to calculate these dynamic tolerances.
Table 1: Ranking of our approach on DB1-a and DB2-a databases of the top 31 participants in FVC2002.

| Criterion | EER | Enrollment process | Matching process |
|-----------|-----|--------------------|------------------|
|           |     | Time               | Rank             | Time               | Rank             |
| DB1-a     | 15  | 3.897              | 29               | 2.446              | 28               |
| DB2-a     | 12  | 2.017              | 30               | 2.009              | 27               |

Table 2: The probability of identification on DB1-a and DB2-a.

| Rank | DB1-a | DB2-a |
|------|-------|-------|
| 1    | 0.88372 | 0.95012 |
| 2    | 0.88479 | 0.95017 |
| 3    | 0.88615 | 0.95029 |
| 4    | 0.88789 | 0.95067 |
| 5    | 0.89198 | 0.96023 |
| 6    | 0.89207 | 0.96323 |
| 7    | 0.89663 | 0.96934 |
| 8    | 0.90934 | 0.97410 |
| 9    | 0.92408 | 0.97570 |
| 10   | 0.94026 | 0.98981 |

Table 3: The average matching time using search-rearrangement method.

| Ordering                           | Average matching time DB1-a (s) | Average matching time DB2-a (s) |
|------------------------------------|---------------------------------|---------------------------------|
| Variables (input minutiae)         | 1.094760                        | 0.894714                        |
| Domains (template minutiae)        | 1.057210                        | 0.829828                        |
| Both variables and domains         | **0.09871**                     | **0.14960**                     |

Figure 9: The CMC curves on DB1-a and DB2-a.

Table 1 shows the average computational time cost of our interminutiae graph matching and the rank of our approach compared to the results obtained by the different algorithms presented to the FVC2002 [26] competition from the same bases (according to the criteria: EER, average enrollment time, and average matching time). All times discussed in this section are conducted on a Pentium IV 3.2 GHz. According to the ranking rule in terms of EER, our constraint graph matching algorithm is in the first 15 and 12 places on DB1-a and DB2-a, respectively. However, in terms of computational time cost, we got less classified places.

5.2. Identification mode

In order to evaluate our constraint graph matching algorithm in the identification case, we report our results according to the following matching protocol. We have used 100 distinct fingerprint images as the reference base and the set of the 700 remaining fingerprints as the test base. The probability of correct identification at rank $n$ means that the probability of the correct match is somewhere in the top $n$ matching scores. A cumulative match characteristic (CMC) is used to show the probability of identification for numerous (inclusive) ranks [26]. The probability of identification achieved on DB1-a and DB2-a are shown in Table 2. Each CMC curve is estimated on the experimental databases DB1-a and DB2-a as shown in Figure 9. Comparison between CMCs shows that our constraint matching algorithm improves greatly the identification performance on DB2-a database with an identification probability of 0.95.

5.3. Heuristic speed improvements

All experiments discussed in the preceding sections are conducted using the backtracking search algorithm without specifying explicitly the order of the selection of minutiae to be assigned neither for the minutiae of the input fingerprint nor for that of template. However, another issue is the proposed rearrangement method (the order in which the matched minutiae from the input fingerprint are considered for instantiations and the order of the ridge count values).

For our constraint graph matching problem, three heuristics have been tested.

(i) Variables ordering: given the sets of matched minutiae from the input fingerprint, we have tried these sets according to the ascending order of their cardinality (split cardinality).

(ii) Domains ordering: given a matched minutia, how to order the ridge count values to try. So, we adopt the same strategy by ordering the matched minutiae from the template fingerprint.

(iii) Variables and domains ordering: the combination of the two preceding strategies.

We experimentally show in Table 3 that the search-rearrangement heuristics led to an improvement about three kinds of the matched minutiae orders for the identification mode on DB1-a and DB2-a. It can be seen that the average matching times reported from the test are 1.094760s on DB1-a and 0.894714s on DB2-a according to the ordered minutiae extracted from the input fingerprints, respectively. In addition, the matching time was also improved by the domains...
tries. But the combination of the two heuristics improved considerably the total time for all databases as indicated in Table 3.

Therefore, we show the efficiency of this search-rearrangement heuristic in our approach by choosing a good minutia ordering or a good order for the instantiation of different values (minutiae matching from the template fingerprint) of a given minutia. However, with these search-rearrangement heuristics, we got first to classify place in terms of computational time cost. Thus, the search-rearrangement heuristics can solve the critical problem of the minutia choice in local matching proposed in approach [18] because the accuracy of their approach depends critically on the way this local matching is started.

6. CONCLUSION

In this work, we have presented a fingerprint recognition system that uses the globally minutiae-based matching which is based on the constraint satisfaction problem resolution. Experiments indicate that the search-rearrangement method is successful enough for the fingerprint binary constraint graph matching and performs much better than a purely backtrack- ing search. The main advantage of our matching process is that it does not require an implicit alignment of two fingerprint images compared to the previous. In the near future, we will use the tree-structured local minutiae matching for the global constraint graph matching decomposition problem.

REFERENCES

[1] A. K. Jain, L. Hong, S. Pankanti, and R. Bolle, “An identity-authentication system using fingerprints,” Proceedings of the IEEE, vol. 85, no. 9, pp. 1365–1388, 1997.
[2] M. Tico and P. Kuosmanen, “Fingerprint matching using an orientation-based minutia descriptor,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 8, pp. 1009–1014, 2003.
[3] A. Luminii and L. Nanni, “Two-class fingerprint matcher,” Pattern Recognition, vol. 39, no. 4, pp. 714–716, 2006.
[4] J. Qi, S. Yang, and Y. Wang, “Fingerprint matching combining the global orientation field with minutia,” Pattern Recognition Letters, vol. 26, no. 15, pp. 2424–2430, 2005.
[5] K. A. Nagaty, “An adaptive hybrid energy-based fingerprint matching technique,” Image and Vision Computing, vol. 23, no. 5, pp. 491–500, 2005.
[6] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, “Finger-code: a filterbank for fingerprint representation and matching,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99), vol. 2, pp. 187–193, Fort Collins, Colo, USA, June 1999.
[7] A. Ross, A. Jain, and J. Reisman, “A hybrid fingerprint matcher,” Pattern Recognition, vol. 36, no. 7, pp. 1661–1673, 2003.
[8] A. M. Bazen and S. H. Gerez, “An intrinsic coordinate system for fingerprint matching,” in Proceedings of the 3rd International Conference on Audio and Video-Based Biometric Person Authentication (AVBPA '01), pp. 198–204, Halmstad, Sweden, June 2001.
[9] M. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition, Springer, New York, NY, USA, 2003.
[10] E. Benhammadi, M. N. Amirouche, H. Hentous, K. Bey Begendad, and M. Aissani, “Fingerprint matching from minutiae texture maps,” Pattern Recognition, vol. 40, no. 1, pp. 189–197, 2007.
[11] J. Qi and Y. Wang, “A robust fingerprint matching method,” Pattern Recognition, vol. 38, no. 10, pp. 1665–1671, 2005.
[12] N. K. Ratha, S. Chen, and A. K. Jain, “Adaptive flow orientation-based feature extraction in fingerprint images,” Pattern Recognition, vol. 28, no. 11, pp. 1657–1672, 1995.
[13] Z. M. Kovacs-Vajna, “A fingerprint verification system based on triangular matching and dynamic time warping,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1266–1276, 2000.
[14] X. Jiang and W. Y. Yau, “Fingerprint minutiae matching based on the local and global structures,” in Proceeding of the 15th International Conference on Pattern Recognition (ICPR 00), vol. 2, pp. 1042–1045, Barcelona, Spain, September 2000.
[15] S. N. Pradeep. M. D. Jain, R. Balasubramanian, and R. Bhar gava, “Local and global tree graph structures for fingerprint verification,” in Proceedings of the 24th IASTED International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA ‘06), pp. 287–293, Innsbruck, Austria, February 2006.
[16] H. Wei, Z. Ou, and J. Zhang, “Fingerprint identification based on ridge lines and graph matching,” in Proceedings of the World Congress on Intelligent Control and Automation (WCICA ’06), vol. 2, pp. 9965–9968, Dalian, China, June 2006.
[17] S. Chikkeruru, V. Govindaraju, and A. N. Cartwright, “K-plet and coupled BFS: a graph based fingerprint representation and matching algorithm,” in Proceedings of the International Conference on Biometrics (ICB ’06), pp. 309–315, Hong Kong, January 2006.
[18] T. Jea, Minutiae based partial fingerprint recognition, Ph.D. thesis, University at Buffalo, New York, NY, USA, 2005, http://www.cedar.buffalo.edu/~govind/Alan_Thesis.pdf.
[19] A. M. Bazen and S. H. Gerez, “Fingerprint matching by thin-plate spline modeling of elastic deformations,” Pattern Recognition, vol. 36, no. 8, pp. 1859–1867, 2003.
[20] A. Ross, S. Dass, and A. Jain, “A deformable model for finger- print matching,” Pattern Recognition, vol. 38, no. 1, pp. 95–103, 2005.
[21] T. Jea and V. Govindaraju, “A minutia-based partial finger- print recognition system,” Pattern Recognition, vol. 38, no. 10, pp. 1672–1684, 2005.
[22] R. Zabih and D. McAllester, “A rearrangement search strategy for determining propositional satisfiability,” in Proceedings of the 7th National Conference on Artificial Intelligence (AAAI '88), pp. 155–156, St. Paul, Minn, USA, August 1988.
[23] R. M. Haralick and G. L. Elliott, “Increasing tree search efficiency for constraint satisfaction problems,” Artificial Intelligence, vol. 14, no. 3, pp. 263–313, 1980.
[24] P. Purdom, “Search rearrangement backtracking and polynomial average time,” Artificial Intelligence, vol. 21, no. 1–2, pp. 117–133, 1983.
[25] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, “FVC2002: second fingerprint verification competition,” in Proceedings of the 16th International Conference on Pattern Recognition, vol. 3, pp. 811–814, Quebec, Canada, August 2002.
[26] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, “The FERET evaluation methodology for face-recognition algorithms,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 10, pp. 1090–1104, 2000.