Robust Fingerprint Authentication Using Local Structural Similarity

Nalini K. Ratha
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
ratha@us.ibm.com

Vinayaka D. Pandit
IBM India Research Lab
New Delhi, India
pvinayak@in.ibm.com

Ruud M. Bolle
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
bolle@us.ibm.com

Vaibhav Vaish
Indian Institute of Technology
New Delhi, India
vaibhav@cse.iitd.ernet.in

Abstract

Fingerprint matching is challenging as the matcher has to minimize two competing error rates: the False Accept Rate and the False Reject Rate. We propose a novel, efficient, accurate and distortion-tolerant fingerprint authentication technique based on graph representation. Using the fingerprint minutiae features, a labeled, and weighted graph of minutiae is constructed for both the query fingerprint and the reference fingerprint. In the first phase, we obtain a minimum set of matched node pairs by matching their neighborhood structures. In the second phase, we include more pairs in the match by comparing distances with respect to matched pairs obtained in first phase. An optional third phase, extending the neighborhood around each feature, is entered if we cannot arrive at a decision based on the analysis in first two phases. The proposed algorithm has been tested with excellent results on a large private livescan database obtained with optical scanners.

1. Introduction

The fingerprint-based *authentication* problem can be informally defined as a system that answers the question "Am I who I say I am?". That is, given an enrolled template from the database and the query template what is the probability that the two templates are from the same finger? In contrast to this, we have the *identification* problem, which can be defined as a system that answers the question "Is the subject enrolled in the database?" Equivalently, given a database of templates and the query template, does the query template closely resemble any of the database templates? The identification problem can be seen as N repeat authentications, where N is the number of candidates in the Database. This is a hard problem as fingerprint images can undergo variable amounts of elastic distortion, and feature extraction may introduce noise like missing/spurious minutiae.

Matching of fingerprint images have attracted attention of researchers for last three decades [15]. The matching techniques for fingerprint authentication can be classified into three categories: (i) image-based; (ii) feature-based; and (iii) combination of the two. The image-based techniques include both optical as well as computer-based image correlation techniques. Recently, several transform-based techniques have also been explored. The feature-based techniques extract significant landmarks from a fingerprint image and these feature sets are matched to arrive at a decision. Hamamoto [7] describes a Gabor filter based matching technique. In the feature-based matching category, there are several techniques. The minutiae features $M_i = (x_i, y_i, \theta_i)$ is a set of points and techniques for registering two point sets have been used in [13]. As an alternate representation, Jain *et al.* [9] use a string matching technique. Isenor and Zaky [8] propose a graph-based fingerprint matching algorithm. A graph is constructed by nodes that represent a fingerprint ridge and edges are the neighboring or splitting ridges. A three step algorithm is employed to find a match between a pair of fingerprints. Fan *et al.* [3] describes a fingerprint verification algorithm based on a bipartite graph construction between model and query fingerprint feature clusters. The fingerprint minutiae features are clustered into several close clusters and 24 attributes of the clusters are used in a fuzzy representation of the fingerprint. Germain *et al.*[5] describe an efficient technique for indexing into large fingerprint databases. Recently Jain *et al.* [10] describe a combined matching algorithm that uses the Gabor filter-based and point based matching technique.

Graphs provide powerful representation techniques in many areas of computer vision including object recogni-

tion. Hence research in graph matching techniques with special tuning for vision problems have received wide attention from computer vision researchers. Eshera and Fu [1] describe an image understanding system using attributed symbolic representation and inexact graph matching. Gold and Rangarajan [6] describe a graph matching algorithm based on graduated assignment by posing the graph matching problem as a nonlinear optimization problem. Messmer and Bunke [11] describe a new algorithm for error tolerant subgraph isomorphism detection. Their algorithm is based on combining the model database graphs to a common compact representation.

In this paper, we present a novel graph-based representation of a fingerprint called Minutiae Adjacency Graph (MAG) and describe a robust and accurate matching technique for MAGs based on local structural similarity. Graphs provide powerful representations in many areas of computer vision including object recognition. Hence research in graph matching with special tuning to vision problems have received very wide attention from computer vision researchers. Of particular interest to us are subgraph isomorphism and inexact graph matching techniques as the fingerprint authentication problem can be cast as an inexact graph matching problem. By representing fingerprint features as graphs, we get the robustness of graph representation. Using the fingerprint minutiae features, a labeled, dynamic neighborhood and weighted graph is constructed from the fingerprints. The algorithm allows for rotation, translation, partial overlap and limited elastic distortion of the images. The rest of the paper is organized as follows. Section 2 describes the definitions and notations along with the representation of a fingerprint feature set. Our proposed algorithm is described in Section 3. The algorithm was tested on a large livescan fingerprint database. The results are presented and analyzed in Section 4. Section 5 provides the summary and conclusions.

2. Representation and Definitions

Given a gray scale image of a fingerprint as shown in Fig. 2(a), it is assumed that we have been able to extract the minutia features as accurately as possible using techniques described in the literature [14]. The features extracted are the ridge ending and ridge bifurcation points described in [4]. We represent a fingerprint feature set as a Minutiae Adjacency Graph (MAG) with the minutiae as the nodes, and straight lines connecting two minutiae satisfying a neighborhood criterion as the edges of the graph. It can be observed that

1. In two matching fingerprints, the local neighborhood of two matching nodes look similar.
2. The geometry of minutiae around large neighborhoods of two matching nodes may look dissimilar because of noise, i.e., missing and spurious minutiae.
3. There may not be complete overlap of exposed neighborhoods of two fingerprints.

Because of these reasons, structural information over small distances tend to be more reliable than over larger distances. We consider neighborhoods of small distances to efficiently extract matching nodes.

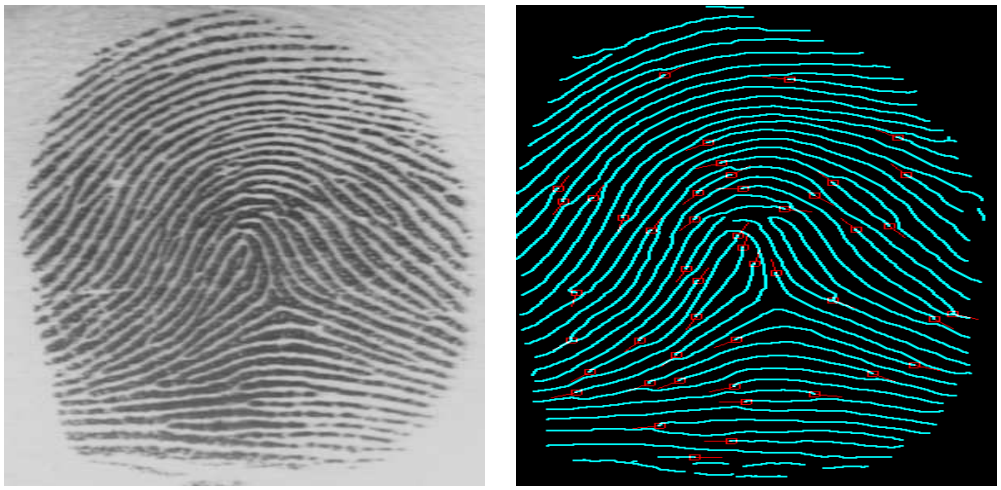
Throughout the discussion of our method, we use the terms node, vertex and minutiae interchangeably. "Matching" or "pairing" two nodes, means pairing a node of one graph with a node of another graph. The proposed algorithm pairs minutiae similar to how a human would go about matching two prints.

- To begin with, we examine the two feature sets to find a minimal set of reliable minutiae that can be paired. This is called as the *strict matching phase*.
- We then relax the criterion of pairing and examine if the pairings can be extended with respect to reliable matches. This is called as the *extension phase*.
- We compute separate costs for both the phases, and combine the costs to arrive at a decision.

Formally, the Minutiae Adjacency Graph (MAG) is represented by $G = (V, E)$, with,

- V the set of vertices representing the set of minutiae.
- E the set of edges.
- A vertex $v \in V$ is represented by a 3-tuple $v = (x, y, \theta)$, where $(v.x, v.y)$ is the x, y coordinate of the corresponding minutiae, and $v.\theta$ is the ridge orientation at the corresponding minutiae.
- An edge $e \in E$ is represented by a 5-tuple $e = (u, v, rad, rc, \phi)$, where $e.u$ is the originating node of the edge, $e.v$ is the destination node of the edge, $e.rad$ is the Euclidean distance between these nodes, $e.rc$ is the ridge count between the two nodes, and $e.\phi$ is the angle subtended by the edge with x-axis.

We use uv to denote an edge between nodes u and v. Note that $(e.rad, e.\phi)$ is the polar coordinate representation of the edge with $e.u$ as origin. The set of nodes, $Nbr(u) = \{x : dist(u, x) \leq d_{max}\}$ is said to be the *neighborhood of* u. The set of edges, $star(u) = \{ux : x \in Nbr(u)\}$ is said to be the *star containing* u. A star is shown in Figure 2(a). Partial MAGs with different d_{max} are shown in Figure 2(b)-(c) to illustrate the neighborhood change as d_{max} increases.



(a) (b)

Figure 1. Fingerprint image and feature extraction. (a) Gray scale image; (b) Features of the image shown in (a).

We now present some of the notations and parameters used in the algorithm. Two nodes of a graph are considered neighbors if the Euclidean distance between them is less than or equal to d_{max}. T_m is the number of matching pairs that *strict matching phase* returns. d_{rel} is the fraction by which two matching edges are allowed to differ in their Euclidean distances. c_{diff} is the maximum allowed difference in the ridge counts of two matching edges. To allow for missing and spurious minutiae, f_{min} is the proportion of the total number of minutiae that are expected to be common to both the prints. N_{min} is the minimum number of neighbors two nodes should have to be considered in *strict matching phase* as we need sufficient evidence in this phase. $C_{m,n}$ is the cost of pairing node m with node n. N_e is the number of matching edges. N_{mat} is the number of matched node pairs. C_{strict} is the cost of strict matching phase. C_{ext} is the cost of the extension phase. $C_{consistency}$ is the cost of consistency checking in strict matching phase. C_{total} is the total cost of matching two fingerprint feature sets. D_α is the maximum orientation difference between two fingerprint images. D_ϕ is the maximum difference of the angle subtended at the minutiae by two consecutive matching edges.

3. Proposed Algorithm

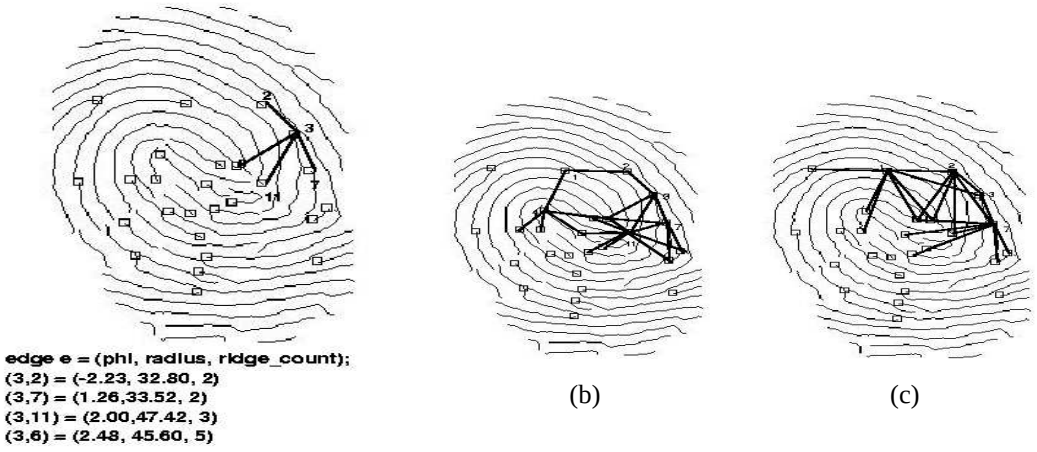
The input to the algorithm are two fingerprints represented by their MAGs, $MAG_1 = (V_1, E_1)$, and $MAG_2 = (V_2, E_2)$. The desired output is a normalized score (say, between 0 and 100) indicating the degree of similarity between the input images. We make the following observations about minutiae sets of matching fingerprints:

- Consider a matching pair, $u \in V_1$, and $v \in V_2$ belonging to parts in fingerprints not affected by noise. We observe that stars around u and v look very similar with respect to distance, angle subtended, ridge counts and ridge orientation. The *strict matching phase* aims to obtain a minimum number of such good matches by comparing stars of all possible pairs.
- If (u_1, v_1) and (u_2, v_2) are two correct matches, then the edge (u_1u_2) should be similar to (v_1v_2). We check this consistency in the *consistency check*.
- We can extend existing matches of a pair (u, v) by comparing the distance and ridge count of u, and v from the corresponding matching nodes obtained in *strict matching phase*. This is done in the *extension phase*.

We allow for elastic distortion and noise by using tolerances, and develop cost functions that measure similarity with respect to distances and angles. In subsequent discussions, for all tuples of the form (u_i, v_i), we imply $u_i \in V_1$ and $v_i \in V_2$.

3.1. Strict matching phase

In this phase, we pair minutiae by comparing them with respect to edge distances, ridge counts and inter-edge angles. That is, we compute the cost of pairing a node m



(a) (b) (c)

Figure 2. MAG definitions.(a) components of MAG; (b) a partial MAG with d_{max} =80; (c) a partial MAG with d_{max}=100.

with node n, $C_{m,n}$, such that $m \in V_1$, and $n \in V_2$, by matching the stars around them. Let S_1 and S_2 be the neighborhood of m and n, respectively. (m, n) is considered for a match only if $min(Size(S_1), Size(S_2)) \geq N_{min}$. The star matching algorithm returns (a) a cost of matching the stars, and (b) a subset $A_1 \subseteq S_1$, and a subset $A_2 \subseteq S_2$ such that $Size(A_1) = Size(A_2) = msize \geq 0.5 * min(Size(A_1), Size(A_2))$. Suppose $A_1 = \{u_1, u_2, \ldots, u_{msize}\}$ and $A_2 = \{v_1, v_2, \ldots, v_{msize}\}$ then,

- u_i matches v_i in the stars around m and n, respectively.
- The order of nodes visited clockwise from u_1 is $u_1, u_2, \ldots, u_{msize}$ and, similarly, for $v_1, v_2, \ldots, v_{msize}$. We call nodes (u_i, u_{i+1}) *consecutive matching nodes*. The corresponding edges with m are called *consecutive matching edges*.

Two edges $e1$, and $e2$ are said to be matching or matchable if $(|e1.rad - e2.rad|/min(e1.rad, e2.rad) \leq d_{rel}) \wedge (|e1.rc - e2.rc| \leq c_{diff})$. If $(e1, e2)$ are *consecutive matching edges* and $(f1, f2)$ are *consecutive matching edges*, and $e1$ matches $f1$ matches then the angles subtended by $e1, \widehat{m}, e2$, and $f1, \widehat{n}, f2$ cannot differ by more than D_ϕ. The orientation of two matching stars with respect to two matching edges cannot differ by more than D_α. To allow for complete rotation, we can set $D_\alpha = 2 * \pi$.

The cost of matching two edges $e1$, $e2$, *radial cost*, can be described by

$$d_r = (|e1.rad - e2.rad|/(min(e1.rad, e2.rad) * d_{rel})). \quad (1)$$

Consider matching *consecutive matching edges* $(e1, e2)$ with $(f1, f2)$, and let θ_1 be the angle between $e1, \widehat{m}, e2$, θ_2 be the angle between $f1, \widehat{n}, f2$, then the *angular cost* is

$$d_\phi = \frac{|\theta_1 - \theta_2|}{D_\phi}. \quad (2)$$

The cost of matching two nodes m and n

$$C_{m,n} = \frac{\sum d_r + \sum d_\phi}{N_e^2} \quad (3)$$

The sum is over all matched edges.

We outline our procedure to match two stars in presence of rotation, noise and elastic distortion. An edge pair of two stars is said to be starting edge pair if the two edges are matchable. For each starting edge pair, we can compute the best match by making one clockwise traversal of both the stars. Thus if the stars have x, and y neighbors, then there are $O(xy)$ starting edge pairs, and $O(x)$ time is required to compute best match for each such pair. The *strict matching phase* computes cost for all possible pairs, and returns a set TOP of best T_m distinct pairs of matches. The cost of the *strict matching phase* is

$$C_{strict} = \frac{\sum_{(u_i, v_i) \in TOP} C_{u_i, v_i}}{T_m}. \quad (4)$$

We check consistency of the matched pairs of TOP in the *consistency check phase*. We consider the set of matches, TOP, and construct cliques Q_1, and Q_2 of the nodes $\{u_i : i = 1, 2, \ldots, T_m\}$, and $\{v_i : i = 1, 2, \ldots, T_m\}$, respectively. To allow for presence of noise, we do not insist on a complete match of cliques but a minimum of $N_m = f_{min} * T_m$ matches to be consistent. So a pair $(u_i, v_i) \in TOP$ is consistent if at least N_m of $\{(u_i u_j, v_i v_j) : j = 1, 2, \ldots, T_m\}$ are matched by distance and ridge count. The set TOP is said to be consistent if there are at least N_m consistent matching pairs. The cost of a consistent matching pair $c_{cons-pair}$ is the average of

radial costs of all matching edges. The cost of consistency, $C_{consistency}$, is the sum of matching costs of all matching pairs divided by the number of matching pairs, N

$$C_{consistency} = \sum \frac{c_{cons-pair}(u_i, v_i)}{N}. \quad (5)$$

3.2. Extension phase

This phase extends the match based on evidence collected with respect to TOP, the set of matching pairs returned by *strict matching phase*. In this phase, we consider the set of unmatched pairs $\{(u, v) : u \neq u_i \wedge v \neq v_i, i, = 1, 2, \ldots, T_m\}$. We compare the edges $\{((u, u_i), (v, v_i)), i = 1, 2, \ldots, T_m \ and \ (u_i, v_i) \in TOP\}$. If more than f_{min} fraction of these edge pairs match, then (u, v) is considered a possible match with cost

$$c_{ext}(u, v) = \sum \frac{d_r}{N_e} \quad (6)$$

We consider all the pairs in increasing order of their cost and get all possible distinct extendable pairs. In cases when a node has more than one extendable node, we choose the pair with the least cost. Let $N_{ext} = f_{min} * min(Size(V_1), Size(V_2))$ be the expected number of matching nodes considering missing and spurious minutiae. Let N_{mat} be the set of pairs of nodes matched in the *extension phase*. The extension cost denoted by C_{ext} is given by

$extsum = \sum(c_{ext}(u, v) :, (u, v) \in N_{mat});$

$$C_{ext} = \frac{extsum * N_{ext}}{(Size(N_{mat}))^2} \quad (7)$$

3.3. Resolution phase

If the two fingerprints being matched have very few minutiae in common, then the default neighborhood size may result in small stars. Secondly, in the case of non-matching fingerprints, the default set of parameters may be sufficient to give rise to some uncorrelated matches. In both cases, the scores reported are neither low enough to indicate a mismatch, nor high enough to indicate a match. We call such pairs *possible matching* pairs. We deal with such cases by recomputing costs with stricter tolerances to eliminate random matches and quickly get evidence for a mismatch. If even in presence of strict tolerances we are not able to decide, it may be because we have very small stars. So we increase neighborhood distance, and recompute the cost. This helps to get more evidence in case of matching fingerprints.

3.4. Parameters and Decision making

We observe that choosing the d_{max} to be a value of 100 to 125 units results in a neighborhood of size 8-12 for most nodes. The other parameter values in our experiments are: $MM = 0.06$; $EL = 0.2$; $CL = 0.2$; $LS = 0.04$; T_m = 8; d_{rel} = 0.15; f_{min} = 0.7; c_{diff} = 3; D_α = 0.75; C_∞ = 100000; N_{min} = 5; and D_ϕ = 0.1.

The combined cost for a pair of graphs is given by $(100 - CONST * C_{strict} * C_{consistency} * C_{ext})$ where $CONST$ is such that, the score is between 0 and 100. We use an empirical threshold to arrive at a YES/NO decision.

4. Performance Evaluation

There are two types of errors in an authentication system: the False Accept Rate (FAR), and False Reject Rate (FRR) are important measures of accuracy of a matcher. The receiver Operating Curve (ROC) [2, 12] is a graph that expresses the relation between FRR and FAR when the matching threshold T is varied. We report the system accuracy using ROCs. It can be easily shown that, for practical choices of neighborhood size, our algorithm takes $O(kmn)$ time for graphs with m, and n nodes where k is the time required to match stars having 8-12 size which is a constant.

We extensively tested our algorithm on two databases of livescan images obtained by optical scanners. Optical image set1, has 50 persons, four impressions per person, thus it has 300 matching pairs, and 19600 mismatches. Optical image set2, has 150 persons, four impressions per finger, and four fingers per person. The ROCs are plotted with FRR on the x-axis, and FAR on the y-axis. An ideal matcher's ROC overlaps with y-axis. The ROCs for the datasets are shown in Figure 3(a)-(b). Note that our ROCs are very close to y-axis. On the first data set, we also have the results plotted using a commercially available matcher.

5. Conclusions

In this paper, we have presented a graph-based representation for fingerprints, a heuristic matching algorithm which allows for anomalies like missing/spurious minutiae, elastic distortion, rotation and translation of the input prints. Our algorithm is based on simple and intuitive cost functions. Its robustness is substantiated by experimental results on large databases. We are considering various parallel implementations to speedup the algorithm. Techniques to sample the minutiae set without performance degradation during matching are being evaluated.

References

[1] M. A. Eshera and K-S. Fu. An image understanding system using attributed symbolic representation and inexact graph-matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-8(5):604–618, September 1986.

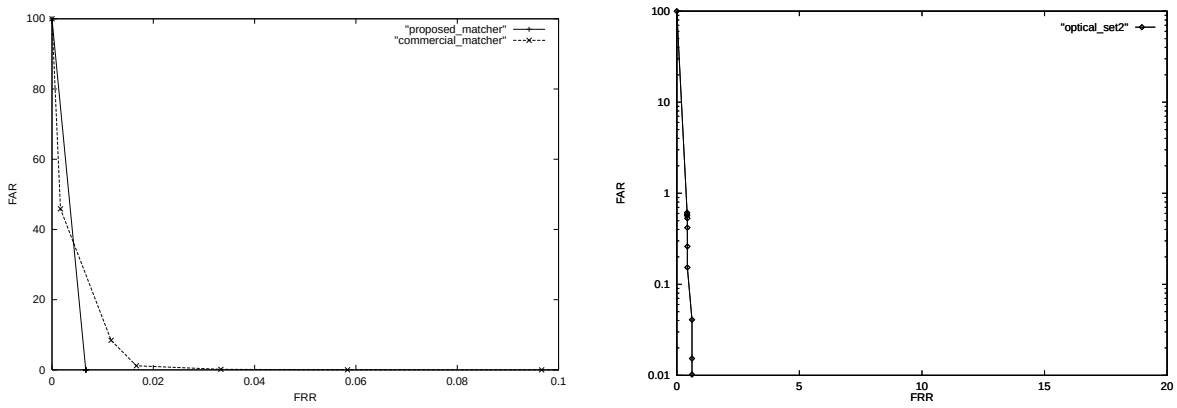


Figure 3. Error rate curves. (a) Optical image set1; (b) Optical image set2.

[2] B. Germain et al. Issues in large scale automatic biometric identification. In *In Proc. IEEE Workshop on Automatic Identification Advanced Technologies*, volume Stony Brook, NY, Nov 1996.

[3] K.-C. Fan, C.-W. Liu, and Y.-K. Wang. A fuzzy bipartite weighted graph matching approach to fingerprint verification. In *In Proc. of the IEEE Intnl. Conf. on Systems, man and cybernetics*, pages 729–733, Oct 1998.

[4] Federal Bureau of Investigation, U. S. Government Printing Office, Washington, D. C. *The Science of Fingerprints: Classification and Uses*, 1984.

[5] R. S. Germain, A. Califano, and S. Colville. Fingerprint matching using transformation parameter clustering. *IEEE Computational Science and Engineering*, pages 42–49, Oct-Dec 1997.

[6] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(4):377–388, April 1996.

[7] Y. Hamamoto. A Gabor filter-based method for identification. In L. C. Jain, U. Halici, I. Hayishi, S. B. Lee, and S. Tsutsui, editors, *Intelligent biometric techniques in Fingerprint and face recognition*, pages 137–151. CRC Press, Boca Raton, 1999.

[8] D. K. Isenor and S. G. Zaky. Fingerprint identification using graph matching. *Pattern Recognition*, 19(2):113–122, 1986.

[9] A. K. Jain, L. Hong, and R. M. Bolle. On-line fingerprint verification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(4):302–313, April 1997.

[10] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti. Filterbank-based fingerprint matching. *IEEE Trans. on Image Processing*, 9(5):846–859, May 2000.

[11] B. T. Messmer and H. Bunke. A new algorithm for error-tolerant subgrpah isomorphism detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(5):493–503, May 1998.

[12] W. Peterson, T. Birdsall, and W. Fox. The theory of sigmal detectability. *Transactions of the IRE*, PGIT-4:171–212, April 1954.

[13] N. K. Ratha, K. Karu, S. Chen, and A. K. Jain. A real-time matching system for large fingerprint database. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(8):799–813, Aug 1996.

[14] Nalini K. Ratha, Shaoyun Chen, and Anil K. Jain. Adaptive flow orientation based texture extraction in finger print images. *Pattern Recognition*, 28(11):1657–1672, November 1995.

[15] J. H. Wegstein and J. F. Rafferty. Matching fingerprints by computer. Technical Report Technical note 466, National Bureau of Standards, 1969.