Critique on Fingerprint Identification Using Graph Matching

Overview of the Problem

Biometric characteristics such as face, fingerprint and iris provide a unique natural signature of a person. Fingerprints have been used as a biometric characteristic for a long time because fingerprints offer unique advantages over other biometrics in terms of acquisition ease, relative temporal invariance, and uniqueness among different subjects. Fingerprint images are matched based on certain features which are unique for every individual. These features are extracted using feature extraction algorithm and then matched using corresponding matching algorithm. While capturing the various instances of a fingerprint image, the image may be subjected to different transformations such as displacement, rotation, stretching or compression and the presence of noise. These differences lead to inaccuracies during feature extraction and matching. The various features of fingerprint which are captured for recognition are minutiae, ridges, singular points (core and delta), texture, pores and 3D features. Based on the feature sets, fingerprint matching algorithms can be characterized into pattern matching, minutiae matching, level 3 features based matching and 3D fingerprint matching techniques.

Algorithm Review

In the given paper [1], authors have proposed a fingerprint identification algorithm to solve the problem of orientation, stretching, and certain types of noise. In this work, features of the fingerprint are encoded in the form of a graph and graph/tree matching algorithm is used to match the two graphs. Graph based algorithm can be considered as a tool used for matching two feature sets such as pattern or minutiae matching. Representing the features in the form of graph gives an edge over other techniques in case of large database handling and feature matching. The algorithm [1] is divided into two sections: encoding and matching.

In the presented algorithm a fingerprint is represented in the form of a graph with nodes representing the ridges and edges representing the adjacency information. This encoding scheme is based on the idea that minutiae locations can be encoded by describing the topological relations of fingerprint ridges. The information for ridges in the graph is recorded as part of the description of the corresponding nodes in the graph. The recorded information are ridge length, nature of minutiae at each end of the ridge, whether the ridge is complete or not, clockwise order of the ridge neighbors and the overlap length for each neighbor. These parameters are used to represent the fingerprint in the form of a directed graph. To compute all these parameters, firstly a ridge thinning algorithm (modified version of Deutsch's Algorithm) is used to reduce all the ridges to a thickness of one pixel. All the ridges are assigned proper level numbers and the ridge joining, ridge ending and the approximate length of the ridges are calculated. To capture the ridge adjacency information, the neighbors (end and side) are calculated which describe the topology in the vicinity of any ridge and are further used to calculate the overlap length.

A post processing step is introduced to handle noise and special features. In this process the ridge joins which can be due to dirt or ink and appear same as a crossover is also repaired. Ridge breaks can be recognized in this graph representation more effectively. Some other special features are also considered as a graph substructure which can be further used for matching.

Matching is divided into three steps: Partitioning, Refinement and Scoring. In partitioning, a partitioning array is created whose size depends on the number of nodes in the two graphs to be matched. The elements in the array have values true or false depending on the relation in the corresponding nodes in the two graphs. This relation is determined by length of the ridges and the number of neighbors on corresponding sides of the ridges. Refinement is done to reduce the number of nodes in the partition array. The mapping between two nodes is retained only if the side neighbors of one node maps onto the corresponding side neighbors of another node. This mapping is checked for both sides of the ridge separately and if any of the sides matches then the mapping is retained. This is done because ideally there should be only one node in each partition for a match and none for a mismatch. The refinement step is followed by scoring which chooses one node from each partition such that the matching score is maximum and no node is used twice. The basic fundamental of matching these graphs is from the premises of dynamic programming approach of tree matching. This calculated matching score is the measure which determines the degree of match of the two fingerprints.

The theoretical complexity of the encoding step is $\underline{O}(P^2)$, where the size of the fingerprint image is PxP, and matching step is $O(N^3)$ where N is the number of ridges in the fingerprint. In this paper the ridge features lead to construction of N^2 trees each having N nodes. But in practical problems, authors have claimed that the complexity of the refinement and scoring is expected to be considerably lower and the complexity of matching step is close to $O(N^2)$.

Authors have tested the algorithms on a scanned fingerprint database taken from RCMP Headquarters, Ottawa. The images are of 1536 X 1536 pixels with 8 bits of gray level information. Some overlapping samples of fingerprint images of size 512 X 512 have been extracted from the original fingerprint and are used for testing. But authors have stated that this database has a very small overlap between two images unless they are same. Authors have claimed that the tree building algorithm in the scoring stage is able to correctly identify matching minutiae but a well defined threshold for comparison could not be achieved. Though authors have not claimed any accuracy but after careful investigation of the provided matching score table, it can be concluded that the algorithm is not able to generate a clear separation for genuine user and imposter. The proposed algorithm takes an average of 16.2 seconds for encoding and 90 seconds for matching on a VAX-780 computer.

Critique

There are several advantages to the algorithm proposed in [1]. It takes into account the presence of noise by breaking the ridge joins and considers even the rotation and translation of the fingerprint by forming a graph of the features considering the clockwise and counter wise direction. The stretching is taken care of as the scoring does not depend on the exact length of the ridges. This algorithm extracts the features efficiently from the scanned inked fingerprints and it should work for the digital images obtained from a fingerprint scanners available nowadays. This paper considers the assertion that minutia locations can be encoded by describing the topological relationships of fingerprint ridges. So it indirectly considers the minutiae features along with the ridge patterns.

Authors have mentioned the unavailability of a proper database. The approach of generating multiple instances from the same fingerprint is not very convincing. They might have prepared a detailed and proper database with multiple instances of the same fingerprint to validate the algorithm. Authors have even mentioned that the images present in the database mostly contain less number of minutia which further leads to lesser number of corresponding matching minutiae. The database images may be a reason for not getting a clear separation for the threshold. Results on a more detailed and proper database would give better understanding of the performance.

The problem with the algorithm is that it does not consider the preprocessing for image enhancement. If the image enhancement and preprocessing are done properly to reduce the noise, then the crossovers can be considered as a unique feature of a fingerprint (instead of breaking them) to increase the accuracy. The algorithm does not consider the boundary ridge which leads to decrease in the matching score. The scoring algorithm adopted is not good enough to give a clear separation between inter-class and intra-class variability.

Suggested Modifications

There are several possible ways in which the algorithm can be further improved to get better results. Incorporating preprocessing steps for noise removal and image enhancement will enhance the performance of the algorithm. There are several other modifications possible to the algorithm. One of them can be, improving the scoring technique to obtain a clear separation. By testing the algorithm on a detailed database, the modifications to improve the scoring can be decided. Graph matching algorithm can be modified to properly match the ridge features. In [6] a graph matching algorithm for large graph is presented which is able to efficiently solve the graph isomorphism and graph-subgraph isomorphism problems on attributed relational graphs. The same fundamental basis can be used here to properly match the graph generated from fingerprint images. In [4] a modified graph matching algorithm is proposed for fingerprint classification; this approach can also be considered as the possible modification for matching.

Another modification to the algorithm could be encoding minutiae features instead of ridges and apply a greedy algorithm. Using the minutiae will reduce the complexity of the algorithm. An algorithm has been presented by Ratha et. al. in [2] which considers the minutiae features. Nodes of the graph are represented by the minutiae, and the edges and vertex represent the characteristics of the minutiae like x, y, coordinates and orientation. For matching, first the matching node pairs are selected based on the neighborhood structures. A comparison based on distances between the matching pairs is done to finally find the matching pairs. The algorithm in [2] is claimed to be translation and rotation invariant and allows for partial overlap and limited elastic distortion of the images.

Ridge topology can also be considered along with the minutiae details. This can be useful in cases where there are spurious minutiae and the matching decision based on minutiae can give false interpretations. The ridge topology obtained from a skeletonized fingerprint can be encoded in the graph representation. In the paper by Werner and Walter [3], authors have given the graph representation of the ridge topology. An algorithm based on dual graph contraction is given to derive graph representation automatically from a preprocessed, skeletonized fingerprint image. The comparison has been done manually in the paper; but it has been suggested that the matching algorithm can be automated.

The tree generation algorithm can be modified by using some learning algorithm like SVM, Kernel based or neural networks. In this paper the authors have generated the tree in all the directions. It is a dynamic approach in which all the substructures have been explored and then the optimal tree is selected having the maximum score. This strategy of selecting the optimal among all the possibilities can be modified by learning the growth of the tree. Any intelligent technique can be used to learn the growth and then the tree with the maximum score can be chosen in a greedy manner, thus reducing the time without decreasing the accuracy. This can allow for matching with any print having lesser or greater amount of information present. Use of these intelligent techniques can even reduce the complexity of score generation.

References

1. D. K. Isenor and S. G. Zaky, "Fingerprint Identification using Graph Matching", Pattern Recognition, 1986, vol. 19, no. 2, pp. 113-122.
2. Nalini K. Ratha, Vinayaka D. Pandit, Ruud M. Bolle and Vaibhav Vaish, “Robust Fingerprint Authentication Using Local Structural Similarity”, In Fifth IEEE Workshop on Applications of Computer Vision, 2000, pp. 29-34.
3. Werner Ölz and Walter G. Kropatsch, “Graph Representation of Fingerprint Topology” In Danijel Skocaj, editor, Computer Vision - CVWW'04, Slovenien Pattern Recognition Society, 2004, pp. 51–58.
4. M. Neuhaus and H. Bunke, “An Error-tolerant Approximate Matching Algorithm for Attributed Planar Graphs and its Application to Fingerprint Classification”, In Proc. of the tenth International Workshop on Structural and Syntactic Pattern Recognition, LNCS 3138, 2004, pp. 180-189.
5. Kuo-Chin Fan; Cheng-Wen Liu and Yuan-Kai Wang, “A fuzzy bipartite weighted graph matching approach to fingerprint verification”, In Proc. of the IEEE International Conference Systems, Man, and Cybernetics, 1998, vol.5, pp. 4363-4368.

6. L. P. Cordella, P. Foggia, C. Sansone, M. Vento, "An Improved Algorithm for Matching Large Graphs", In Proc. of the third IAPR-TC-15 International Workshop on Graph-based Representation, 2001.