A Graph Matching Based Approach to Fingerprint Classification Using Directional Variance

Michel Neuhaus and Horst Bunke

Department of Computer Science, University of Bern
Neubrückstrasse 10, CH-3012 Bern, Switzerland
{mneuhaus,bunke}@iam.unibe.ch

Abstract. In the present paper we address the fingerprint classification problem with a structural pattern recognition approach. Our main contribution is the definition of modified directional variance in orientation vector fields. The new directional variance allows us to extract regions from fingerprints that are relevant for the classification in the Henry scheme. After processing the regions of interest, the resulting structures are converted into attributed graphs. The classification is finally performed with an efficient graph edit distance algorithm. The performance of the proposed classification method is evaluated on the NIST-4 database of fingerprints.

1 Introduction

Fingerprint classification refers to the process of assigning fingerprints in a consistent and reliable way to classes. The main objective is to reduce the complexity of the general fingerprint identification problem, where a fingerprint is to be matched against large databases of fingerprints. The fingerprint classification problem is considered to be difficult because of the large within-class variability and the small between-class separation. For many years, classification methods from various pattern recognition areas have been proposed, commonly divided into rule-based, syntactic, statistical, and neural network based approaches [1, 2]. Although the classification problem is intrinsically of structural nature, it was not until recently that classification systems based on structural pattern recognition methods have been developed [3–5]. In comparison to state-of-the-art classification methods, structural approaches often fall behind in terms of performance. Yet, in the context of multiple classifier combination, structural algorithms have proven effective in improving existing classification methods [5, 6]. We furthermore believe that the strength of structural algorithms has not yet been fully exploited in fingerprint recognition.

In fingerprint identification or verification, where identical fingerprints are to be matched, one usually focuses on local characteristics, such as minutiae points. Conversely, in fingerprint classification, the problem is often addressed by extracting and representing global characteristics, such as the ridge flow or

T. Kanade, A. Jain, and N.K. Ratha (Eds.): AVBPA 2005, LNCS 3546, pp. 191–200, 2005.

singular points [1, 2]. In the present paper, we propose an image filter based on a new definition of directional variance. Following the Galton-Henry classification scheme of five classes, we use the filter to extract regions that are relevant for the classification. Our second contribution consists in applying edit distance based graph matching to the classification problem after extracting the characteristic regions.

In Section 2, the directional variance filter on orientation vector fields is described. A brief review of error-tolerant graph matching follows in Section 3. Section 4 gives a number of experimental results, and some concluding remarks are provided in Section 5.

2 A Directional Variance Algorithm

The key procedure of a large number of fingerprint classification algorithms is based on the robust detection of singular points of the ridge orientation vector field [2]. To assign fingerprints to one of the five classic Henry classes, it is in most cases sufficient to know the number and position of singular points [7, 8]. In this paper, we propose an algorithm for the reliable computation of a directional variance value measured at every position of the ridge orientation field. The variance is defined such that high variance regions correspond to relevant regions for the fingerprint classification task, including singular points.

Weakly related to the statistical variance, we define the directional variance of the ridge orientation field at position (x, y) by

$$\sigma^2_{x,y} = \frac{1}{1-n} \sum_{i,j} \sin^2(\alpha_{i,j} - \bar{\alpha}_{x,y}) \ , \tag{1}$$

where $\alpha_{i,j}$ denotes the vector at position (i, j) of the vector field and the summation is performed over a window of size n around (x, y). The average orientation $\bar{\alpha}_{x,y}$ of the local window around position (x, y) is computed by taking into account that two vectors pointing in opposite directions represent the same orientation [9, 10]. The circular nature of the orientation vectors is also accounted for in the sine term; vectors $\alpha_{i,j}$ that are orthogonal to the local average $\bar{\alpha}_{x,y}$ contribute maximally and vectors close to the local average contribute minimally to the variance.

From Eq. 1 it follows that the directional variance is expected to be low everywhere in smooth orientation fields. But in the local neighborhood of singular points, orientations do not follow a single predominant direction, which is equivalent to a high directional variance. In experiments we could confirm this behavior.

Our objective in this paper is not to detect singular points, but rather to extract regions that allow us to discriminate between fingerprint classes. For this purpose, we propose to use a modified directional variance measure, which differs from the directional variance in Eq. 1 in the computation of the local average orientation $\bar{\alpha}_{x,y}$. In a first step, all orientations are normalized to an angle range

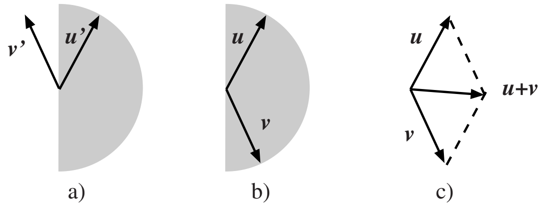

Fig. 1. a) Two vectors representing two ridge orientations, b) the corresponding normalized vectors, and c) their vector sum

of $I = [-\pi/2, \pi/2]$, which corresponds to a vector range of $I' = R_0^+ \times R$. The normalization of two orientation vectors and the normalization range I (see the shaded area) is illustrated in Fig. 1a,b. Normalization consists in reversing any vector that is located outside the shaded area. We proceed by defining the average direction of a number of normalized orientation vectors by their vector sum. In Fig. 1c, the sum of two normalized vectors is illustrated. For a set of vectors in horizontal direction, the vector sum will clearly point in a horizontal direction as well. For a set of vectors in vertical direction, however, the vector sum will not point in vertical direction, but be close to the horizontal direction, as some vectors will point upwards and some will point downwards due to the normalization procedure. In this case, the mean direction $\bar{\alpha}_{x,y}$ does not correspond to the local orientations, which results in a high directional variance. Hence in addition to singular points, the modified directional variance is also responsive to vertical orientation regions. In other words, the proposed new directional variance can be used as a filter that will emphasize not only singular points, but also areas with vertical ridge orientation.

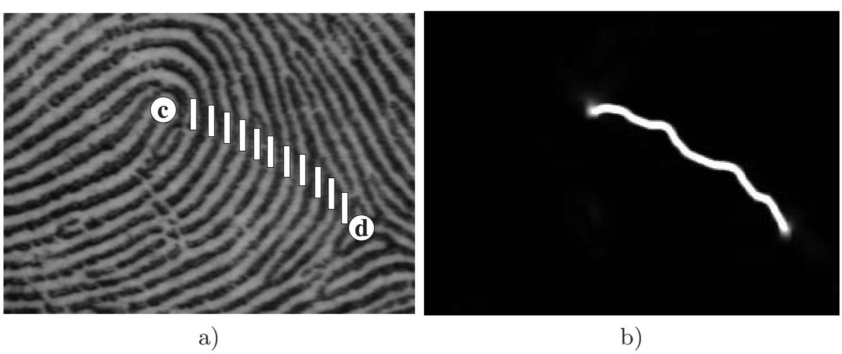

Fig. 2. Left loop fingerprint image a) with core point (c), delta point (d), and marked vertical orientations and b) visualization of the modified directional variance

A closer examination reveals that different fingerprint classes exhibit different characteristics of singular points and vertical orientation regions. Arch fingerprints, for instance, contain no singular points and no vertical ridges, except

a) b) c)

Fig. 3. a) Original fingerprint image, b) visualization of the modified directional variance (bright colors indicate high variance), c) binarized image

for strongly rotated fingerprints. Loop fingerprints, on the other hand, are characterized by a global ridge loop, a core point, and a delta point [7]. The key observation is that one can reach the core point from the delta point via locally almost vertical ridge segments, which is due to the nature of the ridge flow around the delta point and the core point. An illustration of this observation is provided in Fig. 2, where the vertical orientation segments are clearly visible in the loop fingerprint image and in the image resulting from applying the directional variance filter. The same properties are also present in right loop, whorl, and tented arch fingerprints. In a number of experiments, it turns out that the directional variance filter detects the connection between core and delta point much more reliably than a filter simply enhancing vertical orientations. In contrast to other classification methods, the directional variance approach does not solely rely on the detection of singular points, but can also be employed if singular points are not present in the image or distorted by noise.

After filtering the fingerprint, the resulting image is binarized and undergoes a noise removal procedure. The extracted regions can then be used for the purpose of classification. Possible classification criteria include the number of extracted regions and the position and main direction of the regions. An illustration of the extraction of the characteristic regions in a whorl image is shown in Fig. 3. It is easy to verify that the ending points of the two extracted regions correspond to the four singular points, and the regions to the vertical orientation areas of the ridge orientation field. Further examples from the *left loop*, *right loop*, and *whorl* class are shown in Fig. 4. To perform the actual classification based on the extracted regions, various classifiers could potentially be employed. One such method based on graph matching is described in the following sections.

3 Error-Tolerant Graph Matching

Graph matching refers to the process of evaluating the structural similarity of attributed graphs, that is, the similarity with respect to nodes, edges and attributes attached to nodes and edges. A large number of graph matching methods from various research fields have been proposed in recent years [11], ranging from isomorphism-based systems to algorithms based on the spectral decomposition of graph matrices and the definition of positive definite kernel functions on graphs.

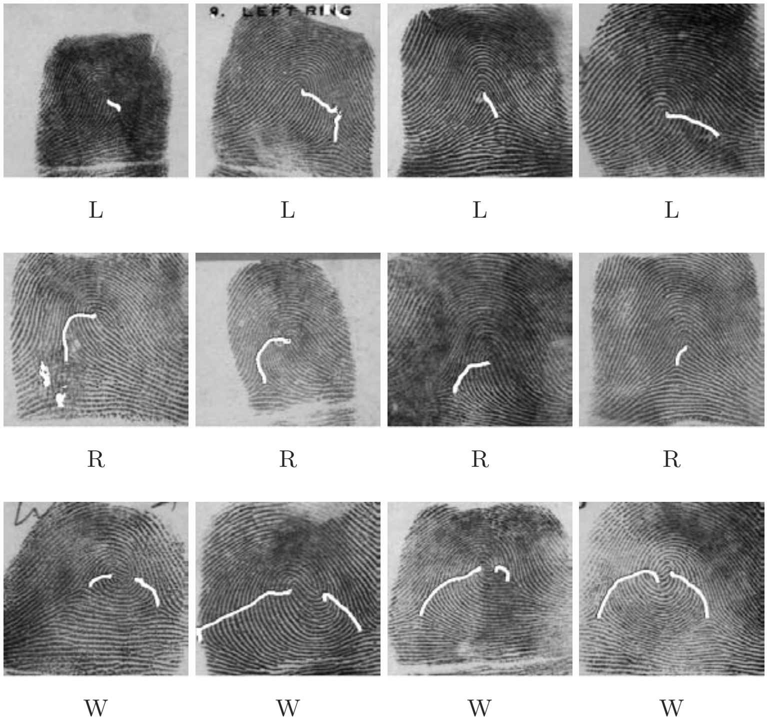

Fig. 4. Visualization of the modified directional variance for *left loop* (L), *right loop* (R), and *whorl* (W) fingerprints

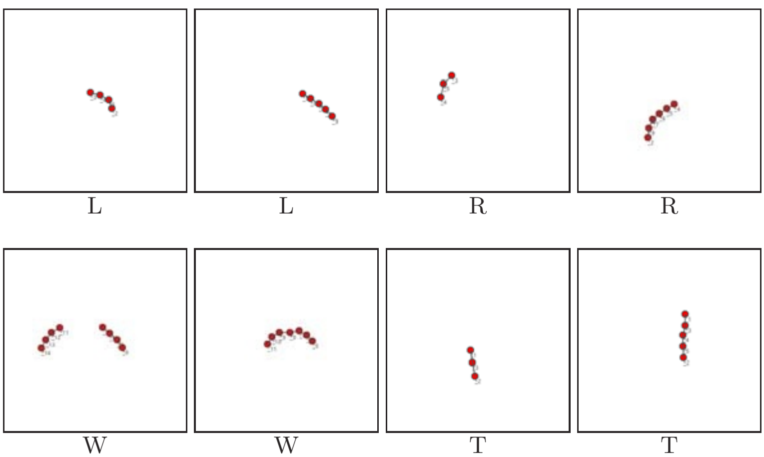

Fig. 5. Sample prototype graphs for the *left loop* (L), *right loop* (R), *whorl* (W), and *tented arch* (T) class

3.1 Graph Edit Distance

One of the most intuitive error-tolerant graph matching approaches consists in the computation of the graph edit distance [12, 13]. The graph edit distance is defined in the context of a basic distortion model, where structural distortions are performed by edit operations on graphs. The standard set of edit operations comprises a node insertion, a node deletion, a node substitution, an edge insertion, an edge deletion, and an edge substitution operation. A sequence of edit operations $(e_1, \ldots, e_l)$ transforming graph g into graph g' is termed an edit path from g to g', and $E(g, g')$ denotes the set of edit paths from g to g'. Given a cost function $c : E(g, g') \to R^+ \cup \{0\}$ assigning non-negative costs to edit paths, we can then define the edit distance of g and g' by

$$d(g, g') = \min_{p \in E(g,g')} c(p) \ . \tag{2}$$

The edit distance of two graphs is thus given by the least expensive transformation of the first graph into the second graph in the underlying distortion model. The cost function is usually defined on single edit operations with respect to the attributes attached to nodes and edges.

An edit distance based system can be tailored to a specific application by adjusting the cost functions accordingly. The basic idea is that weak distortions should result in low costs, whereas strong distortions should correspond to higher costs. The cost functions implicitly define, for instance, when the removal of a node n followed by the insertion of another node n' is less expensive than the substitution of n with n', and therefore preferred in an optimal edit path. In other words, the edit distance is derived from the most reasonable explanation of the structural differences of two graphs in the edit operation framework.

The actual computation of the edit distance is performed by constructing and traversing a search tree. In spite of pruning criteria and look-ahead techniques, however, the computational complexity both in terms of running time and memory requirements is high – in fact, it is exponential in the number of nodes of both graphs. For unconstrained graphs of arbitrary size, the edit distance approach is largely unfeasible. Therefore, a fast approximate version of the edit distance algorithm for large graphs is employed in the experiments of this paper. A brief description of the algorithm follows in the next section.

3.2 Approximate Graph Edit Distance

The development of efficient graph matching algorithms for special classes of graphs, for instance bounded-valence graphs, trees, or planar graphs, has been an issue in the graph matching literature for years [11]. In the graph edit distance context an efficient approximate algorithm has recently been proposed that turns out to be very fast and sufficiently accurate for certain graph problems [14]. This approximate algorithm requires the graph nodes to be embedded in the plane. That is, for every node a meaningful position attribute providing a spatial context needs to be present. For graphs extracted from images it is usually easy to derive such a node embedding. Examples include graphs representing interest points and their relations, or region adjacency graphs.

Instead of exploring the full search space, only a subset of all edit paths is considered in the approximate algorithm. Starting from an initial node substitution $n \to n'$, the least costly transformation from the neighborhood of n to the neighborhood of n' is computed by optimizing local minimum-cost criteria. The computation is performed by means of an efficient cyclic string matching algorithm based on dynamic programming. The result is a valid edit path between two graphs, but not necessarily the optimal one. To account for the dependence on the initialization, the computation is carried out for a number of initial substitutions, and the minimum cost edit path among them is kept. In contrast to the exponential computational complexity of the exact edit distance, the approximate algorithm runs in polynomial time. In practical experiments, the approximation has shown to be feasible and fast, even for large graphs with more than 200 nodes and edges, whereas the exact edit distance algorithm can only be computed for graphs with a size of about 10 nodes [14]. In the following, the approximate edit distance will be used to obtain distance values between fingerprint graphs and subsequently perform the classification.

3.3 Fingerprint Graph Representation and Edit Cost Function

From the results of the region extraction process based on the modified directional variance filter described in Section 2, an attributed graph can be extracted in various ways. In this paper, we follow a simple method to generate structural skeletons. We proceed by applying a one-pass thinning operator [15] to the extracted regions and represent ending points and bifurcation points of the resulting skeleton by graph nodes. Additional nodes are inserted along the skeleton at regular intervals. An attribute giving the position of the corresponding pixel is attached to each node. Edges containing an angle attribute are used to connect nodes that are directly connected through a ridge in the skeleton. An illustration of several graphs of this kind is given in Fig. 5. The simple edit cost function we employ assigns constant costs to insertions and deletions independent of involved attributes; substitution costs are defined proportional to the Euclidean distance of attributes.

4 Experimental Results

The NIST-4 database [16] consists of 4,000 grayscale images of fingerprints with class labels according to the five most common classes of the Galton-Henry classification scheme: *arch*, *tented arch*, *left loop*, *right loop*, and *whorl*. We proceed by extracting an attributed graph from every image as described previously to obtain 4,000 graphs. On the average, these graphs contain 6.1 nodes and 10.3 edges. To classify fingerprint graphs by means of the edit distance, a set of reference graphs for each class needs to be defined. Although an automatic method would be desirable, it proved efficient in recent studies [17, 18] to use a manual construction procedure for this purpose. Adopting a similar approach, we define prototype graphs by manually selecting promising candidates from a training set of graphs. Where appropriate, a few nodes are deleted from prototype candidates to provide for class representatives as general as possible. By means of this procedure we obtain about 60 prototypes overall. The classification can then be performed based on the nearest-neighbor paradigm: An input graph is assigned the class of the most similar prototype graph. The structural similarity is derived from the corresponding approximate graph edit distance between prototype graph and input graph. An illustration of some prototype graphs is provided in Fig. 5.

The first 1,000 fingerprints from the database are used for the development of the class prototypes and are therefore considered a *Training set*. The remaining 3,000 fingerprints constitute the independent *Test set 1*, and the subset of *Test set 1* consisting of the last 2,000 fingerprints of the database is termed *Test set 2*. The classification rates obtained on the various data sets are summarized in Table 1, where GED refers to the graph edit distance approach proposed in this paper, MASKS, RNN, and GM refer to graph matching approaches reported in [18] using dynamic masks, recursive neural networks, and graph edit distance, respectively, whereas MLP refers to a non-structural neural network approach [19].

From the experimental results we find that the proposed method performs clearly better than the best graph matching approach reported in [18]. A comparison of the training error and test error reveals that a slight overfitting occurs. However, the ability of the graph matching approach to generalize well on unseen data seems to be sufficiently strong. Using the approximate matching algorithm, the classification runs very fast in comparison to other graph edit distance methods. On a regular workstation it takes 27 minutes to conduct a (non-optimized) graph classification of all 4,000 fingerprints of the NIST-4 database. Although the exact edit distance computation would be feasible for these graphs, experiments indicate that the classification takes by far longer (100h instead of 3 minutes for 500 graphs) and results in a lower classification rate.

It is well known that the definition of adequate cost functions is crucial for the performance of a graph edit distance based classification system. In our experiments, we used simple edit costs based on constant costs and Euclidean distances. One major drawback of this edit cost function, and thus a shortcoming of our classification approach, is that all costs are defined in a location-independent way; that is, the information where in the attribute space an edit operation occurs is not taken into account. For a number of graph matching problems, it turns out that location-dependent edit cost functions automatically learned beforehand from a sample set of graphs can significantly improve the recognition performance [20], which may also be of interest in future investigations in the context of fingerprint graph classification.

5 Conclusions

In the present paper we propose a fingerprint classification system by means of error-tolerant graph matching. Our main contribution is an algorithm for the extraction of regions in the ridge orientation field that are relevant for the

Table 1. Fingerprint classification rate on the NIST-4 database

Data set	Classifier	5 classes
Training set	GED	82.6
Test set 1	GED	80.27
Full database	GED	80.85
Test set 2	GED	80.25
	RNN [5, 18]	76.75
	MASKS [17, 18]	71.45
	GM [18]	65.15
	MLP [18, 19]	86.01

classification. Extracted regions correspond to singular points and characteristic connections between core and delta points. To assign one of the five most common Henry classes to fingerprints, we use a graph edit distance approach. In experiments on the NIST-4 fingerprint database, the proposed method is found to outperform graph matching systems reported in recent years. In the future we intend to address the classification problem based on the proposed directional variance with non-structural classifiers and study whether combinations of classifiers may lead to more robust performance results. In addition we plan to investigate if more complex edit cost functions than the one used in this paper could further improve the classification accuracy.

Acknowledgment

This research was supported by the Swiss National Science Foundation NCCR program "Interactive Multimodal Information Management (IM)2" in the Individual Project "Multimedia Information Access and Content Protection". The authors would also like to thank Alessandra Serrau and Prof. Fabio Roli for helpful discussions.

References

1. Maltoni, D., Maio, D., Jain, A., Prabhakar, S.: Handbook of Fingerprint Recognition. Springer (2003)
2. Yager, N., Amin, A.: Fingerprint classification: A review. Pattern Analysis and Applications **7** (2004) 77–93
3. Maio, D., Maltoni, D.: A structural approach to fingerprint classification. In: Proc. 13th Int. Conf. on Pattern Recognition. (1996) 578–585
4. Lumini, A., Maio, D., Maltoni, D.: Inexact graph matching for fingerprint classification. Machine Graphics and Vision, Special Issue on Graph Transformations in Pattern Generation and CAD **8** (1999) 231–248
5. Marcialis, G., Roli, F., Serrau, A.: Fusion of statistical and structural fingerprint classifiers. In Kittler, J., Nixon, M., eds.: 4th Int. Conf. Audio- and Video-Based Biometric Person Authentication. LNCS 2688 (2003) 310–317
6. Yao, Y., Marcialis, G., Pontil, M., Frasconi, P., Roli, F.: Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines. Pattern Recognition **36** (2003) 397–406

7. Karu, K., Jain, A.: Fingerprint classification. Pattern Recognition **29** (1996) 389–404
8. Kawagoe, M., Tojo, A.: Fingerprint pattern classification. Pattern Recognition **17** (1984) 295–303
9. Kass, M., Witkin, A.: Analyzing oriented patterns. Computer Vision, Graphics, and Image Processing **37** (1987) 362–385
10. Bigun, J., Granlund, G.: Optimal orientation detection of linear symmetry. In: Proc. 1st Int. Conf. on Computer Vision, IEEE Computer Society Press (1987) 433–438
11. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. Int. Journal of Pattern Recognition and Artificial Intelligence **18** (2004) 265–298
12. Sanfeliu, A., Fu, K.: A distance measure between attributed relational graphs for pattern recognition. IEEE Transactions on Systems, Man, and Cybernetics **13** (1983) 353–363
13. Bunke, H., Allermann, G.: Inexact graph matching for structural pattern recognition. Pattern Recognition Letters **1** (1983) 245–253
14. Neuhaus, M., Bunke, H.: An error-tolerant approximate matching algorithm for attributed planar graphs and its application to fingerprint classification. In Fred, A., Caelli, T., Duin, R., Campilho, A., de Ridder, D., eds.: Proc. 10th Int. Workshop on Structural and Syntactic Pattern Recognition. LNCS 3138 (2004) 180–189
15. Zhou, R., Quek, C., Ng, G.: A novel single-pass thinning algorithm and an effective set of performance criteria. Pattern Recognition Letters **16** (1995) 1267–1275
16. Watson, C., Wilson, C.: NIST special database 4, fingerprint database. (1992)
17. Cappelli, R., Lumini, A., Maio, D., Maltoni, D.: Fingerprint classification by directional image partitioning. IEEE Transactions on Pattern Analysis and Machine Intelligence **21** (1999) 402–421
18. Serrau, A., Marcialis, G., Bunke, H., Roli, F.: An experimental comparison of fingerprint classification methods using graphs. In: Proc. 5th Int. Workshop on Graph-based Representations in Pattern Recognition. (2005) Submitted.
19. Jain, A., Prabhakar, S., Hong, L.: A multichannel approach to fingerprint classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **21** (1999) 348–359
20. Neuhaus, M., Bunke, H.: A probabilistic approach to learning costs for graph edit distance. In Kittler, J., Petrou, M., Nixon, M., eds.: Proc. 17th Int. Conference on Pattern Recognition. Volume 3. (2004) 389–393