Graph Representation of Fingerprint Topology *

Werner Ölz, Walter Kropatsch
Pattern Recognition and Image Processing Group
Institute for Computer Aided Automation, Vienna University of Technology
Favoritnerstr. 9, A-1040 Vienna, Austria
e-mail: {oelz, krw}@prip.tuwien.ac.at

Abstract

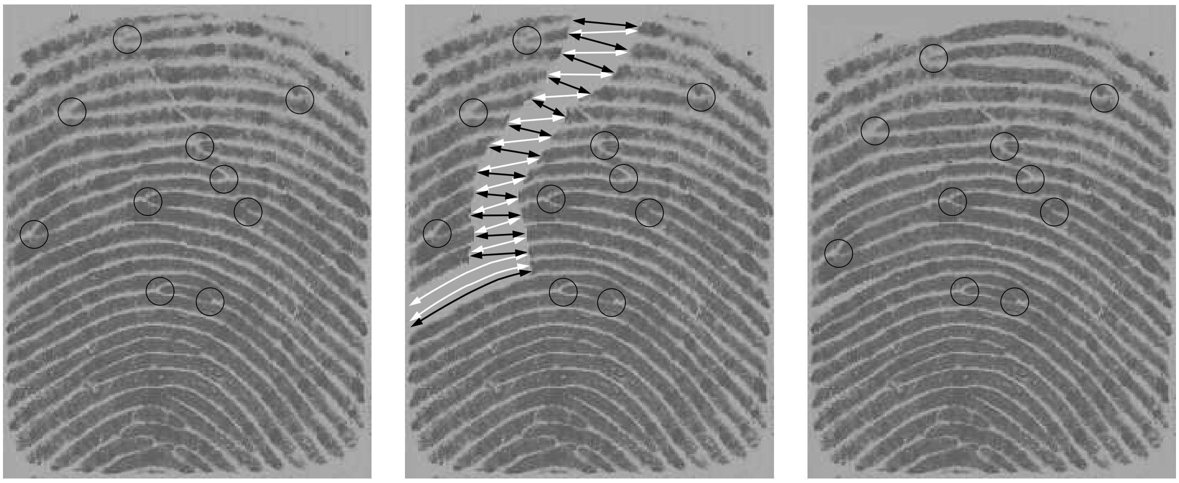
Fingerprints are one of the most important biometrics in use today, and various algorithms and methods exist for automatic fingerprint identification and verification. However, this paper characterizes situations in which the most important class of automatic fingerprint matching algorithms can easily fail, because they do not take the ridge topology as a whole into consideration. We then present a graph-based fingerprint representation that encodes the line topology. and might prove useful in differentiating between such topologically different fingerprints that only have similar minutiae. An algorithm based on dual graph contraction will be given to derive this graph representation automatically from a preprocessed, skeletonized fingerprint image.

1 Introduction

Fingerprints are one of the most important biometrics in use today. They are used in various areas (e.g. access control systems, criminal investigations) for identification and identity verification purposes. These prints consist of ridges on the finger tips that form flow-like line patterns which are, according to our present knowledge, characteristic of an individual person and can therefore be used to identify a person uniquely.

There are various methods for fingerprint matching, but the most widely used method for automatic fingerprint matching systems is minutiae matching [4]. Minutiae are special points in a fingerprint where ridges end or bifurcate (see Fig. 1a). Basically, these methods consist of extracting the minutiae along with certain minutiae features in the fingerprints and comparing the resulting minutiae sets. Two fingerprints are reported as identical, iff a certain number of minutiae are identical in the two prints with respect to the minutiae features extracted (e.g., their location or direction with respect to a reference point). The various minutiae matching algorithms differ mainly in the minutiae features used and the metric for measuring minutia similarity. See [2] for one representative minutiae matching algorithm.

*This paper has been supported by the Austrian Science Fund under grant P14445-MAT.



(a) (b) (c)

Figure 1: Two fingerprints with almost identical minutiae, but different topology: (a) Original fingerprint. (b) Connections between left and right part in the original (white arrows) and the modified (black arrows) fingerprint. (c) Modified fingerprint.

However, minutiae and minutiae features are not sufficient to distinguish between different fingerprints under certain circumstances. In section 2, situations will be characterized in which fingerprints with different line topologies and similar minutiae cannot be distinguished, if only the minutiae are taken into consideration for the comparison. We then present in section 3 a novel fingerprint representation that encodes the line topology of a fingerprint as a graph and is able to capture the difference between the fingerprints we characterized in section 2. Section 4 gives some results, and section 5 contains a conclusion and an outlook.

2 Motivation

Minutiae features are local properties of a certain point (i.e. the minutia itself) or region (around a minutia) in a fingerprint. In comparing two minutiae sets, information in between them is not used in the matching process in general. To see, why this can lead to false positive results, take a look at Fig. 1: Fig. 1a shows a scanned fingerprint in which the minutiae in the upper part are marked. Fig. 1b shows the same fingerprint, but this time cut into two parts that are connected by a series of black and white arrows. These arrows indicate two different ways in which the ridges in the two parts of the fingerprint can be connected. The white arrows indicate the original connections in between them, as shown in Fig. 1a, whereas the black arrows show an alternative way of connecting them. In Fig. 1c, these alternative connections were realized by inserting them into the original fingerprint (Fig. 1a) and shifting the left upper part downwards in order to “straighten out” the resulting connections between the two parts.

This cut runs through a part of the fingerprint that does not contain any minutiae, so any modifications made in this part of the fingerprint should not severely influence the result of a minutiae matching algorithm. However, the line topology of the resulting fingerprint is

Figure 2: Preprocessed, skeletonized fingerprint

severely changed. The minutiae in the two parts are now connected differently, and a human fingerprint examiner would recognize them as different.

We ran these two fingerprints through a minutiae matching algorithm [2]. The resulting match score amounted to 80% of the maximum possible match score, indicating that the two fingerprints match. We also ran them through a commercial fingerprint matching system which gave a 90,5% match score, indicating a match as well. These results suggest that current minutiae matching algorithms cannot distinguish between different fingerprints with similar minutiae under certain conditions. Namely, when there is a larger region without minutiae that connects minutiae on both sides, minutiae matching algorithms have trouble distinguishing between fingerprints in which these minutiae are interconnected differently. The local geometry at each minutia point and their distribution over the fingerprint are similar, yet the global ridge topology is different.

Therefore, we propose in the next section a graph representation for fingerprints that captures the ridge topology. This representation should then allow us to differentiate between different fingerprints in cases like the fingerprints in Figs. 1a and 1c.

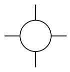
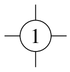
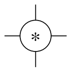
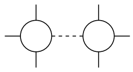
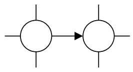
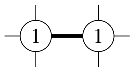
3 Graph representation

In [1], graphs were used to represent technical line drawings and their topology. Here, we use the same basic idea, but with a different representation of lines, line endings and junctions in the graph. Contrary to [1], we start directly at a pixel based grid graph and perform a series of simplification steps to achieve a condensed graph representation of the fingerprint topology.

This graph representation is derived from a preprocessed and skeletonized fingerprint. Since this work is not concerned with fingerprint enhancement or minutiae detection, it is assumed throughout this paper that the fingerprint to be represented went through an enhancement stage and is already skeletonized (see Fig. 2 for an example).

The successive simplification steps of the pixel based grid graph are performed via dual graph contraction [3]. Dual graph contraction is a process by which the dual image graph of one level is contracted into the smaller dual image graph of the next level, thus building an

Table 1: Node and edge labels

Label	Symbol	Explanation
0–node		contains no line element
1–node		a single line element
*–node		a bifurcation point
0–edge		adjacent nodes not on same line
shortest 0–edge		0–edge on shortest path to a line segment
1–edge		adjacent nodes on same line

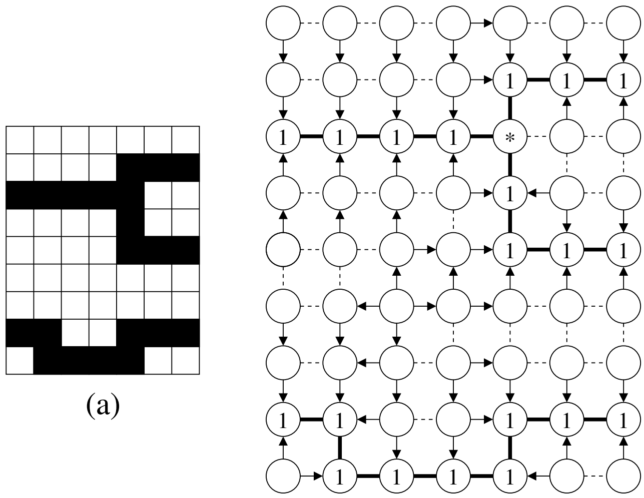
irregular graph pyramid. By taking into account the dual graph, unnecessary double edges and self loops which do not contribute to the topology of the graph can be identified and thus eliminated. A formal outline of dual graph contraction can be found in [3].

The surviving nodes are chosen according to rules that are formulated in terms of node and edge labels. The labels and rules are based on [1], but differ mainly in that we don't differentiate between line endings and other line elements (1-nodes and 2-nodes in terms of [1]). As a result, a single straight line is not encoded by two connected nodes representing the two line endings, but by a single node representing the whole line segment. As we will see later, information about line endings that are minutiae is not lost this way.

3.1 Node and edge labels

The labels we used are listed in Table 1. Before assigning the node and edge labels, a distance transform is performed that calculates for each white (i.e. background) pixel the minimum distance to a black pixel (i.e. line segment). The labels are then assigned as follows:

- Each node that corresponds to a white pixel becomes a 0-node.
- Each node that corresponds to a black pixel with three or more adjacent black pixels becomes a *-node (indicating a bifurcation point in the fingerprint).
- All remaining nodes become 1-nodes (indicating a single line segment).
- All edges adjacent to at least one 0-node become a 0-edge.
- All remaining edges become 1-edges.



(a) (b)

Figure 3: (a) Example line segments used in Fig. b. (b) Labeled grid graph corresponding to Fig. a

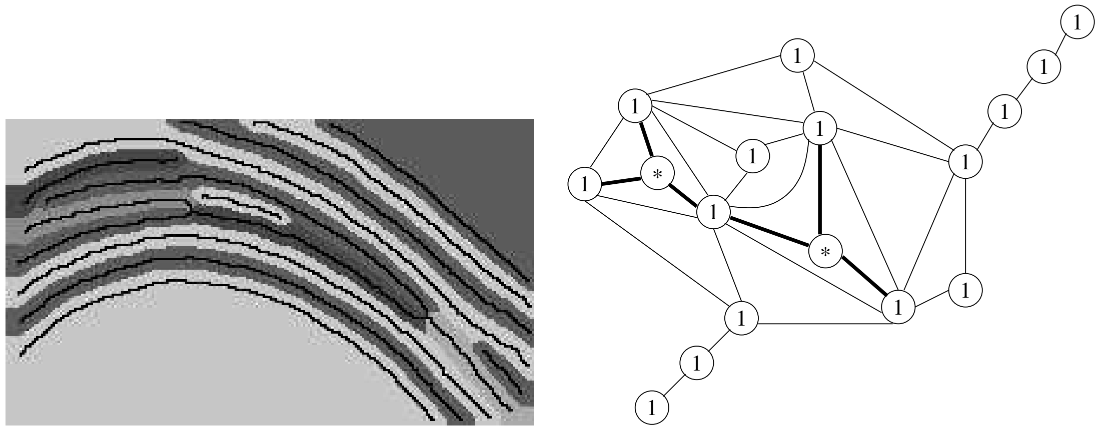
- For each 0-node, the incident edge lying on the shortest path to the closest line segment (as indicated by the distance transform performed beforehand) is marked as the shortest 0-edge.

Since the shortest path is not always uniquely determined, collision rules are applied in order to make sure that only one path is chosen. Notice also that if the shortest path leads to a *-node, the next best path to a 1-node must be chosen, as *-nodes cannot merge with 0-nodes (see below). Fig. 3 shows a short example for a graph that results from these label assignments. The arrows mark shortest 0-edges and point in the direction of decreasing distance for illustration purposes only; the resulting graph is not directed. Notice that no collision rules for multiple shortest paths have been applied in Fig. 3b yet.

3.2 Contraction rules

The contraction rules are as follows:

- *-nodes cannot merge with any other node, they always survive.
- 1-nodes can merge with adjacent 1-nodes, if they are connected by a 1-edge, i.e. if they are part of the same line segment.
- 1-nodes can merge with 0-nodes, if they are connected by a shortest 0-edge.
- 0-nodes can merge with adjacent 0- and 1-nodes, if they are connected by a shortest 0-edge.



(a) (b)

Figure 4: (a) Segmentation induced by the labeling and contraction rules. (b) Final graph.

Since the shortest 0-edges were determined using a distance transform, the segmentation of the fingerprint implied by the contraction rules (see Fig. 4a) looks regular in the sense that each line segment only covers those parts of the background that are closest to it. Note that these segmented regions correspond to the receptive fields of the nodes in the final graph.

Iteratively, surviving nodes are chosen randomly. Only 0-nodes are disfavored as surviving nodes, because they should (and will) disappear sooner or later. For each surviving node, the contraction rules determine, which other node(s) can merge with it. These iterations are performed, until no further edges can be contracted.

3.3 Properties of the resulting graph

The resulting graph does not contain any 0-nodes or shortest 0-edges. There are exactly as many *-nodes in the final graph as there were at the beginning. For every straight line segment in the fingerprint, there is exactly one 1-node in the final graph (see Fig. 4. 0-edges in the final graph indicate that the receptive fields of the two connected nodes are adjacent. 1-edges only occur next to *-nodes; they connect a bifurcation point with the line segments that meet in that point.

The minutiae are represented in two different ways in the graph:

- Bifurcations are encoded by a single *-node that is connected to the incident line segments by a 1-edge.
- Line endings do not appear as nodes in the graph, but as faces. At a line ending, a line segment is cut off by two surrounding line segments, thus forming a face in the graph.

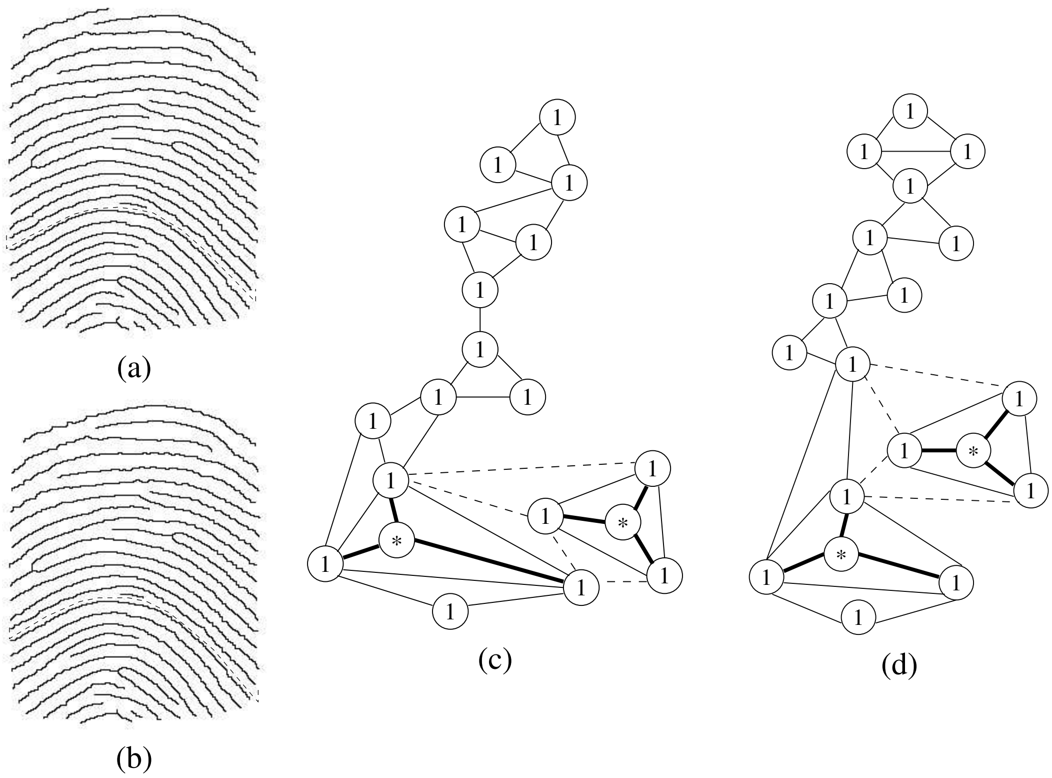


Figure 5: (a) and (b) Skeletonized fingerprints from Fig. 1. (c) and (d) Corresponding graph representations

4 Results

Fig. 5 shows the graph representation of the two fingerprints in Fig. 1. Notice that only the upper part is shown where we modified the original fingerprint. Two important differences can be noticed in these graphs:

- The bifurcation points in the lower part are interconnected differently. The edges that changed are dashed in Fig. 5.
- The minutiae in the upper part (encoded as faces) are related differently, as well.

Since this graph representation is capable of capturing the modifications introduced in section 2, we believe that fingerprint matching schemes based on such a graph representations can be used to differentiate between fingerprints with similar minutiae, but different ridge topology.

5 Conclusions and outlook

We have presented a graph representation for fingerprints that captures the topology of its ridge structure. This representation is capable of capturing differences between fingerprints that minutia-based matching methods cannot always detect. Since we can characterize cases where these methods tend to report false positive matches, while our representation shows clear differences, matching these graphs may prove to be a useful supplement to minutia matching algorithms. It might be used, for example, to verify matches reported by minutia matching algorithms in fingerprint identification applications.

What needs to be done now, is to develop algorithms that take advantage of this graph representation to compare fingerprints. Until now, we can only visually compare the resulting graphs for equality or similarity (using the matching software in our heads). The next step will be to develop a graph matching algorithm that matches two fingerprint graphs automatically. An essential aspect in doing so will be identifying different node configurations that are compatible, i.e. that may originate from the same fingerprint and differ only because one of the fingerprint scans is of bad quality (e.g. a line ending "merging" with a straight line in a bad quality scan, thus forming a bifurcation point).

Finally, in order to use these graphs alone for fingerprint matching, the distinctiveness of the ridge topology in general must be investigated. We have argued in this paper that in certain cases, it can distinguish between different fingerprints that minutiae matching algorithms don't recognize as different. But its overall distinciveness remains to be further investigated.

6 Acknowledgements

We would like to thank Anil Jain and Arun Ross for supporting our experiments.

References

[1] Burge M. and Kropatsch W., A Minimal Line Property Preserving Representation of Line Images, *Computing*, Vol. 62, pp. 355-368, 1999.

[2] Jain A., Hong L. and Bolle R., On-Line Fingerprint Verification, *IEEE Trans on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 4, pp.302-313, 1997.

[3] Kropatsch W., Building irregular pyramids by dual-graph contraction, *IEE Proc. Vision Image Signal Process.*, Vol. 142, No. 6, pp. 366-374, 1995.

[4] Maltoni M., Maio D., Jain A. and Prabhaker S., *Handbook of Fingerprint Recognition*, Springer-Verlag, 2003.