STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LIV**, Number 1, 2009

FINGERPRINT MATCHING ON SMALL IMAGES

FLORINA MUNTENESCU

Abstract. The uniqueness of the ridges' flow pattern is the basis of forensic applications on *fingerprints*, the accuracy of matching two or more fingerprints being its most important issue. A big problem arises when the fingerprint image used is small. In this paper we aim at introducing a method of analyzing a small image so that valid minutiae points can be extracted. For this purpose a constraint satisfaction based technique for finding a match between a candidate fingerprint with one from a database of fingerprints is introduced. We also provide experimental results and a comparison with other similar existing approaches.

1. Introduction

Fingerprints were used as a method of identifying a person for the first time in the 19th century when the classification methods based on patterns appeared and rapidly developed [2]. By 1946 FBI had processed 100 million manually maintained fingerprint cards and by 1971, 200 million cards, the necessity of an automated fingerprint identification system became obvious [2]. Nowadays fingerprints are not used only in forensic science but also as a biometric system of security in everything that requires a high accuracy rate: from devices that allow users to log in to Windows, to enter restricted areas or to authorize transactions [3]. The images used by FBI have 8 bits of grayscale resolution and 768 x 768 pixels, turning a single fingerprint card in 10 Mb of data.

In the approach from this study we are focusing on small images of 90 x 125 pixels, in grayscale. Small images are a solution when we don't have access to large images of a fingerprint and when large images, that are harder to process and also require more disk space for their storage, are not really needed. When using small images in extracting minutiae points [5] it is almost

Received by the editors: April 17, 2009.

2000 *Mathematics Subject Classification.* 68T10, 68U10.

1998 *CR Categories and Descriptors.* I.2.8 [**Computing methodologies**]: Artificial Intelligence – *Problem Solving, Control Methods, and Search*; I.5.0 [**Computing Methodologies**]: Pattern Recognition – *General*.

Key words and phrases. Pattern recognition, Fingerprint matching.

impossible to extract core and delta points, that is why we have focused on introducing elements that, to our knowledge, were not used in fingerprint matching, so far. More precisely the division of the image in quadrants, adding to the features of a minutiae point the quadrant to which it belongs was considered. Another challenging problem that we managed to overcome is to reliably extract minutiae in these small images eliminating many fake points.

The aim of this paper is to introduce small images as a trustworthy source for extracting minutiae points and also to propose a new technique in fingerprint matching based on a constraint satisfaction approach. The rest of the paper is structured as follows. Section 2 describes minutiae points, the way they are extracted from an image and also presents a few matching techniques existing in the literature. A new method of fingerprint matching is introduced in Section 3. Some experimental results are presented in Section 4 and some conclusions of the paper and further work are given in Section 5.

2. Minutiae points extraction and fingerprint matching techniques

In this section we will give a definiton for minutiae points and will present some fingerprint matching techniques existing in the literature.

When using a minutiae based algorithm [7] the fingerprint image is seen as a set of *minutiae points*, i.e., a set of *ridge endings* and *bifurcations* with uniquely determined orientation and location. Even though there are as many as eighteen types of ridge features defined, they can all be considered as complex combinations of two basic features: ridge end and bifurcation, referred together as minutiae points [6]. A ridge end is a point at which a ridge terminates. Bifurcation is a point where a single ridge splits into two ridges. A fingerprint can be seen as an xOy plane, therefore, a point in this plane can be seen in terms of Ox and Oy coordinates. Also an orientation of the ridge can be defined [1]. But to the usual features, we add a new one: the quadrant to which the point belongs. Consequently, we managed to create Equation (1) to define a minutiae point as a 5 tuple, and to illustrated it in Figure 1:

$$MP = \{(x, y, \theta, t, q) \| x \in Ox, y \in Oy, \theta \in [0, 360), \quad t \in \{end, bifurcation\}, q \in \{1, 2, 3, 4\}\} \tag{1}$$

where:

- (x, y) are the coordinates of the point;
- (x, y) θ is the orientation angle;
- t is the type of the point: ridge end or bifurcation;
- q is the quadrant.

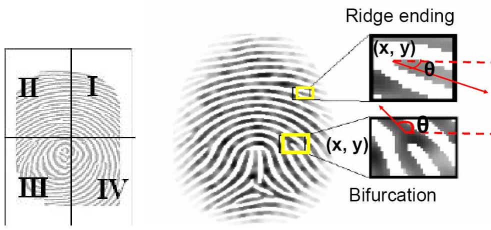

FIGURE 1. Left: A fingerprint divided in quadrants; Right: Minutiae points defined by (x,y) position, θ - orientation and type (ending or bifurcation)

Given two (an input and one from the database) representation, the matching technique determines whether the associated prints are impressions of the same finger. The matching phase typically defines a similarity between two fingerprint representations but finding a similarity function is usually very difficult.

There are various algorithms for fingerprint matching, each with its advantages and disadvantages, and with a more or less possibility of being applied for small images. Next, we are going to briefly describe three of these algorithms.

2.1. **Point Based Matching.** The minutiae point based techniques [5] typically match the input and the template minutia point by first aligning them and afterwards counting the number of matched minutiae. The alignment can be obtained using the orientation field of the fingerprints, the location of singular points as the core and the delta, inexact graph matching on the minutiae graphs etc. The inapplicability of this technique is given by the difficulty of finding the core and the delta points. These points are used to compute the amount of translation and also to easily compare the Euclidian distance to the first ridges that are in the vicinity of the points.

2.2. **String Distance-based Minutiae Matching Algorithm.** Each set of extracted minutiae features is converted to polar coordinates with respect to an anchor point [5]. The two-dimensional features are reduced to one-dimensional string by concatenating points in an increasing order of radial angle in polar coordinates. The string matching algorithm is applied to compute the edit distance between two strings. The edit distance is then used to compute a matching score.

2.3. **Hough Transform-based Minutiae Matching.** First the algorithm estimates the transformation parameters: translation on x and y directions, the rotating angle and a scaling factor. The two sets of minutiae points with the estimated parameters are aligned and the matching pairs are counted. The previous two steps are repeated for the set of discretized allowed transformations. The transformation with the highest matching score is considered to be the correct one [5].

3. Our Approach in Fingerprint Matching

In this section we introduce an approach for fingerprint matching, describing first how we extract the minutiae points and next, how we manage to establish whether two impressions belong to the same finger.

Every image is, in fact, a collection of pixels, each containing three colors in different ratio, range over 0 to 255 RGB values, and where a 0 value of the RGB means that the pixel contains black and 255 white. The image has to be transformed into a matrix having the size of the image that would contain 1 for the pixels where there is a ridge (dark color) and 0 for those where is a valley (light color). So, the pixels that are closer to 0 in RGB are 1 and the rest of them 0. We have noticed that a very important part in this step towards minutiae points' extraction is the value of the threshold, because a value too large could lead to overlapping one or more ridges, loosing in such a way important points: ridges that end could be connected with continuous ones and a ridge that bifurcates in other two can be considered by the computer as just one thicker ridge. But, on the other hand, choosing a very small value for the threshold could have as consequences the appearances of ridge endings in places where the ridge is just thicker or where the finger pressure on the scanner was lighter, and the transformation of a bifurcation point to three ridge endings.

In order to extract the bifurcation points we parse the binary matrix in order to find a 3x3 pixels area whose sum is greater than 5 and the center of the area is a pixel with the value 1. The image being small, the angle of the minutiae can only be 0 or 180. To find the ridge endings we proceed like in the bifurcation points' case but we search for a sum equal with 2. The angles for end points can be: 0, 45, 90, 135, 180, 225, 270 or 315, depending on where the other pixel is placed, with respect to the center of the area.

So far, the algorithm appears to be simple due to the small image but when analyzing the points found, we realize that many of the minutiae points found are *fake*. Our algorithm detects in the area from the edge of the fingerprint a lot of points (both end and bifurcation) that are actually just the border of the fingerprint and those have to be eliminated. Another problem appears at the bifurcation points where the algorithm seems to find consecutive minutiae points around one valid point. So, the solution to these fake points is to delete the sequences of consecutive points, keeping just the first one, which is the valid point.

When trying to find a match for an input fingerprint in a large database, computational time is an important issue. So, instead of applying a time consuming matching algorithm for each fingerprint, we propose a different approach: we first use an algorithm that would *filter* the fingerprints, keeping only those that have a good chance of being a possible match. The algorithm does not work on partial fingerprints.

Our problem has a number of constraints that are, one by one eliminated: number of points per quadrant, translation and rotation factors.

Based on experimental results we have noticed that the number of points from each quadrant is different: for quadrant one is between 9 and 25, for quadrant two between 10 and 50, for quadrant three: 1 to 15 points and for quadrant four 0 to 15. This means that the first step in matching two fingerprints is searching for possible matches in certain ranges, specific for each quadrant. Moreover, we have noticed that we can find a scope even for end and bifurcation points. Therefore, when we remain with a smaller number of fingerprints, hence number of minutiae points, we can 'afford' applying a matching algorithm.

The second step is finding the translation factors on Ox an Oy [4]. In order to do that, we search again in a certain range, the number of pixels (Euclidian distance) with which the image had translated, finding the most possible position to which the point had relocated. The higher number of points that have the same distance, the higher the probability of the fingerprint of being a match. When searching for the analogous points we search between points in the same quadrant, that have the same type and, in case of the bifurcation minutiae points, the same angle.

Knowing the distance, consequently, the translated corresponding points, the third step is to find between the ridge endings how many of them were rotated with the same angle. If the angles seem to vary a lot and a rotation cannot be found the two impressions surely do not match and the fingerprint from the database is removed from the search space. The previous two steps are repeated for each set of minutiae points and the fingeprint that has the highest score is considered to be the match.

4. Experimental results

The algorithms described in this paper were tested on fingerprint images that were obtained from notebook scanners alongside with artificially generated ones, resulting a database with almost 22,000 fingerprints and more than 1,300,000 minutiae points. The subjects were taken 2 impressions for each finger, from which one was inserted in the database and the other one used for testing. Partial fingerprints were removed. At the first step, 10 to even 1 possible fingerprints were found. At the second filtering 3 to 1 possible fingerprints were found and at the last 1 possible fingerprint was found. The very small number of fingerprints found after the filtering step that we have proposed show the effectiveness of this preprocessing step applied before matching.

Errors were noticed to appear when the pressure on the scanner was not constant, resulting, especially in the upper one and two quadrants, a very large number of ridge endings that made the algorithm consider that the fingerprint does not have a match.

5. Conclusions and Further Work

We have presented in this paper a fingeprint matching algorithm for small images that can be applied when time and space consuming techniques cannot be used. The capabilty of the minutiae extracting and matching was proved in experimental results and the benefits of using these algorithms on small images in comparison with others, were proved. The further work will be revolving around accepting partial fingerprints.

References

[1] Jidnya Shah Arun Ross and Anil K. Jain, *Towards reconstructing fingerprints from minutiae points*, Proc. Of SPIE Conference On Biometric Technology For Human Identification (2005).
[2] Christopher M. Brislawn, *The fbi fingerprint image compression standard.*
[3] A.K. Jain S. Prabhakar D. Maltoni D. Maio, *Introduction*, Handbook of fingerprint recognition, 2003, pp. 1–53.
[4] Venu Govindaraju Praveer Mansukhani Sergey Tulyakov, *Using support vector machines to eliminate false minutiae matches during fingerprint verification.*
[5] Nalini Ratha and Ruud Bolle, *Fingerprint matching*, Automatic fingerprint recognition systems, 2003, pp. 229–246.
[6] Nalini K. Ratha Ruud M. Bolle Andrew W. Senior and Sharath Pankanti, *Fingerprint minutiae: A constructive definition.*
[7] S. M. Zamshed Farhan S. M. Mohsen and M. M. A. Hashem, *Automated fingerprint recognition: Using minutiae matching technique for the large fingerprint database*, 3rd International Conference on Electrical and Computer Engineering (2004).

Department of Computer Science, Babeş-Bolyai University, 1, M. Kogălniceanu Street, Cluj-Napoca, Romania

E-mail address: `mfie0602@scs.ubbcluj.ro`