Improvements and Adaptations in Fingerprint Processing Techniques for the Creation of a High-quality Minutia Database

Mateus Campos Puttini Fernandes and Ricardo Staciariini Puttini

Abstract - The use of fingerprints for biometry presents some interesting technological challenges, especially when large populations are involved. This paper describes how a high-quality database of fingerprint minutiae can be created. First research was carried out to establish which algorithms are state of the art. Certain techniques are then selected, and improvements are proposed for some. The results were analyzed aiming to evaluate the quality of the biometric data base generated. The information in the database so obtained is quite good (approximately 92% of the detected minutiae are real), but further work can doubtless improve the quality of the results.

1 - Introduction

Biometry is rapidly becoming one of the most effective modes of human identification [1]. Many modern information systems associate biometric information with legitimate users’ personal data in order to improve security and control access.

Compared to tokens, keys and passwords, biometric information is considered more difficult to imitate or share [1]. As such, it can greatly improve the security level of an information system when used in conjunction with more usual techniques.

The most common biometric data used at present are fingerprints [1]. They require little processing time, are easy to sample, and demand only inexpensive technology.

Biometric identification consists of two steps: enrollment, where biometric information is recorded and introduced into a database; and identification, where the database is used to recognize individuals. This paper focuses on the enrollment phase. Indeed, the creation of a high-quality database is its main topic. The basic process proposed for creating this database is illustrated in Figure 1.

The data most frequently used by fingerprint recognition systems are the locations of ridge bifurcations and endings, known as minutiae [1]. Other unique characteristics can be extracted from the ridges, but this paper considers only ridge endings and bifurcations.

A literature review yielded several promising techniques related to the fingerprint enrollment process. A selection of these, shown in the
Improvements and Adaptations in Fingerprint Processing Techniques...  

"Minutiae Extraction" subprocess of Figure 1, were implemented and tested on the first FVC2004 fingerprint database (DB1) [2].

This paper also suggests improvements and adjustments to some of the techniques shown in Figure 1: Pore Elimination [3], Useful Area Definition [4], Thinning [5], and Spurious Minutiae Filtering [6].

All the proposed changes aim to improve the quality of information in the final database. A large body of research evaluates the quality of algorithms and processes used in the matching phase, reporting false acceptance rates (FAR) and false rejection rates (FRR) among other statistics [7]. As this paper focuses on the enrollment process, and more specifically on the minutia extraction procedure, such assessments do not provide the desired vision. Rather, this work will evaluate the process of biometric identification as a whole.

In this paper, the quality of a database is defined as the percentage of true biometric information therein. As the fingerprint database used for testing (FVC2004 DB1) does not inform the location or amount of minutiae in each of its images, this percentage was obtained by visual inspection of each image. In some cases, true and false minutia could also be counted by automatic means. Some works focused on false minutia filtering do assess their outcome by the counting method [6], but none have used it to validate the whole enrollment phase. For this reason, new procedures have been established to test the feasibility of the method.

2 - Pore elimination

This process consists of finding and eliminating the small white dots that appear in fingerprint ridges. These dots are skin pores, and if not removed can lead other algorithms to find minutiae in places where they do not really exist. The algorithm used here is based on reference [3], where the pores are modeled by a slightly modified 2-dimensional Gaussian function (1).

\[
P(x, y) = 1 - e^{-(x^2 + y^2)/2}
\] (1)

The values of \(P(x, y)\) represent depth, and lie in the interval [0,1]. Zero corresponds to the center (bottom) of the pore, while 1 represents the surface of the skin. The fingerprint images, however, consist of grayscale values ranging from 0 to 255. In this case the ridges (surface of the skin) correspond to zero, while the valleys correspond to 255. Thus, two transformations
must be applied to the greyscale image: first it is inverted, then it is rescaled to the [0,1] interval. Equation (2) accomplishes both conversions.

\[ F(x,y) = 1 - \frac{I(x,y)}{255} \]  

(2)

The size of the pore model is an important consideration: the more points it predicts, the more computationally expensive finding pores becomes. On the other hand, a higher resolution results in more accurate pore identification. Reference [3] claims that a 3×3 model is enough to find most pores, but some pores were not detected by this model in the present research. In hopes of improving performance, models with 5×5 and 7×7 pixels were tested as well, with results to be described later in this section. Figure 2 shows a 5×5 model generated by Equation (1).

\[ \xi = E - \frac{1}{w \cdot h} \sum_{x=1}^{w} \sum_{y=1}^{h} [E(x,y) - \bar{E}]^2 \]  

(3)

where
- \( \xi \) is the maximum error threshold;
- \( \bar{E} \) is the average error;
- \( w \) is the fingerprint image width; and
- \( h \) is the fingerprint image height.

This procedure generates a map of areas with a high probability of containing pores. The minimum error points are very likely to be actual pores. These minimum points are searched by means of an inspection window that defines the neighborhood to be considered. This paper considered a 5×5 pixel window.

Once the pores have been located, they should be eliminated from the original fingerprint image. Reference [3] does not prescribe a way to restore the underlying surface. This work proposes the inverse rescaled pore model given by Equation (4), where \( \max(I) \) is the maximum grayscale value in the original fingerprint image. Figure 3 illustrates the filling model.

\[ PP(x,y) = \max(I) \cdot e^{-kx^2-y^2} \]  

(4)

Figure 2 – A 5x5 pixel pore model.

A quadratic error inspection of the local fingerprint image surrounding each pixel is then performed, comparing the pore model to the image values. This procedure employs the same expression proposed in reference [3], but a maximum error value could not be encountered in their suggested method. Therefore, this threshold was defined as in Equation (3):

Figure 3 - Pore Filling Model
2.1 - Testing procedure

For each fingerprint image in DB1 of FVC2004, pores were extracted as described above. The total number of pores was automatically determined, but false pores had to be visually counted. This was done by superposing a map of detected pores on the original image. False pores do not lie in fingerprint ridges, but between two close ridges. Figure 5 shows an example.

Table 1 – Spurious pores by model.

<table>
<thead>
<tr>
<th>Pore model dimensions (pixels)</th>
<th>Average number of false pores (% – number)</th>
<th>Average number of pores</th>
</tr>
</thead>
<tbody>
<tr>
<td>3×3</td>
<td>3.94% – 12.2</td>
<td>309</td>
</tr>
<tr>
<td>5×5</td>
<td>2.53% – 9.5</td>
<td>374</td>
</tr>
<tr>
<td>7×7</td>
<td>2.89% – 11.2</td>
<td>386</td>
</tr>
</tbody>
</table>

Another useful and related measure is the number of false minutiae found in each image at the end of enrollment processing. Table 2 shows the results.

Table 2 – Relation between pore model dimensions and false minutiae.

<table>
<thead>
<tr>
<th>Pore model dimensions (pixels)</th>
<th>Average number of false minutiae (% – amount)</th>
<th>Average total number of minutiae</th>
</tr>
</thead>
<tbody>
<tr>
<td>3x3</td>
<td>23.8 % – 8.1</td>
<td>34</td>
</tr>
<tr>
<td>5x5</td>
<td>7.8 % – 2.2</td>
<td>28</td>
</tr>
<tr>
<td>7x7</td>
<td>11.3 % – 3.4</td>
<td>30</td>
</tr>
</tbody>
</table>

From these results it became clear that the 5×5 pore model is best. The 3×3 model finds a substantially smaller number of pores (about 20% fewer) and clearly produces more false minutiae. The 7×7 model and 5×5 models have similar performance, but the former finds more false pores in the spaces between ridges. When filled, these points join the ridges and generate more false minutiae.
3 - Useful area determination

This step determines which parts of the fingerprint image represent background rather than ridges and valleys, and therefore should not be considered in any other step of fingerprint processing. Thus, as early and precisely as possible, this procedure is applied to create the useful area mask.

Two techniques were combined in this paper. The first is based on the standard deviation of image grayscale values. Useful regions of the image have a large grayscale variation (ridges and valleys), so one can simply require that the standard deviation in a local neighborhood is greater than some threshold. The second method verifies that the ridge frequency calculated for a certain region lies within a fixed interval. For images with a resolution of about 500 dpi, the local ridge frequency should lie between $1/3$ and $1/25$ (per pixel) [8,9].

To calculate the local value of the standard deviation, a window size must be defined. If too large, the window will not correctly define the background of the image. If too small, however, the chance of finding a small value of the standard deviation within an obviously useful area increases. For this research, a 16x16 window was used.

However, there is another issue with this technique: the direction in which the image is covered. For example, if the image is tiled from left to right with 16-pixel windows but the image width is not a multiple of 16, the last window of each row will be truncated (see Figure 6c). The probability of finding a standard deviation value above the threshold will not be the same in such truncated windows.

To deal with this problem, the image was analyzed in four directions: left to right and top-down, left to right and bottom-up, right to left and top-down, and finally right to left and bottom-up. Note that the useful areas are not identical: for example, in the upper left corner of the left-to-right (Figure 6b) and right-to-left maps (Figure 6c). The four masks are summed to create the final mask.

The frequency map technique is implemented following reference [4]. This procedure associates a frequency value with each image pixel, resulting in a useful area mask with a much higher-resolution contour. However, the border also presents some irregularities (Figure 7b) such as discontinuous regions. To correct these artifacts, a sequence of dilations and erosions is performed to make the final mask more regular. This procedure is implemented following reference.
Improvements and Adaptations in Fingerprint Processing Techniques...  

3.1 - Results assessment

The goal here is to evaluate the contributions of both techniques to the final result. The number of hidden minutiae was recorded for each technique as well as for their combination.

Table 3 – Number of minutiae hidden by useful area masks.

<table>
<thead>
<tr>
<th>Useful area definition technique</th>
<th>Average number of hidden minutiae</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard deviation</td>
<td>73</td>
</tr>
<tr>
<td>Frequency map</td>
<td>32</td>
</tr>
<tr>
<td>Both techniques</td>
<td>93</td>
</tr>
</tbody>
</table>

Some conclusions can be drawn from Table 3. First, there is obviously some overlap between the two sets of minutiae. Second, the average number of minutiae eliminated by both techniques is only 12. These points can be clarified by the Venn diagram shown in Figure 9.

Clearly, the number of potentially spurious minutiae eliminated by both techniques is greater than the number that can be removed by either individually. This shows that combining the two techniques can improve the quality of biometric information extracted from fingerprint images.

4 - Thinning

The purpose of thinning is to reduce ridges in the fingerprint image to well-defined lines, a task also known as skeletonization. This paper has adapted a generic thinning procedure, first proposed in reference [5]. While this procedure has been used by several works in the literature, none
of them have analyzed the technique. This paper proposes some ways of adapting the technique to work with binary fingerprint images.

The basic algorithm combines serial and parallel thinning, avoiding some of the problems caused by using either method in isolation [5]. It then searches the binary image for template patterns which permit one or more points to be eliminated without losing connectivity. However, the algorithm is generic and does not take into account the specific conditions of fingerprint thinning.

When thinning is complete, minutiae location becomes a very simple procedure (Figure 10). The lines are inspected, and any points with only one neighbor are counted as ending minutiae. Pixels with two neighbors are considered part of a continued line. Those with three or more neighbors are marked as bifurcations, another kind of minutia. If a ridge point has no neighbors, it is not marked as a minutia.

Implementing the original algorithm yielded false minutiae in some specific thinned ridge patterns. Specifically, these patterns present up to four minutiae where only one should be placed (Figure 11); each of the points shown has three neighbors (including continuing points off the grid).

Figure 10 – Minutiae location:
- Analyzed pixel;
- Neighbors;
- Ridge points;
- Background.

Figure 11 – The additional templates proposed by this paper.

The central point of each pattern presented above can be removed without changing the connectivity of the ridges. By doing so, the false minutiae go away. Figure 12 below illustrates one such case.

Figure 12 – One of the proposed corrections.

The pointed pattern shown in left image of Figure 12 shows four points with three neighbors, while the right image shows that only one point will be considered as a bifurcation for having three neighbors.

4.1 - Results assessment

In order to validate the choice of thinning procedure, we compare the number of false minutiae to that obtained running an alternative algorithm presented in reference [4].

<table>
<thead>
<tr>
<th>Thinning technique</th>
<th>Average number of false minutia per image</th>
<th>Average total number of minutiae</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per ref. [4]</td>
<td>12.2</td>
<td>38</td>
</tr>
<tr>
<td>Adapted from ref. [5]</td>
<td>2.2</td>
<td>28</td>
</tr>
</tbody>
</table>

The adapted algorithm used in this paper produces a significantly better result than that used in reference [4]. The latter is much less strict about maintaining the connectivity of thinned
Improvements and Adaptations in Fingerprint Processing Techniques... lines. For example, if a ridge line is broken by the thinning process two false ending minutiae are likely to be created.

Given that the differences between the two columns are exactly the same, it seems highly probable that all extra minutiae appearing in the images processed by the less strict algorithm [4] are false and due to violations of connectivity.

5 - Spurious minutia filtering

The technique used in this paper to identify false minutiae in a thinned image is based on reference [6], which verifies that each minutia is not part of a known invalid line structure. Some invalid configurations are shown in Figure 13.

![Figure 13 – False minutiae at invalid ridge structures (adapted from [6]).](image)

Roughly speaking, the original algorithm analyzes all points in its inspection window then goes over the entire perimeter to count the number of lines intersecting its borders.

This paper presents a faster way to detect false minutiae using the same structures, described in the following algorithm.

For each minutia found in the thinned image:

1. Create $L$, an empty image of size $WxW$ centered on the minutia, such that each point of $L$ corresponds to a pixel in the thinned image.

2. If it’s an ending minutia:
   a) Being $l_a$ the point in $L$ under analysis, initialize $l_a$ with the correspondent value in the analyzed minutia;
   b) Fill $l_a$ with the value 1 (one);
   c) If $l_a$ belongs to the border of $L$, the minutia is considered true and the procedure goes on to the next minutia (step 1); otherwise, proceed to next step;
   d) Search the 8 neighbors of $l_a$ for ridge points with the value 0 in corresponding point of $L$;
   e) If such a point is found, attribute its position to $l_a$ and go to step “2.b”;
   Otherwise, the minutia is considered FALSE and the procedure goes to the next minutia;

3. If it’s a bifurcation minutia:
   a) Assign the values 1, 2, 3 etc. to the points of $L$ corresponding to all 8 neighbors of the candidate minutia, in the clockwise direction;
   b) Assign the value ? 1 to that point of $L$ corresponding to the candidate minutia;
   c) For $l = 1$ to 8:
      i. If $l$ has no more values to assume, the minutia is considered to be true and the procedure goes to the next minutia; otherwise, proceed to next step;
      ii. Position $l_a$ in the point relative to the minutia neighbor pixel that has the same value of $l$;
      iii. If $l_a$ belongs to the border of $L$, the procedure goes on to the next value of $l$ (step “3.b”); otherwise, proceed to next step;
     iv. Verify if, among the 8 neighbors of $l_a$, there is any ridge points that has the 0 value in $L$;
     v. If there is such point, attribute its position to $l_a$ and goes to step “3.c.iii”; otherwise, the minutia is considered false and the procedure goes to the next minutia;
5.1 - Results assessment

The new algorithm should produce results at least as good as that proposed in reference [6], with less computational cost. In addition, the average time spent on minutiae filtering was recorded for both algorithms. Table 5 shows the results.

Table 5 – False minutiae filtering.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average time spent to filter the minutiae of one fingerprint (milliseconds)</th>
<th>Average number of false minutiae found (amount – %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per ref. [6]</td>
<td>8.03</td>
<td>2.184 – 7.6%</td>
</tr>
<tr>
<td>Adapted by this paper</td>
<td>1.27</td>
<td>2.184 – 7.6%</td>
</tr>
</tbody>
</table>

It is obvious from Table 5 that the algorithm proposed in this paper produced exactly the same results in about 84% less time.

6 - Conclusions and future work

The overall quality of a biometric database can be assessed by calculating the percentage of authentic information contained therein. Table 5 shows that the average proportion of true minutiae is 92.2% after the final step of processing. This result is based on all 800 fingerprint images in DB1 of FVC2004, however, and some of these are of very low quality.

Thus, this work has accomplished its stated goal of generating a high-quality biometric database from fingerprint images. Nevertheless, the final content may still be enriched by improving some of the procedures and criteria.

Some possibilities for future work: 1) a deeper study of fingerprint enhancement techniques, to find one that improves on the Gabor filtering used in this work; 2) the algorithm could be parameterized to deal with images captured by other kinds of fingerprint scanners, without decreasing the overall quality of the database; 3) the database could be enriched by researching and selecting a singular point detection method (i.e., cores and deltas [10]); and 4) an assessment of computer costs for the whole algorithm, in terms of CPU time and the computer memory required.

References

R. S. Puttini received his B.Sc. in Electrical Engineering from the University of Brasilia (UnB) in 1995, his M.Sc. in Electrical Engineering from UnB, in 1997, and his Ph.D. in Electrical Engineering from UnB in 2004. Currently, he is an Associate Professor at the University of Brasilia.