Ridge-Based Fingerprint Matching Using Hough Transform

Conference Paper · January 2005

DOI: 10.1109/SIBGRAPI.2005.45 · Source: IEEE Xplore

2 authors, including:

Aparecido Nilceu Marana
São Paulo State University
90 PUBLICATIONS **975** CITATIONS

SEE PROFILE

Ridge-Based Fingerprint Matching Using Hough Transform

[1]Aparecido Nilceu Marana and [2]Anil K. Jain
[1]*UNESP – Faculdade de Ciências – Departamento de Computação – Bauru – SP – Brazil*
[2]*Michigan State University – College of Engineering – East Lansing – Michigan - USA*
{marana44, jain}@cse.msu.edu

Abstract

Despite the efficacy of minutia-based fingerprint matching techniques for good-quality images captured by optical sensors, minutia-based techniques do not often perform so well on poor-quality images or fingerprint images captured by small solid-state sensors. Solid-state fingerprint sensors are being increasingly deployed in a wide range of applications for user authentication purposes. Therefore, it is necessary to develop new fingerprint-matching techniques that utilize other features to deal with fingerprint images captured by solid-state sensors. This paper presents a new fingerprint matching technique based on fingerprint ridge features. This technique was assessed on the MSU-VERIDICOM database, which consists of fingerprint impressions obtained from 160 users (4 impressions per finger) using a solid-state sensor. The combination of ridge-based matching scores computed by the proposed ridge-based technique with minutia-based matching scores leads to a reduction of the false non-match rate by approximately 1.7% at a false match rate of 0.1%.

1. Introduction

Recognition of persons by means of biometric characteristics is an emerging technology in our society. Among the possible biometric traits like face, iris, speech, and hand geometry, fingerprint is the most widely used trait, because of its distinctiveness and persistence over time [1].

Fingerprint is a reproduction of the fingertip epidermis, produced when the finger is pressed against a smooth surface. The most evident structural characteristic of a fingerprint is a pattern of interleaved ridges and valleys [1]. Ridges vary in width from 100 μm, for thin ridges, to 300 μm for thick ridges. Generally, the period of a ridge/valley cycle is about 500 μm [2]. Ridges and valleys often run in parallel, and sometimes they can suddenly come to an end (termination), or can divide into two ridges (bifurcation). Ridge terminations and bifurcations are considered minutiae (small details). There are other types of minutiae in a fingerprint, but the most often used are terminations and bifurcations [1].

Fingerprint recognition is a complex pattern recognition problem. Designing algorithms capable of extracting salient features and matching them in a robust way is quite hard, especially for poor quality images. Despite substantial progress over the last 20 years, fingerprint recognition is still a challenging and important pattern recognition problem [1]. Most fingerprint matching systems are based on matching minutia points between the query and the template fingerprint images. The matching of two minutiae sets is usually posed as a point pattern matching problem and the similarity between them is proportional to the number of matching minutia pairs [1,3].

Although the minutiae pattern of each finger is quite unique, noise and distortion during the acquisition of the fingerprint and errors in the minutia extraction process result in a number of missing and spurious minutiae. Another problem is that the rotation and displacement of the finger placed on the sensor, can lead to different images of the same fingerprint to have only a partial overlap area resulting in only a small number of corresponding minutiae points.

Compact solid-state fingerprint sensors are being increasingly incorporated into keyboards and cellular phones for a wide range of civilian and commercial applications where user-authentication is required [4]. The advent of solid-state fingerprint sensors presents a challenge to traditional minutiae-based fingerprint matching. The problems with minutiae extraction can be more severe if the fingerprint is acquired using a compact solid-state sensor. Solid-state sensors provide only a small contact area for the fingertip and,

therefore, capture only a limited portion of the fingerprint pattern [4].

Given that it is difficult to reliably obtain the minutia points from poor quality fingerprint images or from the small sensor images, other features like ridge orientation and ridge shape should be used for fingerprint matching. Matchers based on the orientation and shape features can also be used to complement the minutia-based techniques. Recently, the use of hybrid fingerprint matchers by using more than one approach has been proposed. Ross et al. [5] have suggested the use of both minutiae and texture information to represent and match fingerprints. Nandakumar and Jain [6] also have suggested the use of both minutiae and ridge information, but in their approach the query image is aligned to match the template image using only the ridges associated with the minutiae.

In this paper, we present a ridge-based technique for fingerprint matching. This technique extracts the major straight lines that match the fingerprint ridges and uses these lines to estimate the rotation and translation parameters necessary to register the query and the template fingerprint images. After the registration, the matching score is computed based on the ridge alignment. The proposed technique has been tested on the MSU-VERIDICOM database, which consists of fingerprint impressions obtained from 160 users (4 impressions per finger) using the Veridicom solid-state sensor. The combination of ridge-based matching scores computed by the proposed ridge-based technique with minutia-based matching scores resulted in an improvement in the overall matching performance.

The rest of this paper is organized as follows. In Section 2, we described our ridge-based fingerprint matching technique. In Section 3, the matching performance of the proposed ridge-based fingerprint matching technique is compared with a minutia-based matcher. We also show that combining these two matchers can lead to better performance than the individual matchers. Finally, conclusions are presented and future research directions are pointed in Section 4.

2. Ridge-Based Fingerprint Matching

The proposed ridge-based technique for fingerprint matching consists of the following steps:

- The gray-scale query fingerprint image is pre-processed and converted into a thinned image where the fingerprint ridges are detected and represented by a single pixel-width;
- The ridges of the query and the template fingerprints are detected and stored in two lists of ridges, R_q and R_t, respectively;
- In order to detect only the straight lines which better match each ridge, the Hough transform is applied on each ridge separately;
- A threshold is used to detect the peaks of the Hough space for each ridge. The Hough space peaks of the query and template fingerprint are stored in two sets of peaks, S_q and S_t, respectively;
- The straight lines (Hough space peaks) detected from each ridge are used to classify it into one of five categories. The ridge category number is proportional to the ridge curvature (category 1 is for almost straight ridges, and category 5 is for almost circular ridges).
- Each element of the sets S_q and S_t is characterized by a triplet (θ_i, ρ_i, v_i), where θ_i is the orientation of the perpendicular to the *i-th* straight line, ρ_i is the distance of the *i-th* straight line to the origin, and v_i is the value of the peak p_i (the number of collinear ridge pixels that lie on that straight line);
- The query fingerprint is aligned to the template fingerprint using the rotation and translation parameters estimated from their Hough space peak sets;
- Finally, a matching score is computed for the alignment using a matrix of ridge alignments.

Figure 1 shows a diagram with the major steps of the proposed ridge-based matching technique.

2.1 Ridge Extraction

The first step in our ridge-based fingerprint matching is the extraction of the fingerprint ridges. This stage is crucial, since the success of all the subsequent steps depends on the correct ridge detection and extraction. The ridge extraction algorithm proposed by Jain et al. [3] has been used in our experiments.

The first stage of ridge extraction technique is the estimation of the orientation field of the fingerprint image. This is followed by the segmentation of the fingerprint area from the background. Next, the ridges are extracted from the input image by applying two masks that adaptively capture the maximum gray level values along the direction perpendicular to the ridge orientation. Then, several heuristics are applied to

remove holes and speckles in the binary ridge map. Finally, the extracted ridges are thinned.

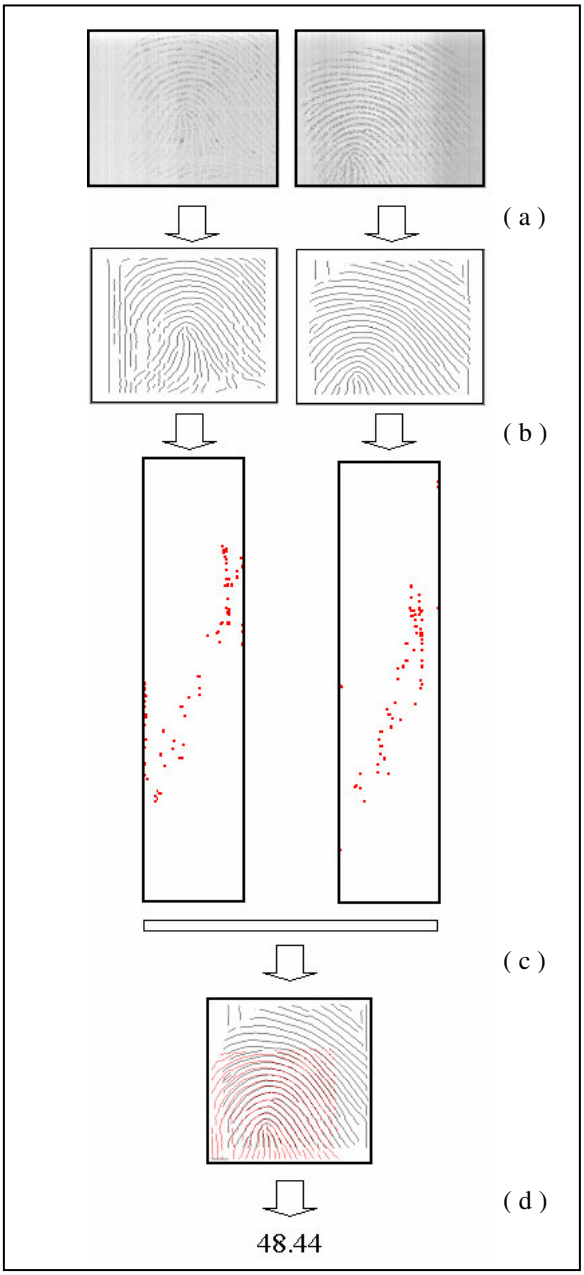


Figure 1. Major steps of the ridge-based matching technique. (a) Fingerprint ridges are detected and thinned; (b) Most significant Hough space peaks are detected; (c) The rotation and translation parameters for fingerprint registration are estimated from the two sets of peaks; (d) A correlation score is computed for the alignment.

2.2 Straight Line Extraction

The second step in our matcher is the extraction of the straight lines that approximate the fingerprint ridges. This is carried out by using the Hough transform [7]. Hough transform is a method for detecting curves in images, and in particular, can be used for straight-line detection.

Hough transform algorithm requires an accumulator array whose dimension corresponds to the number of parameters of the curve being detected. In the case of straight lines, since the equation $y=ax+b$ has unbounded parameters, the equation $\rho=x\cos\theta+y\sin\theta$ is generally used (where ρ is the perpendicular distance from the origin to the straight line and θ is the angle made by the perpendicular with the x-axis). So, for the straight-line detection, it is necessary to use a two-dimensional accumulator array. The accumulator array accumulates evidence for the existence of the straight line $\rho=x\cos\theta+y\sin\theta$ in a bin HS(R,T), where R and T are the quantized values of ρ and θ, respectively.

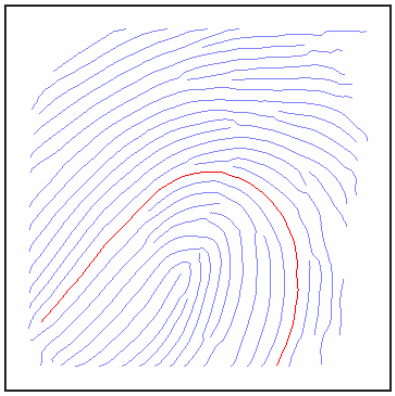
Using the accumulator array HS, the Hough procedure examines each pixel of a given ridge and increments the accumulator bins corresponding to all possible straight lines that pass through that pixel. After all the pixels of the given ridge have been processed, the accumulator array is searched for peaks. The peaks indicate the parameters of the most likely straight lines that *match that ridge* in the image. Since the ridges of a fingerprint run in parallel, in only a few directions, we expect many peaks only in a few columns of the HS.

Figure 2 shows an example of straight-line detection from a given fingerprint ridge. In this example, the given ridge is classified as category five, since the orientations of the straight lines that match it are spread in the range (0^o, 180^o).

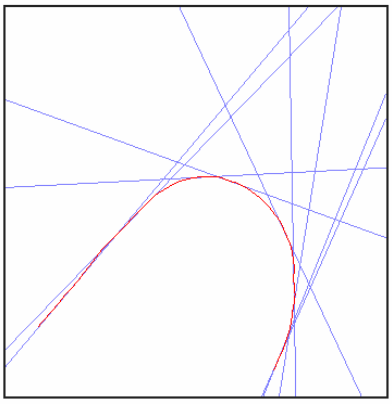
2.3 Fingerprint Registration

The third step in our algorithm is the alignment of the query fingerprint image with the template fingerprint image. The rigid transformation parameters (rotation, translation, and scale) are estimated by using the generalized Hough transform proposed by Ratha et al. [8], which is adapted to take into account the Hough space peaks obtained in the previous step.

(a)



(b)



(c)

Figure 3. (a) Fingerprint image; (b) Thinned ridges detected from the fingerprint image; (c) Straight lines detected from a given (highlighted) fingerprint ridge by using the Hough transform.

From the two Hough space peak sets, S_q and S_t, the rotation parameter can be easily estimated by using a 1-D parameter space, R. For each pair of peaks (q_i,t_i), where $q_i \in S_q$ and $t_i \in S_t$, the bin $R(\theta_{ti}-\theta_{qi})$ is incremented.

Estimating the translation parameter is more time consuming and it requires a 2-D parameter space, TR. Every pair of peaks from the set S_q, with different orientations and detected from the same ridge, is rotated by a rotation parameter estimated in the previous step. Then, the intersection point, p_q, of the two straight lines corresponding to these peaks is computed, and for every pair of peaks from the set S_t, with different orientations and detected from the same ridge, the intersection point, p_t, of the two straight lines corresponding to these peaks are also computed. The bin $TR(p_{ty}-p_{qy},p_{tx}-p_{qx})$ is then incremented by a weighted value based on the maximum straight line lengths used to find the intersection points p_q and p_t.

The parameter spaces R and TR accumulate evidences for the most likely rotation and translation parameters, respectively. For a rigid transformation, the coordinates of the maximum peak of R and TR are the most probable values for the rotation and translation parameters. In practice, as the fingerprint transformation is not rigid, a thresholding technique is used. This means that apart from the maximum peaks, all the parameter values whose evidence are greater than a given threshold value are considered as possible alignment parameters.

Since all the fingerprint images in our database were obtained using the same sensor, the scale factor of the transformation was set to 1. Figure 4 shows some examples of genuine fingerprint alignment obtained by using the described approach.

2.4 Fingerprint Matching

For each triplet of rotation and translation parameters $(\Delta_\theta, \Delta_x, \Delta_y)$ estimated in the previous stage, the query image is aligned to the template image and a matching score is calculated. The final matching score is the maximum score obtained for all triplets of transformation parameters.

In our technique, the matching score between two aligned fingerprint images is proportional to the number of matching ridges. In order to determine this number, a matrix of ridge alignments, $C_{m,n}$, is calculated, where m and n are the number of ridges detected in the query and template fingerprints, respectively.

Figure 4. Examples of genuine fingerprint alignment using the ridge-based technique. (a) Query fingerprint image; (b) Template fingerprint image; (c) Alignment obtained.

The *(i,j)-th* element of matrix *C* indicates how many pixels of the *i-th* ridge of the query fingerprint coincide with pixels of the *j-th* ridge of the template fingerprint after their alignment.

C is not symmetric, since the number of pixels from ridge *i* that align pixels from ridge *j* can be different from the number of pixels from ridge *j* that align pixels from ridge *i*.

If the query and template images are exactly the same, *C* is a diagonal matrix, where the *k-th* diagonal element is the exact number of pixels of the *k-th* fingerprint ridge.

For genuine matching, it is expected that *C* has high-valued elements located near the main diagonal. On the other hand, for impostor matching, it is expected that *C* has low-valued elements spread over of matrix.

Figures 5 and 6 show examples of matrices of ridge alignments obtained from genuine and impostor matching fingerprint pairs, respectively.

The fingerprint matching score *s* is calculated from the ridge alignment matrix *C*, according to the equation:

$$s = \frac{2\left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} C(i,j)^2\right)}{a+b} \qquad (1)$$

where n_1 and n_2 are the numbers of ridges of the query and template fingerprints, respectively, and a and b are given by:

$$a = \sum_{i=1}^{n_1} (R_q(i)_{nop})^2 \qquad (2)$$

$$b = \sum_{i=1}^{n_2} (R_t(i)_{nop})^2 \qquad (3)$$

where $R_q(i)_{nop}$ is the number of pixels of the *i-th* ridge of query fingerprint, and $R_t(i)_{nop}$ is the number of pixels of the *i-th* ridge of the template fingerprint.

Only the ridges of one fingerprint image that intercept at least one ridge of the other fingerprint image are considered in the computation of *a* and *b*.

In order to penalize ridge crossings, which are very frequent in impostor alignments, the rows and columns of *C* that have more than *n* non-zero values are all discarded during the calculation of the score *s*. On the other hand, in order to emphasize the genuine alignments, the *C(i,j)* values corresponding to ridges i and j of the same category are increased. The increasing amount is proportional to the ridges category number.

3. Experimental Results

The matching performance of the proposed ridge-based technique is evaluated on the MSU-VERIDICOM fingerprint database. This database consists of fingerprint impressions of 160 users obtained using a Veridicom solid-state sensor. Each

user provided four impressions of each of the four fingers: the left index, the left middle, the right index, and the right middle. The results reported in this paper are based only on the four impressions of the right index finger.

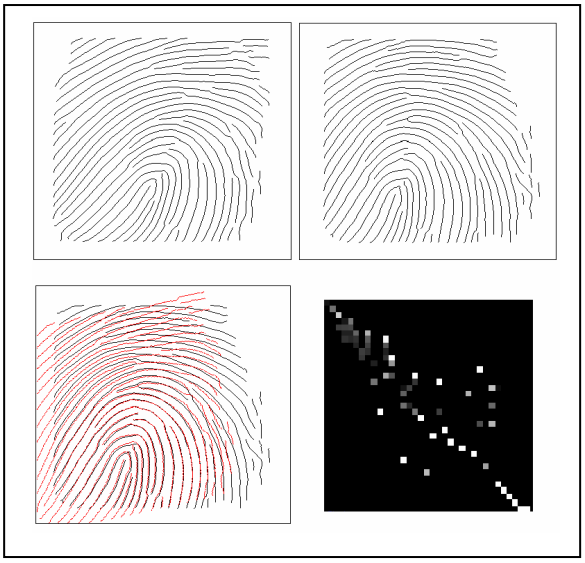


Figure 5. Matrix of ridge alignments obtained from a genuine fingerprint pair. The matrix has mostly high-valued peaks spread along the diagonal.

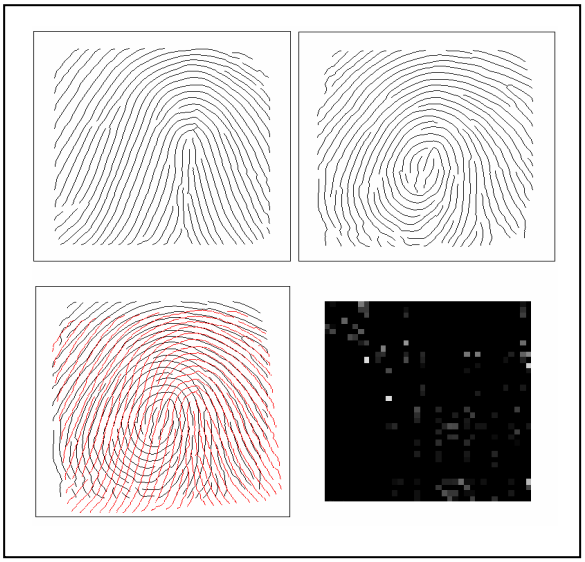


Figure 6. Matrix of ridge alignments obtained from an impostor fingerprint pair. The matrix has mostly low-valued peaks spread over the matrix.

The fingerprint verification performance of the ridge-based matcher was compared with the performance of the 2D dynamic programming based minutiae matcher proposed by Jain et al. [9]. The performance of a hybrid matcher obtained by combining the two minutiae and ridge-based matchers by using the max rule and the min-max normalization [10] is also presented.

Table 1 shows the equal error rates (EER), and the genuine accept rates (GAR) at two different values (0.1% and 1%) of false-accept rates (FAR), obtained by the three matchers. Figure 7 shows the Receiver Operating Characteristic (ROC) curves for the three matchers.

As can be observed in Table 1 and Figure 7, the best matcher is the hybrid matcher, which has a GAR of 93.38% at a 0.1% FAR. While the minutiae-based approach has a GAR of 91.69% at the same value of FAR, the ridge-based approach obtained a corresponding GAR of only 66.87%.

Figure 7 also shows that the ridge-based matcher gave the worst performance. This was not a surprise, since the higher distinctiveness of the minutiae compared with the ridges is well known. Despite the success of the ridge-based approach for genuine fingerprint alignment (as can be observed in Figure 4), for a considerable number of impostor fingerprint alignments, the matching scores obtained with the ridge-based matcher are quite high, leading to a low performance. Figure 8 shows some examples of impostor fingerprint alignments obtained with the ridge-based approach that had high matching scores.

On the other hand, the ridge-based matcher correctly accepted some genuine fingerprint pairs rejected by the minutia-based matcher. Figure 9 shows an example of a genuine fingerprint pair rejected by the minutia-based matcher and accepted by the ridge-based matcher. The minutia-based matcher failed in this case because there are very few corresponding minutia in the query and template fingerprint and this leads to a low matching score.

In summary, despite the lower distinctive power of the ridges, ridge features combined with minutia features can lead to more accurate fingerprint matchers, especially when the images are of poor-quality, or have very few overlapping minutiae, as in the case of images captured by small solid-state sensors.

Table 1. Error rates of the three fingerprint matchers.

Matcher	EER	GAR (at 0.1% FAR)	GAR (at 1% FAR)
Minutia-based	3.53%	91.69 %	95.49 %
Ridge-based	8.25%	66.87 %	79.68 %
Hybrid (Minutiae-Ridge)	3.03%	93.38 %	96.08 %

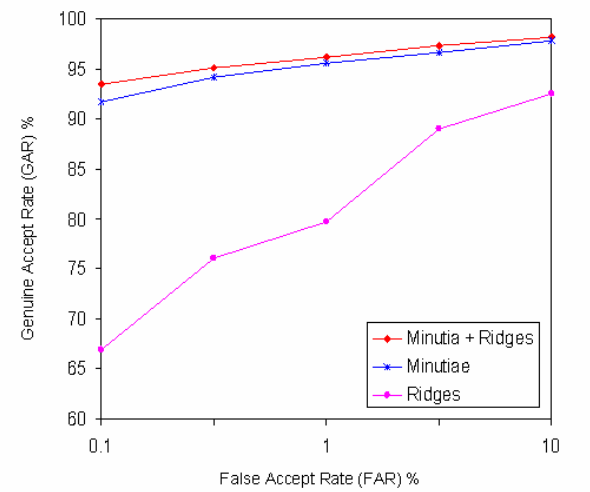


Figure 7. Receiver Operating Characteristic curves for the three fingerprint matchers.

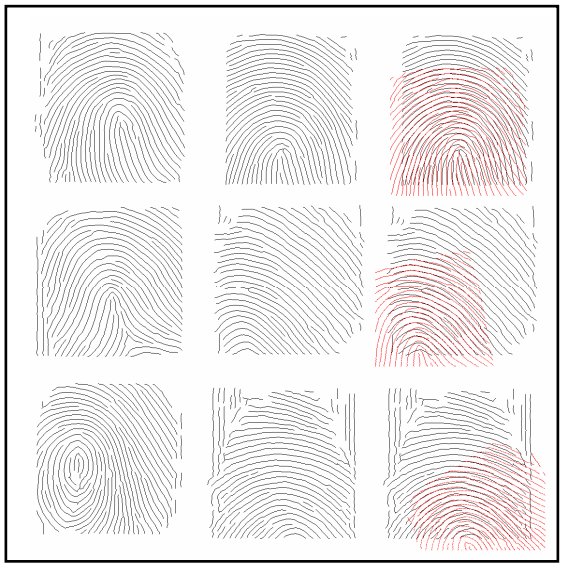


Figure 8. Examples of three impostor fingerprint alignments that resulted in high scores with the ridge-based approach.

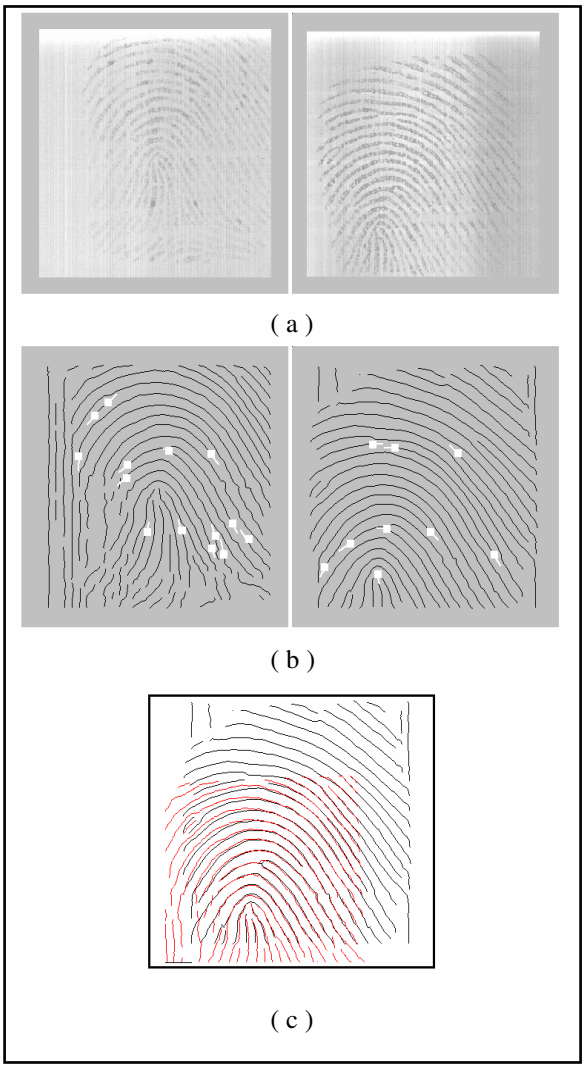


Figure 9. (a) A genuine fingerprint match rejected by the minutia-based matcher and correctly accepted by the ridge-based matcher; (b) Minutia points detected; (c) Alignment obtained by the ridge-based technique.

4. Conclusions and Future Work

We have presented a fingerprint matching technique that uses ridge features to align and match fingerprints. Straight lines that approximate each fingerprint ridge are separately extracted using the Hough transform. All detected Hough space peaks are then used to estimate the rigid transformation parameters between the query and the template fingerprint images. After the alignment, a matching score is computed from a matrix of ridge alignments.

Despite the good performance of the proposed technique for genuine fingerprint alignments, it also gives high matching scores for a number of impostor pairs, leading to a lower performance compared with the minutia-based approach. On the other hand, some genuine fingerprint pairs, which were rejected by the minutia-based matcher, are correctly aligned and accepted by the ridge-based matcher.

These results allow us to conclude that combining additional features like the ridge features with minutiae is a promising approach to decrease the error rates of fingerprint matchers, particularly for images of poor quality or images captured by small solid-state sensors.

In order to increase the genuine fingerprint alignment scores and decrease the false fingerprint alignment scores, we are investigating the use of inter-ridge distance in our matching algorithm. Regarding the processing time, the current algorithm takes around 8 seconds to provide a matching score. We are also working on the development of a faster matching algorithm.

Acknowledgements

Aparecido Nilceu Marana would like to thank Fapesp (proc. 04/00551-8) and Capes (proc. BEX 0126-04-7) for the financial support.

The authors would like to thank Karthik Nandakumar, Yi Chen, Dirk Colbry, and Francesc J. Ferri, for their valuable suggestions and comments.

References

[1] Maltoni, D., D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, Springer-Verlag, New York, 2003.

[2] Stosz, J. D., and L. A. Alyea, "Automated System for Fingerprint Authentication Using Pores and Ridge Structure", *Proc. of SPIE (Automatic Systems for the Identification and Inspection of Human)*, vol. 2277, 1994, pp. 210-223.

[3] Jain, A. K., L. Hong, and R. Bolle, "On-line Fingerprint Verification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19(4), 1997, pp. 302-314.

[4] Jain, A. K., A. Ross, and S. Prabahkar, "Fingerprint Matching Using Minutiae and Texture Features", *Proceedings of Int. Conference on Image Processing (ICIP)*, Thessaloniki, Greece, Oct 7-10, 2001, pp. 282-285.

[5] Ross, A., A. K. Jain, and J. Reisman, "A Hybrid Fingerprint Matcher", *Pattern Recognition*, vol. 36(7), 2003, pp. 1661-1673.

[6] Nandakumar, K., and A. K. Jain, "Local Correlation-Based Fingerprint Matching", *Proceedings of the ICVGIP*, Kolkata, India, December 2004.

[7] Shapiro, L. G., and G. Stockman, *Computer Vision*, Prentice-Hall, New Jersey, 2001.

[8] Ratha, N. K., K. Karu, S. Chen, and A. K. Jain, "A Real-Time Matching System for Large Fingerprint Database", *IEEE Trans. PAMI*, 18(8), 1996, pp. 799-813.

[9] Jain, A. K., S. Prabhakar, and S. Chen, "Combining Multiple Matchers for a High Security Fingerprint Verification System", *Pattern Recognition Letters*, 20, No. 11-13, 1999, pp. 1371-1379.

[10] Snelick, R., M. Indovina, J., J. Yen, and A. Mink, "Multimodal Biometrics: Issues in Design and Testing", *Proc. of Fifth International Conference on Multimodal Interfaces*, Vancouver, Canada, November 2003, pp. 68-72. 1859-1867.