A Fingerprint Retrieval Technique using Fuzzy Logic-based Rules

Rosario Arjona and Iluminada Baturone

Microelectronics Institute of Seville (IMSE-CNM),
University of Seville and Spanish National Research Council (CSIC), Seville, Spain
{arjona,lumi}@imse-cnm.csic.es

Abstract. This paper proposes a global fingerprint feature named QFingerMap that provides fuzzy information about a fingerprint image. A fuzzy rule that combines information from several QFingerMaps is employed to register an individual in a database. Error and penetration rates of a fuzzy retrieval system based on those rules are similar to other systems reported in the literature that are also based on global features. However, the proposed system can be implemented in hardware platforms of very much lower computational resources, offering even lower processing time.

Keywords: Fingerprint retrieval, Fuzzy rules, Low computational cost

1 Introduction

A fingerprint identification system requires to compare the query fingerprint against all the fingerprints registered in a database. This operation can be very time consuming if the comparison process is complex and the database is large (several millions of fingerprints can be registered in forensic and government applications) [1]. The objective of retrieval techniques is to apply a complex comparison process to a small number of registered individuals instead of considering the whole database.

Several retrieval techniques have been proposed in the literature. They can be grouped into techniques based on *Exclusive Classification* and based on *Continuous Classification*. In the *Exclusive Classification*, fingerprints are grouped into pre-defined disjoint classes (each fingerprint is associated to one class). The most common fingerprint classification, which was proposed in [2] and extended in [3], distinguishes five fingerprint classes (arch, whorl, tended arch, left loop, and right loop). The problem is that most of fingerprints are only distributed into three classes (right loop, left loop, and whorl) and the number of comparisons are not reduced enough for a large fingerprint database [1]. In addition, determining the correspondence between a fingerprint and a class is usually a fuzzy and ambiguous operation, even for a human. *Continuous Classification* is more suitable to cope with such fuzziness. It consists of two phases: (1) in the indexing phase, a numeric vector (an index) is stored in the database to register a fingerprint; (2) in the retrieving phase, a list of M candidates are selected

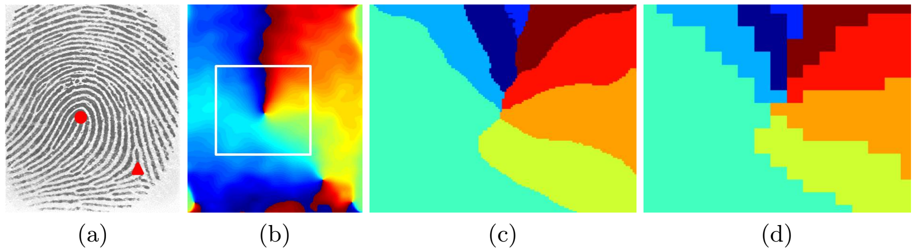


Fig. 1. Features extracted from a fingerprint image: (a) Singular points, (b) Directional image, (c) Segmented directional image of the window depicted in (b), and (d) QFingerMap.

among the N stored candidates based on a simple comparison process between the N stored indices and the index of the query fingerprint [4].

Fingerprints, which are represented by ridges (defined by black colors in Fig. 1(a)) and valleys (defined by bright colors in Fig. 1(a)), are captured with a high variability because individuals do not always place their fingers on the sensors in the same way. Hence, the extraction of distinctive as well as simple indices to be stored and compared is not an easy task. Local features such as the traditional minutiae (which mean small details) require a detailed analysis of the fingerprint image to detect endings (ridges which end) and bifurcations (ridges which are divided into two ridges). In contrast, global features require a coarser analysis. An example is the directional image (also known as orientation image, field or map, or directional field or map), which contains the ridge orientations at the pixels (orientations are represented by colors in Fig. 1(b)). Other global features are the singular points (depicted in Fig. 1(a)): cores which are the points where ridges converge (represented by a circle in Fig. 1(a)) or deltas which are the points where ridges diverge (represented by a triangle in Fig. 1(a)).

Minutiae give high accuracy for identification purposes [5] [6] [7], but the extraction process of minutiae is complex. Fuzzy rule-based systems have been proposed for minutiae extraction [8], selection of the optimal set of minutiae [7], and minutia-based fingerprint matching [9]. For a retrieval application, whose objective is to find a small number of candidates with very low effort, global features offer acceptable recognition results with low computational complexity [10] [11] [12] [13]. Fuzzy rule-based systems have also been employed for directional image description [14], and to classify fingerprints based on textures [15].

This paper proposes a new global feature named QFingerMap that provides fuzzy information of the fingerprint image. Instead of registering the finger of an individual by a crisp index, a finger is registered by a fuzzy rule whose antecedent part considers the QFingerMaps that can be extracted from it. Given a query finger, several rules will be activated at certain degree. A list of possible candidates can be obtained from the activation degree of the rules.

The paper is organized as follows. Section 2 describes the extraction of the feature QFingerMap and the fuzzy rule base that employs it. Section 3 summarizes the design decisions taken to extract QFingerMaps and evaluates the performance of the rule base with two fingerprint databases. Results are compared to other approaches proposed in literature. Finally, Section 4 gives conclusions.

2 Fuzzy Retrieval System

2.1 Fuzzy Fingerprint Feature

As commented in Introduction, the directional image contains information about the tangent directions, $D(i,j)$, to the ridges at the pixels of the fingerprint. Horizontal, G_x, and vertical, G_y, gradients can be computed by using different filters such as Sobel, Gaussian, or Prewitt. The values of G_x and G_y are usually obtained after the convolution of a window centered at each pixel with the horizontal and vertical filter matrix. The fingerprint image should be enhanced, and gradient values should be combined, averaged, and smoothed in order to remove incorrect directions because the computation of the directional image is sensitive to noise in fingerprint images. The method reported in [16], and implemented in [17], can be summarized as follows:

$$D(i,j) = \frac{\pi}{2} + \frac{\arctan\left(\frac{sin2theta(i,j)}{cos2theta(i,j)}\right)}{2} \tag{1}$$

where:

$$sin2theta(i,j) = \frac{G_{xy}(i,j)}{\sqrt{G_{xy}(i,j)^2 + (G_{xx}(i,j) - G_{xy}(i,j))^2}} \tag{2}$$

$$cos2theta(i,j) = \frac{G_{xx}(i,j) - G_{xy}(i,j)}{\sqrt{G_{xy}(i,j)^2 + (G_{xx}(i,j) - G_{xy}(i,j))^2}} \tag{3}$$

$$\begin{aligned} G_{xx}(i,j) = G_x(i,j) \cdot G_x(i,j)\,; G_{xy}(i,j) = G_x(i,j) \cdot G_y(i,j)\,; \\ G_{yy}(i,j) = G_y(i,j) \cdot G_y(i,j) \end{aligned} \tag{4}$$

The direction values are obtained after applying three Gaussian filters to the computation of: (1) gradient values (G_x and G_y); (2) covariance data of the image gradients (G_{xx}, G_{xy} and G_{yy}); and (3) sine and cosine of the double angles (*sin2theta* and *cos2theta*).

Since most of discrimination information in the directional image is around the convex core, let us consider a window of BxC pixels centered at the convex core (as depicted in Fig. 1(b)). If such a window is employed to extract a global feature and each direction value is encoded with 8 bits, the number of bits to store per fingerprint is BxCx8. An study was carried out to simplify not only the number of bits to store (in order to reduce memory requirements of the retrieval system) but also the computational cost of extracting the fingerprint feature (in order to reduce processing time and/or hardware cost).

The first simplification considered to reduce memory is to cluster the continuous direction values in the range [0°, 180°] into several representative directions. For example, if 8 representative directions are considered, the result is that the number of bits to store is reduced to BxCx3 (plus 8x8 to store the values of the representative directions). Clustering techniques have been employed to classify fingerprints [18]. However, those clustering techniques, which are based on a genetic algorithm, increase the processing time considerably. The clustering algorithm considered in our case is K-means, which finds K representative directions, $V = (\boldsymbol{v_1}, \boldsymbol{v_2}, ..., \boldsymbol{v_K})$, among the BxC directions, by minimizing the following target function.

$$J\,[V; X] = \sum_{i=1}^{K} \sum_{j=1}^{BxC} d(\boldsymbol{x_j}, \boldsymbol{v_i}) \tag{5}$$

where $d(\boldsymbol{x_j}, \boldsymbol{v_i})$ is the distance between the directions and the representative directions (or prototypes).

The K-means algorithm with different values of K was applied to directional images of the fingerprints of two databases. One of them is the public and standard database FVC 2000 DB2a [19], with 800 fingerprints (from 100 fingers and 8 samples from each finger) captured by a capacitive sensor. The other database was created by the authors for on-line recognition. It consists of 560 fingerprints (from 112 fingers and 5 samples from each finger) captured by an optical sensor. An interesting result is that the prototypes found for each fingerprint are, in general, different from the representative directions of another fingerprint. However, the values are similar. For example, for K=8, the mean values of the representative directions found in the fingerprints of the FVC 2000 DB2a were 13.37°, 35.15°, 51.65°, 68.57°, 84.98°, 101.48°, 117.71°, and 139.53°, while they were 15.20°, 39.33°, 56.94°, 76.19°, 97.24°, 117.89°, 138.74°, and 164.26° for the fingerprints of the on-line database. Since the mean values were very similar for many fingerprints, the second simplification considered was to employ a set of K fixed and equispaced prototypes. For example, for K=8, the following prototypes are considered: 11.25°, 33.75°, 56.25°, 78.75°, 101.25°, 123.75°,

IF	THEN
$(\lvert G_x\rvert = 0\ OR\ \lvert G_y\rvert \geq 2.413 \cdot \lvert G_x\rvert)\ AND\ G_x \cdot G_y \geq 0$	Symbol is 0
$(\lvert G_x\rvert = 0\ OR\ \lvert G_y\rvert \geq 2.413 \cdot \lvert G_x\rvert)\ AND\ G_x \cdot G_y < 0$	Symbol is 1
$(\lvert G_x\rvert = 0\ OR\ \lvert G_y\rvert < 0.414 \cdot \lvert G_x\rvert)\ AND\ G_x \cdot G_y \geq 0$	Symbol is 2
$(\lvert G_x\rvert = 0\ OR\ \lvert G_y\rvert < 0.414 \cdot \lvert G_x\rvert)\ AND\ G_x \cdot G_y < 0$	Symbol is 3
$(\lvert G_y\rvert < \lvert G_x\rvert\ AND\ \lvert G_y\rvert \geq 0.414 \cdot \lvert G_x\rvert)\ AND\ G_x \cdot G_y \geq 0$	Symbol is 4
$(\lvert G_y\rvert < \lvert G_x\rvert\ AND\ \lvert G_y\rvert \geq 0.414 \cdot \lvert G_x\rvert)\ AND\ G_x \cdot G_y < 0$	Symbol is 5
$(\lvert G_y\rvert \geq \lvert G_x\rvert\ AND\ \lvert G_y\rvert < 2.413 \cdot \lvert G_x\rvert)\ AND\ G_x \cdot G_y \geq 0$	Symbol is 6
$(\lvert G_y\rvert \geq \lvert G_x\rvert\ AND\ \lvert G_y\rvert < 2.413 \cdot \lvert G_x\rvert)\ AND\ G_x \cdot G_y < 0$	Symbol is 7

Table 1. Processing to obtain symbols from gradient values.

146.25°, and 168.75°. Each prototype can be represented by a symbol from 0 to 7, encoded with 3 bits.

The main advantage of the second simplification is the high reduction of computational cost. Not only the prototypes should not be computed (using Equation (5)) but also no trigonometric, powering or square-rooting operations are needed. Once gradient values, G_x and G_y, are obtained, they are compared to determine directly which symbol is associated to each pixel. Hence, complex operations and subsequent clustering is reduced to simple comparisons between gradient values as shown in Table 1.

As in any technique that calculates directional images, smoothing process is also required to obtain homogeneous direction regions. Among the wide set of filters that can be used to perform smoothing [1], a non linear filter based on maximum operator has been selected. It considers the neighboring pixels inside a SxS window centered at the analyzed pixel and assigns it the symbol value with the highest number of occurrences inside the window. The result after smoothing with a 27x27 window is shown in Fig. 1(c) (each representative direction is defined by a color).

The third simplification considered to reduce memory is to apply downsampling in order to remove possible redundant information. A simple way is to take 1 between n consecutive pixels (downsampling by a factor of n), being the pixels swept in the BxC window of the segmented directional image from left to right and from up to bottom. Fig. 1(d) shows the result after applying a downsampling factor of 8.

The simplifications considered result in a fuzzy fingerprint feature that is advantageous in terms of memory required to be stored as well as processing time to be extracted. This feature is named QFingerMap because it is a map of directions whose extraction process is quick. Table 2 shows a comparison in terms of memory and processing time between the initial and final fingerprint features commented above. The results in the first row correspond to a global feature obtained from a 129x129 window centered at the convex core and formed by the direction values calculated as in Equations 1 to 4 and post-processed as done in [17]. The results in the second row correspond to a QFingerMap obtained from a 129x129 window, using 8 representative directions, and a downsampling factor of 8 (as shown in Fig. 1(d)). While the directional image was extracted using a platform with an Intel i7 processor running at 3.20 GHz, the QFingerMap was extracted using dedicated hardware in a Field Programmable Gate Array (FPGA) running at 25 MHz. The gain in memory and time does not mean a loss of distinctiveness, as will be described in Section 3.

2.2 Fuzzy Rule Base

A QFingerMap is invariant to translations because it is centered at the convex core but it changes if the finger is placed on the sensor with a different orientation. Fig. 2 illustrates examples of QFingerMaps extracted from fingerprints from the same and different fingers. Hence, several QFingerMaps should be extracted from the same finger to take into account the variability of the captures.

Fingerprint Feature	Platform	Memory (bits)	Time (ms)	Gain in Memory	Gain in Time
Directional Image	Intel i7 @ 3.20 GHz	129x129x8	240	x1	x1
QFingerMap	Virtex 6 FPGA @ 25 MHz	17x17x3	23	x154	x10

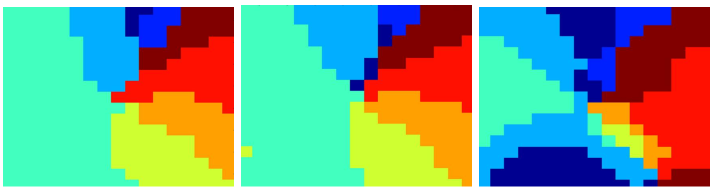
Table 2. Comparison between features based on directional image and QFingerMap.

IF	THEN
$QFM^{'}$ is QFM_{11} OR ... OR $QFM^{'}$ is QFM_{1R}	Individual 1
...	...
$QFM^{'}$ is QFM_{N1} OR ... OR $QFM^{'}$ is QFN_{NR}	Individual N

Table 3. Classification fuzzy rules for the retrieval system.

If R is the number of captures considered, QFM_{t1}, ..., QFM_{tR} are extracted to register the finger of the t-th individual. An intuitive way to evaluate if a query QFingerMap, *QFM'*, corresponds to a registered individual is to apply the if-then rules in Table 3. The if-then rules can be seen as classification rules based on matching fuzzy patterns [20] where the fuzzy patterns are the QFingerMaps. The rules' antecedents used in the fuzzy system employed with the FVC 2000 DB2a combines three QFingerMaps for each individual. If one fingerprint image is captured from each individual in the enrollment process, the first QFingerMap in the rule's antecedent is extracted from that image, the second QFingerMap is extracted from that image rotated 11.25° clockwise, and the third QFingerMap is extracted from the image rotated 11.25° counterclockwise. The rules' antecedents used in the on-line database only employs one QFingerMap for each fingerprint image (because the sensor employed forced the user to introduce the finger always with the same orientation).

Since a QFingerMap is a fuzzy fingerprint feature, if the query QFingerMap has been extracted from the finger of the t-th individual, *QFM'* should be similar to at least one of the QFM_{t1}, ..., QFM_{tR}, but, surely, it will not be identical to any of them. The similarity between two QFingerMaps can be evaluated by



(a) (b) (c)

Fig. 2. (a) and (b) QFingerMaps of fingerprints from the same finger but different captures, and (c) from a different finger.

using many measures. In our case, since the retrieval process should be fast, the following simple measure has been selected.

$$similarity(QFM', QFM) = 1 - \frac{1}{W}\sum_{i=1}^{W} d_i(QFM', QFM) \tag{6}$$

where $QFM' = (v_1^{'}, ..., v_W^{'})$, $QFM = (v_1, ..., v_W)$, and

$$d_i(QFM', QFM) = \begin{cases} 1 \text{ if } v_i^{'} \neq v_i \\ \\ 0 \text{ otherwise} \end{cases} \tag{7}$$

Similarity ranges from 0, which means that all the symbols (or representative directions, v_i and $v_i^{'}$) assigned to the pixels in the same location are different, to 1, which means that all the symbols are the same. For example, Fig. 3 shows the similarities between a query QFingerMap and the QFingerMaps in two rules' antecedents.

The rules' antecedents combine the similarity degrees between QFingerMaps by a disjunctive conjunction, *OR*. As usual in fuzzy rule bases, the s-norm maximum has been selected as *OR* operator. Hence, the activation degree of each rule is computed as follows.

$$\begin{aligned} activation\_degree_{rule_t} = \\ max_{r=1,...,R}\left\{similarity(QFM^{'}, QFM_{tr})\right\} \end{aligned} \tag{8}$$

The conclusion provided by the rule base can be an individual or a set of candidates (M). In the first case, the consequence (the individual) of the most activated rule is selected. In the second case, a set of M individuals are given, each of them with the certainty of being the true candidate given by the activation degree of its corresponding rule, as illustrated in Fig. 3.

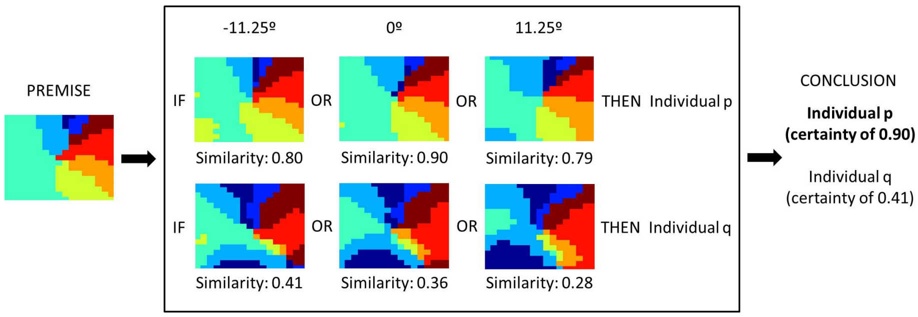
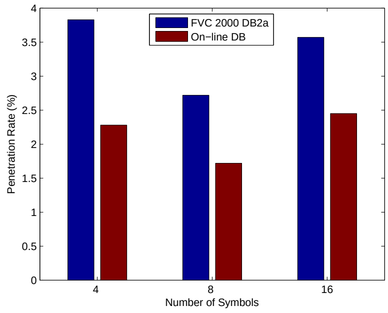
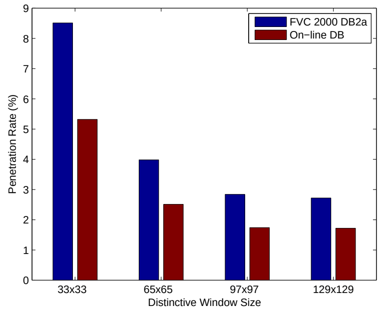


Fig. 3. Example of inference with two fuzzy rules.

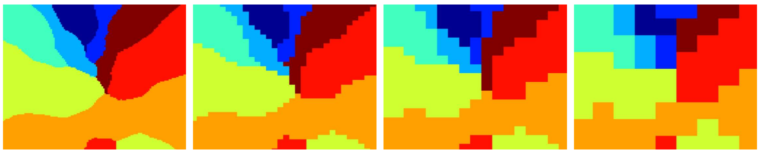


(a)



(b)

Fig. 4. (a) Influence of number of symbols and (b) distinctive window size of QFingerMaps in the Penetration Rate.



(a) (b) (c) (d)

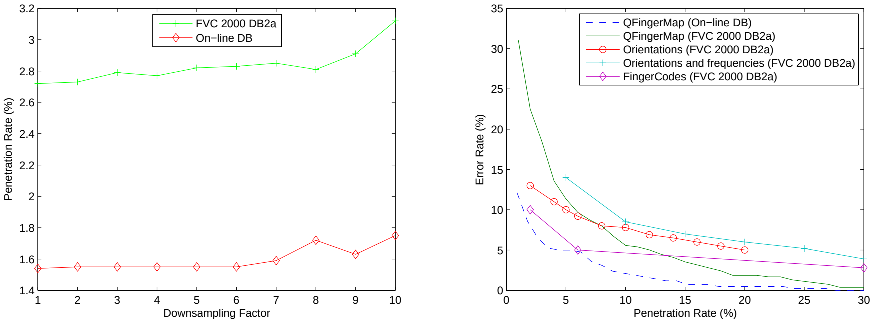
Fig. 5. (a)-(d) Influence of several downsampling factors in the extraction of a QFingerMap: (a) 1, (b) 4, (c) 8, and (d) 16.

3 Design and Evaluation of the Fuzzy Retrieval System

The design decisions taken to extract the QFingerMaps and define the rules were based on evaluating the main performance indicators of a retrieval system. Performance indicators depend on the indexing scenario. In a *Non Incremental Search* scenario (where the candidate list is truncated to M) there is a trade-off between *Error Rate* and *Penetration Rate*. *Error Rate* is the percentage of searched fingerprints (rules in this case) whose mate is not present in the candidate list and *Penetration Rate* is the portion of the rule base that the system

Feature	**Time (ms)**	**Platform**
QFingerMap	23	Virtex 6 FPGA @ 25 MHz
Orientations and Frequencies [12]	67	Intel Pentium 4 @ 2.26 GHz
Orientations and Frequencies [10]	1.6	Intel Core 2 Quad @ 2.26 GHz
Minutiae [6]	1400	Intel Xeon @ 1.7 GHz (PCI @ 33 MHz)
Minutiae Triplets [5]	1000	Sun Ultra 2 @ 143 MHz
Minutiae Cylinder Code [21]	90000	Intel Pentium 4 @ 2.8 GHz

Table 4. Time to search among 2000 individuals.



(a) (b)

Fig. 6. (a) Penetration Rate depending on downsampling factor. (b) Tradeoff between Penetration Rate and Error Rate for different fingerprint features.

has to search on the average (M/N). The objective is a low *Error Rate* with a low *Penetration Rate* but the problem is that *Error Rate* increases as M is smaller. In an *Incremental Search* scenario, there are not retrieval errors because the candidate list is not truncated. The search finishes as soon as the true mate is retrieved. The worst case is when the corresponding mate is in the rule with the lowest activation degree. For an *Incremental Search* scenario, the only performance indicator is *Penetration Rate* [4].

The recognition and discrimination capability offered by QFingerMaps depend on the number of symbols considered. Fig. 4(a) shows the Penetration Rates obtained for different number of symbols (representative directions) and fingerprint databases. 4, 8 and 16 symbols were studied. 4 symbols offered limited information to distinguish individuals. 8 symbols offered finer information. 16 symbols did not imply to improve the discrimination capability because they increased intra-class variations, that is, representations of captures from the same finger were more different. Hence, 8 symbols were selected as the best option.

The resulting image after the symbol assignation should be smoothed to remove isolated and noisy direction values. Several smoothing window sizes were analyzed: 3x3 smoothing, 9x9 smoothing, and 27x27 smoothing. The last smoothing was selected because a 27x27 window provided good performance for the different types of sensors analyzed.

The influence of the size of the singular area (BxC) in the Penetration Rates was also analyzed. The best results were provided by a 129x129 window as illustrated in Fig. 4(b). Smaller window sizes did not contain information enough while sizes larger than 129x129 could not be analyzed in many fingerprint images. For example, enlarging the window size from 129x129 to 257x257 implies increasing the number of fingerprint images with uncompleted windows from 18.6% to 100% in the FVC 2000 DB2a database, and from 4.1% to 48.2% in the fingerprint database created for on-line recognition. For most fingerprint sensors, which capture a fingerprint size of approximately, 300x300 pixels, 129x129 is a suitable size.

Fig. 5 shows different QFingerMaps resulting from different downsampling factors applied to a 129x129 window. If the downsampling factor is high, discriminative information is removed. This is illustrated in terms of Penetration Rate in Fig. 6(a). A downsampling factor of 8 was selected, which means a QFingerMap with 17x17 symbols. If the symbols are encoded with 3 bits then the 17x17 symbol vector requires 867 bits (17x17x3 bits), as shown in Table 2.

According to a non incremental search scenario, the tradeoff between Error Rate and Penetration Rate was evaluated. Fig. 6(b) shows the results obtained with the two databases considered and compares the results with other approaches reported in the literature [11] [12] [13]. The designed fuzzy retrieval system offers competitive performance.

The first row in Table 4 shows the time to search among 2000 individuals using the proposed technique implemented in a FPGA working at 25 MHz. The other rows in Table 4 show the time reported in the literature using other techniques based on both global and local features, implemented in other hardware platforms.

4 Conclusions

Despite the simplicity of the fingerprint feature proposed, processing of fuzzy information by using fuzzy rules is able to find the individual in a database with competitive error and penetration rates. This has been analyzed with two fingerprint databases. The fuzzy retrieval system implemented in a Virtex 6 FPGA provides low processing time even working at low frequency.

Acknowledgments. This work has been partially supported by TEC2011-24319 and IPT-2012-0695-390000 projects from the Ministry of Economy and Competitiveness of the Spanish Government (with support from the PO FEDER-FSE). The work of R. Arjona has been supported by a Post-Doc Fellowship from the Regional Government of Andalusia.

References

1. Maltoni, D., Maio, D., Jain, A. K., Prabhakar, S.: Handbook of Fingerprint Recognition. Springer-Verlag (2009)
2. Galton, F.: Finger Prints. Macmillan and Co. (1892)
3. Henry, E.: Classification and Uses of Finger Prints. George Routledge and Sons (1892)
4. Cappelli, R., Ferrara, M.: A Fingerprint Retrieval System based on Level-1 and Level-2 Features. Expert Systems with Applications: An International Journal, vol. 38, issue 12, pp. 10465–10478. Elsevier (2012)
5. Bhanu, B., Tan, X.: Fingerprint Indexing based on Novel Features of Minutiae Triplets. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, issue 5, pp. 616–622 (2003)

6. Chung, Y., Kim, K., Kim, M., Pan, S., Park, N.: A Hardware Implementation for Fingerprint Retrieval. In: Proceedings of the 9th International Conference on Knowledge-based Intelligent Information and Engineering Systems (KES 2005). Lecture Notes in Computer Science (LNCS), vol. 3683, pp. 374–380 (2005)
7. Iancu, I., Constantinescu, N.: Intuitionistic Fuzzy System for Fingerprints Authentication. Applied Soft Computing, vol. 13, pp. 2136–2142. Elsevier (2013)
8. Sagai, V. K., Koh Jit Beng, A.: Fingerprint Feature Extraction by Fuzzy Logic and Neural Networks. In: 6th International Conference on Neural Information Processing (ICONIP 99), vol. 3, pp. 1138–1142. IEEE Press, New York (1999)
9. Chen, X., Tian, J., Yang, X.: A New Algorithm for Distorted Fingerprints Matching Based on Normalized Fuzzy Similarity Measure. IEEE Transactions on Image Processing, vol. 15, issue 3, pp. 767–776. IEEE Press (2006)
10. Cappelli, R.: Fast and Accurate Fingerprint Indexing based on Ridge Orientation and Frequency. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 41, issue 6, pp. 1511-1521 (2011)
11. de Boer, J., Bazen, A. M., Gerez, S. H.: Indexing Fingerprint Databases based on Multiple Features. In: Proceedings of the 12th Annual Workshop on Circuits, Systems and Signal Processing Workshop (ProRISC 2001), pp. 300–306 (2001)
12. Jiang, X., Liu, M., Kot, A. C.: Fingerprint Retrieval for Identification. IEEE Transactions on Information Forensics and Security, vol. 1, issue 4, pp. 532–542 (2006)
13. Leung, K. C., Leung, C. H.: Improvement of Fingerprint Retrieval by Statistical Classifier. IEEE Transactions on Information Forensics and Security, vol. 6, issue 1, pp. 59–69 (2011)
14. Arjona, R., Gersnoviez, A., Baturone, I.: Fuzzy Models for Fingerprint Description. Fuzzy Logic and Applications, Lecture Notes in Computer Science, vol. 6857, pp. 228–235. Springer (2011)
15. Galar, M., Sanz, J., Pagola, M., Bustince, H., Herrera, F.: A Preliminary Study on Fingerprint Classification Using Fuzzy Rule-based Classification Systems. In: Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 554–560 (2014)
16. Bazen, A. M., Gerez, S. H.: Systematic Methods for the Computation of the Directional Fields and Singular Points of Fingerprints. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, issue 7, pp. 905–919 (2002)
17. Software Implementation for Directional Image Extraction (2011): `http://www.csse.uwa.edu.au/~pk/research/matlabfns/FingerPrints/ridgeorient.m`
18. Cappelli, R., Lumini, A., Maio, D., Maltoni, D.: Fingerprint Classification by Directional Image Partitioning. IEEE Transactions On Pattern Analysis and Machine Intelligence, vol. 21, issue 5, pp. 402–421 (1999)
19. Fingerprint Verification Competition-onGoing (2013): `https://biolab.csr.unibo.it/fvcongoing/UI/Form/Home.aspx`
20. Pedrycz, W.: Fuzzy Sets in Pattern Recognition: Methodology and Methods. Pattern Recognition, vol. 23, issue 1/2, pp. 121–146 (1990)
21. Cappelli, R., Ferrara, M., Maltoni, D.: Minutia Cylinder-Code: A New Representation and Matching Technique for Fingerprint Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, pp. 2128–2141 (2010)