FINGERPRINT VERIFICATION BASED ON IMAGE PROCESSING SEGMENTATION USING AN ONION ALGORITHM OF COMPUTATIONAL GEOMETRY *

M. POULOS
Dept. of Informatics University of Piraeus,
P.O. BOX 96,
49100 Corfu, Greece
E-mail: marios.p@usa.net

A. EVANGELOU
Dept. of Exp. Physiology University of Ioannina,
P.O. BOX 1186,
45110 Ioannina, Greece
E-mail: evagel@uoi.gr

E. MAGKOS
Dept. of Archives and Library Sciences,
Palea Anaktora
49100 Corfu, Greece
E-mail: emagos@unipi.gr

S. PAPAVLASOPOULOS
Dept. of Archives and Library Sciences,
Palea Anaktora
49100 Corfu, Greece
E-mail: sozon@ionio.gr

In this study, we applied a digital image processing system using the onion algorithm of Computational geometry to develop fingerprint verification. This method may be characterized as an alternative method to the used minutiae extraction algorithm proposed by Ratha et al. The proposed algorithm is also compared to a well-known commercial verification algorithm that is based on Ratha's algorithm. In the experimental part the results of the above comparison showed that the proposed method yields correct positive and correct negative verification scores greater than 99%.

*This work is supported by University of Piraeus.

1. INTRODUCTION

In this paper the problem of fingerprint verification via the Internet is investigated. Specifically, the method that is used for the above purpose is based on a traditional finger scanning technique, involving the analysis of small unique marks of the finger image known as minutiae. Minutiae points are the ridge endings or bifurcations branches of the finger image. The relative position of these minutiae is used for comparison, and according to empirical studies, two individuals will not have eight or more common minutiae. [1,2]. A typical live-scan fingerprint will contain 30-40 minutiae. Other systems analyze tiny sweat pores on the finger that, in the same way as minutiae, are uniquely positioned. Furthermore, such methods may be subject to attacks by hackers when biometric features are transferred via Internet [3].

In our case we developed a method that addresses the problem of the rotation and alignment of the finger position. The proposed method is based on computational geometry algorithms (CGA). The advantages of this method are based on a novel processing method using specific extracted features, which may be characterized as unique to each person. These features depend exclusively on the pixels brightness degree for the fingerprint image, in contrast to traditional methods where features are extracted using techniques such as edge and ridge - minutiae points detection. Specifically, these feature express a specific geometric area (convex layer) in which the dominant brightness value of the fingerprint ranges.

For the testing of the accuracy of the proposed method we selected a well-known commercial verification algorithm that is based on Ratha's algorithm. The fingerprint data used in the testing procedure was received from the database of a commercial company. [a]

Thereinafter, we tested the CGA method against Ratha's algorithm with regard to correct positive and negative verification procedures. Finally, the statistical results of both methods were evaluated.

2. METHOD

In brief, the proposed method is described in the following steps:

(1) *Pre-processing stage* — The input image is made suitable for further processing by image enhancement techniques using Matlab [4].

[a] is available free on the Internet :
http://www.neurotechnologija.com/download.html

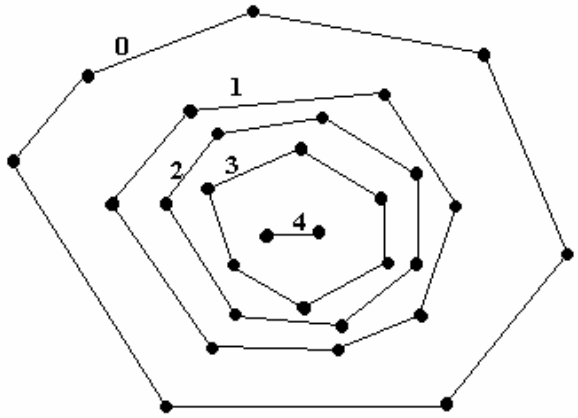

Figure 1. Onion Layers of a set of points (coordinate vector).

(2) *Processing stage* — The data, which comes from step 1, is submitted to specific segmentation (data sets) using computational geometry algorithms implemented via Matlab (see Figure 1).
(3) *Meta-processing stage (during registration only)* — The smallest layer (convex polygon) of the constructed onion layers is isolated from the fingerprint in vector form (see Figure 2).
(4) *Verification stage* — This stage consists of the following steps:

    (a) An unknown fingerprint is submitted to the proposed processing method (Steps 1 and 2), and a new set of onion layers is constructed.
    (b) The referenced polygon that has been extracted during the registration stage is intersected with the onion layers and the system decides whether the tested vector identifies the onion layers correctly or not.

(5) *Evaluation of the algorithm in comparison to Ratha's algorithm* — The above procedure is repeated using a well-known commercial verification algorithm that is based on Ratha's algorithm.

2.1. *Pre-processing stage of CGA method*

In this stage a fingerprint image, which is available from any of the known image formats (tif, bmp, jpg, etc), is transformed into a matrix (a two-dimensional array) of pixels [5]. Consider, for example, the matrix of pixel values of the aforementioned array. Then the brightness of each point is proportional to the value of its pixel. This gives the synthesized image of a bright square on a dark background. This value is often derived from the output of an A/D converter. The matrix of pixels, i.e. the fingerprint image, is usually square and an image will be described as N x N m-bit pixels [6,7], where N is the number of points along the axes and m controls the number of brightness values. Using m bits gives a range of 2 m values, ranging from 0 to 2 m -1. Thus, the digital image may be denoted as the following compact matrix form:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,N-1) \end{bmatrix} \tag{1}$$

The coordinate vector of the above matrix is:

$$\mathbf{S} = [f(x,y)] \tag{2}$$

Thus, a vector $1 \times N^2$ of dimension is constructed, which is then used in the next stage [8].

2.2. *Processing stage of CGA method*

Proposition: We considered that the set of brightness values for each fingerprint image contains a convex subset, which has a specific position in relation to the original set. This position may be determined by using a combination of computational geometry algorithms, which is known as Onion Peeling Algorithms [9] with overall complexity *O(d*n log n)* times.

Implementation: We consider the set of brightness values of a fingerprint image to be the vector $\mathbf{S}$ (eq.2). The algorithm starts with a finite set of points $\mathbf{S} = \mathbf{S_0}$ in the plane, and the following iterative process is considered. Let $\mathbf{S}_1$ be the set

$$S_0 - \partial \mathrm{H}(S_0) : S,$$

minus all the points on the boundary of the hull of **S** . Similarly, define

$$S_{i+1} = S_i - \partial \mathrm{H}(S_i).$$

The process continues until the set is (see Figure 1). The hulls are called the layers of the set, and the process of peeling away the layers is called onion peeling for obvious reasons (see Figure 1). Any point on is said to have onion depth, or just depth . Thus, the points on the hull of the original set have depth 0 (see Figure 1).

2.3. *Meta-processing stage of CGA method*

In our case we consider that the smallest convex layer that has depth 3 (see Figure 1) carries specific information, because this position gives a geometrical interpretation of the average of the fingerprint brightness [5]. This feature may be characterized as unique to each fingerprint because the two (2) following conditions are ensured:

(i) The selected area layer is non-intersected with another layer.
(ii) The particular depth of the smallest layer is variable in each case.

Thus, from the proposed fingerprint processing method two (2) variables are extracted: the area of the smallest onion layer and the depth of this layer, which is a subset of the original fingerprint set S values.

2.4. *Verification stage of CGA method*

In this stage we tested the subset $\mathbf{S_{xy}}$ against a new subset set $\mathbf{N_{xy}}$, which came from the processing of another set **N**. This testing takes place at the following 3 levels (see Figure 2). Subset $\mathbf{S_{xy}}$ is cross-correlated with subset $\mathbf{N_{xy}}$.

(i) The depths of the iterative procedure, from which the subsets were extracted, are compared.
(ii) The intersection between subset $\mathbf{N_{xy}}$ convex layer and one of set **S** onion layers is controlled.

Furthermore, it is considered that subset $\mathbf{N_{xy}}$ identifies set **S** as the parent onion layers when:

(i) The cross-correlation number of subset $\mathbf{S_{xy}}$ $\mathbf{N_{xy}}$ is approximately 1
(ii) The intersection [11] between the convex layer of subset $\mathbf{N_{xy}}$ and one of the onion layers of set **S** is 0.

Otherwise, subset $\mathbf{N_{xy}}$ does not identify set $\mathbf{S}$ as the parent onion layers.

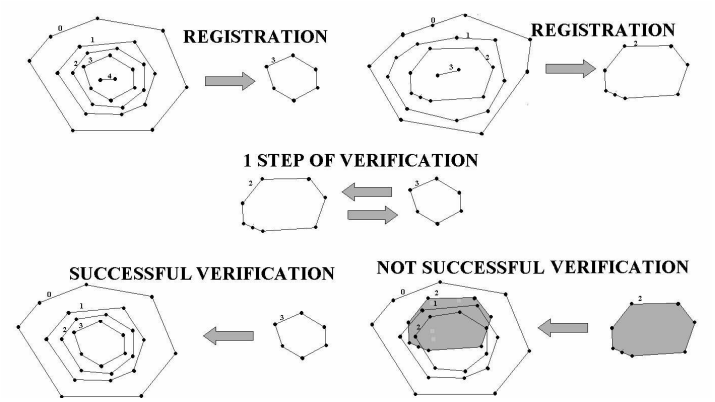

Figure 2. Theoretical presentation of the registration and verification stages of two (2) onions' layers.

2.5. *Verification stage based on Ratha's algorithm*

Fingerprint verification based on Ratha's algorithm is a technique [11,12] to assign a fingerprint into one of the several pre-specified types previously described. Fingerprint verification can be viewed as a coarse level matching of the fingerprints. An input fingerprint is first matched at a coarse level to one of the pre-specified types and then, at a finer level, it is compared to the subset of the database containing that type of fingerprint only. We have developed an algorithm to classify fingerprints into five classes, namely, whorl, right loop, left loop, arch, and tented arch. The algorithm separates the number of ridges present in four directions (0 degree, 45 degree, 90 degree, and 135 degree) by filtering the central part of a fingerprint with a bank of Gabor filters. This information is quantized to generate a FingerCode, which is used for classification [13,14].

3. EXPERIMENTAL PART

In this experiment forty-eight (48) index-finger prints belonging to six (6) individuals (6x8=48) were tested. More specifically, each index-finger print of an individual was tested against the other seven (7) in its group and the forty (40) prints of the other five individuals. In total 2256 or $2 * C_2^8 = \frac{8!}{2!*(8-2)!} = 2256$ verification tests for each of the two methods took place.

3.1. *Pre-processing stage*

In our experiment, each of the recorded fingerprints in TIFF format is represented by a complete 255×255 image matrix (equation 1), which came from a converting quantization sampling process implemented via the *imread.m* Matlab function.

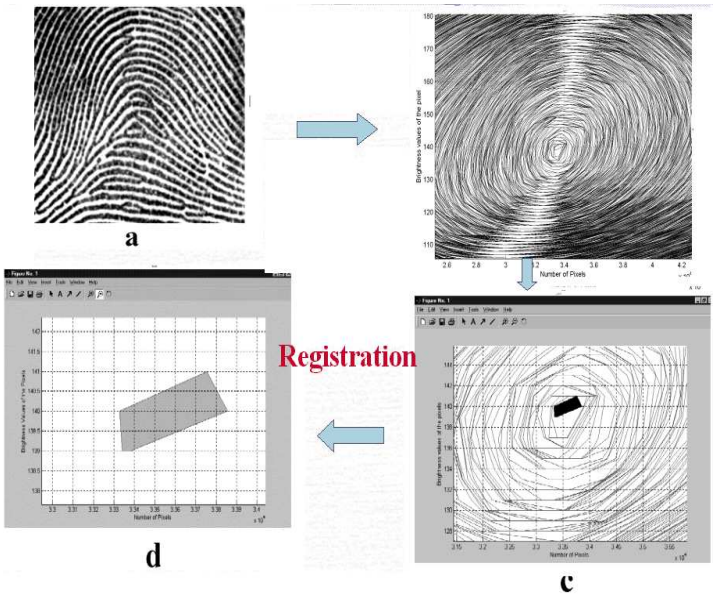

Figure 3. The analytical procedure of the feature extraction of a fingerprint in 4 frames.

(i) Each pixel of the used fingerprint consists of 8 bits, therefore m=8 and the gray levels of brightness range between 0 and 255.
(ii) The dimension of the created compact matrix $f(x, y)$ of equation 1 is **S** and the coordinate vector is respectively.

3.2. *Processing stage*

The coordinate vector, which was extracted in the pre-processing stage, is submitted to further processing. In particular, the onion layers of vector S are created according to the computational geometry algorithm (figure 3a), which was described in Section 2.2. Thus, a variable number of layers (convex polygons) were extracted for each fingerprint case. In this case, the

created onion consisted of 944 layers (convex polygons), and the number of vertices of the smallest internal layer was five (5). Furthermore, the average of vector value **S** in this example was 140,67.

3.3. *Meta-processing stage*

As can be seen in figure 3d the area that encloses the smallest internal layer contains the aforementioned average value. In other words, the area of this layer may be characterized as a specific area in which the dominant brightness value of the fingerprint ranges.

3.4. *Verification stage*

In this stage, it is assumed that the referenced polygon A, must lead to a rejection decision. Then we applied the aforementioned VERIFICATION conditions in order for the system to decide whether polygon B is correctly identified or not. The final decision of this system is that the tested fingerprint is not identified correctly for the following reasons:

Table 1. Positive and Negative Fingerprint Verification Scores using Ratha's and Onion Algorithms.

Individuals	Ratha's						Onion					
A	45	1	0	1	0	0	48	0	0	0	0	0
B	1	44	1	0	0	0	0	47	1	0	0	0
C	1	1	45	1	1	1	1	0	48	0	0	0
D	0	1	1	44	1	0	0	1	0	46	0	0
E	1	0	0	0	44	0	0	0	1	0	47	0
F	0	0	0	1	0	45	1	0	0	0	0	46

(i) The depth of the smallest referenced layer (polygon) was 944 in contrast to that of the tested vector that was 677 respectively.
(ii) The layer of the tested polygon intersected the other layers.

Especially in the negative correct verification case the final decision of the system depended on the position and the sizes of the final characteristic polygons.

3.5. *Ratha's algorithm*

In this case we used a well-known commercial verification algorithm that is based on Ratha's algorithm.

4. RESULTS

In this experiment forty-eight (48) index-finger prints belonging to six (6) individuals (6x8=48) and called A, B, C, D, E and F, were tested. More specifically, each index-finger print of an individual was tested against the other seven (7) in its group and the forty (40) prints of the other five individuals. In total 2256 verification tests for each of the two methods took place.

4.1. *Statistical evaluation*

As can be seen from the diagonal scores on in the above table (1) the correct positive verification test score for the Ratha method, 156/168=0.93 or 93% and for the CGA method is 165/168=0.98 or 98%. Furthermore, the correct negative verification score for the Ratha algorithm is 933/940=0,99 or 99% and the correct negative verification for the CGA method is 938/940 or approximately 100%. In contrast, the false positive verification scores for the Ratha method is 7% and for the CGA method 2%. At this point it is to be noted that the false results of the Ratha method were yielded when the tested image had variations for rotation reasons. On the other hand, the CGA false results were yielded for those tested fingerprint specimens that were not complete.

5. CONCLUSION

From the results of the experiment it is ascertained that the proposed method, bearing in mind security considerations, can be used for accurate and secure fingerprint verification purposes because the proposed feature extraction is based on a specific area in which the dominant brightness value of the fingerprint ranges. Moreover, the proposed method promisingly allows very small false acceptance and false rejection rates, as it is based on specific segmentation. It has to be noted that biometric applications will gain universal acceptance in digital technologies only when the number of false rejections / acceptances approach zero. The results of this comparison showed that the proposed method yields correct positive and correct negative verification scores greater than 99%. In particular, the proposed CGA

method produced extremely reliable results even in cases where the tested fingerprints were complete specimens yet the position or pressure applied was not consistent [15]. The computational complexity of the proposed algorithm may also be characterized as extremely competitive.

References

1. A. K. Jain, A. Ross and S. Pankanti, Fingerprint matching using minutiae and texture features, *Proc. International Conference on Image Processing ICIP,Thessalonica, GR*, 281-285 (2001).
2. D. Maio, D. Maltoni, Direct gray-scale minutiae detection in fingerprints, *IEEE Transactions on PAMI* **19(1)**, 27-40 (1997).
3. L. O'Gorman, Fingerprint verification, in Biometrics, *S.: Kluwer Acadenic Publishers* (1999).
4. T. Poon , P. Banerjee, Contemporary Optical Image Processing With Matlab, *Hardcover: Elsevier Science Ltd* (2001).
5. R. Bracewell, Two-Dimensional Imaging, *NJ: Prentice - Hall, Upper Sandle River* (1995).
6. M. Nixon, A. Aguado, Feature Extraction and Image Processing, *GB: Newnes-Oxford* (2002).
7. R. Gonzales, R. Woods, Digital Image Processing, *NJ: Prentice - Hall, Upper Sandle River* (2002).
8. M. Spiegel, Theory and Problems of Vector Analysis, *London: McGraw-Hill* (1974).
9. J. O'Rourke, Computational Geometry in C, *NY: Cambridge University Press* (1993).
10. J. O'Rourke, J. Chien, C. Olson, and T. Naddor,A new linear algorithm for intersecting convex polygons, *Comput. Graph. Image Proces.* **19 (4)**, 384-391 (1982).
11. C. Calabrese, The Trouble with Biometrics, *Login* **24(4)**, 56-61 (1999).
12. G. Hachez, F. Koeune and J.J. Quisquater, Biometrics, Access Control, Smart Cards: a Not So Simple Combination, *Proc. of the 4th Working Conference on Smart Card Research and Advanced Applications (CARDIS 2000), Bristol, GB* 273-278 (2000).
13. B. Schneier, Applied Cryptography, Protocols, Algorithms and Source Code in C, *GB: Elsevier, 2nd Edition* (1996).
14. A.K. Jain, L. Hong, R. Bolle and S. Pankanti, System and method for deriving a string-based representation of a fingerprint image *US Patent* **12(6)**, 487-496 (2002).
15. A.K. Jain, L.Hong, R.Bolle and S. Pankanti, Determining An Alignment Estimation Between Two (Fingerprint) Images *US Patent* **11(6)**, 314-319 (2001).