Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 4, pp. 525-531

Fingerprints Identification using a Fuzzy Logic System

I. Iancu, N. Constantinescu, M. Colhon

Ion Iancu, Nicolae Constantinescu, Mihaela Colhon
Department of Informatics
University of Craiova,
Al.I. Cuza Street, No. 13, Craiova RO-200585, Romania
E-mail: i_iancu@yahoo.com, nikyc@central.ucv.ro, mghindeanu@yahoo.com.

Abstract: This paper presents an optimized method to reduce the points number to be used in order to identify a person using fuzzy fingerprints. Two fingerprints are similar if n out of N points from the skin are identical. We discuss the criteria used for choosing these points. We also describe the properties of fuzzy logic and the classical methods applied on fingerprints. Our method compares two matching sets and selects the optimal set from these, using a fuzzy reasoning system. The advantage of our method with respect to the classical existing methods consists in a smaller number of calculations.
Keywords: fuzzy models, fingerprint authentication, cryptographic signature model.

1 Introduction

Fingerprint identification is the most mature biometric method being implemented at an early level since 1960. The recognition of a fingerprint can be done with two methods: "one-to-one" (verification) and "one-to-many" ($1:N$ identification). The first method is applied when we have two fingerprints and we want to verify if they belong to the same person. The second one is used when we have one fingerprint and we search it in a data base. The verification is much easier and faster because we have the two fingerprints and we just need to compare them. On the other hand, the identification implies more time for extracting the fingerprint because there are needed much more details.

The fingerprints are not compared with images, they use a method based on characteristic points named "minutiae". These points are characterized by *ridge ending* (the abrupt end of a ridge), *ridge bifurcation* (a single ridge that divides in two ridges), *delta* (a Y-shaped ridge meeting), *core* (a U-turn in ridge pattern), etc. All these features are grouped in three types of lines: *line ending*, *line bifurcation* and *short line*. After the minutiae points are localized, a map with all their locations on the finger is created. Every minutiae point has associated two coordinates (x,y), an angle for orientation and a measure for the fingerprint quality. The matching of two fingerprints depends on the position and on the rotation. For this reason, every fingerprint is represented, not only, as a group of points with two coordinates, but also, as a group of points with coordinates relative to other points. This allows obtaining an unique positioning of a point regarding to other three points. The three selected points must not be collinear. When two fingerprints are compared, first are compared the relative coordinates. If this stage ends successfully, these coordinates are transformed in 2D coordinates and verified.

After verifying the fingerprints, the result will tell us if they are from the same person with a high probability. Still, the cases when the belonging probability of a fingerprint is 0 (false) or 1(true) are rarely. In most of the cases, the probability will be a number $p \in [0,1]$. This fact leads to a fuzzy logic. The values in fuzzy logic can range between 0 and 1 (1 is for absolute truth, 0 for absolute falsity). A fuzzy value for an element x will express the degree of membership of x in a set X. It is essential to realize that fuzzy logic uses truth degrees as a mathematical model of the vagueness phenomenon while probability is a mathematical model of randomness.

2 State of the Art

Two fingerprints are similar if n out of N points match. To verify this, Freedman et al. introduced the fuzzy matching protocols [3]. Using these protocols, the information about the fingerprint we want to identify (or verify) will not be revealed if no match is found. To describe the fuzzy private matching problem we will take a set of words $X = x^1 \dots x^N$ where $\{x^i\}$ are the letters. Two words $X = x^1 \dots x^N$ and $Y = y^1 \dots y^N$ match only if: $n \leq |\{k : x^k = y^k \mid 1 \leq k \leq N\}|$ and this relation is denoted with $X \approx_n Y$. In the subsequent we will name the set X as the *total set for selection*. The input of the protocol will be two sets of words ($X = X_1 \dots X_m$ for the client and $Y = Y_1 \dots Y_s$ for the server) and the parameters m, s, N and n. While the output of the server is empty, the output of the client will be a set $\{Y_i \in Y | \exists X_i \in X : X_i \approx_n Y_i\}$, where $A \approx_n B$ means that the points A and B are very close. This set is, in fact, the intersection of the two input sets [1]. It was demonstrated that this protocol leads information about the input even if no match is found [1]. Another protocol, based on Freedman's protocol, was presented in [1]. It uses σ as a combination of n different indices $\gamma_1, \gamma_2 \dots \gamma_n$ and $\sigma(X) = x^{\gamma_1} || \dots || x^{\gamma_t}$ for a word X. After the parameters and the public key are sent, the client constructs a polynomial representation of the points set:

$$P_\sigma = (x - \sigma(X_1)) * (x - \sigma(X_2)) * \dots * (x - \sigma(X_m))$$

This is a feedback polinomial value for a set of fingerprints. Then he sends $\{P_\sigma\}_{k_2}$ to the server. The server analyzes every received polynomial $\{P_\sigma\}$ at the point $\sigma(Y_i)$ and computes $\{w_i^\sigma\}_{k_2} = \{r * P_\sigma(\sigma(Y_i)) + Y_i\}_{k_2}$, where r is a random value. After all the calculations, the server sends $\{w_i^\sigma\}_{k_2}$ to the client. The client will decrypt all the messages and if w_i^σ matches with any word from X then it is added to the output set. $\{w_i^\sigma\}_{k_2}$ is a combination between fingerprint points value and the parameter which characterizes the common information between a base set and the current set of collected values.

A particular scheme of fingerprint authentication describes a method which is not based on the minutiae points [12], but by the texture of the finger, called FingerCode. Such a FingerCode is a vector composed from 640 values between 0 and 7. The vector is ordered and stable in size. The method uses Euclidean distance to find the matching. After estimating the block orientation, a curvature estimator is designed for each pixel. Its maximal value is, in fact, the morphological searched center. Using a properly tuned Gabor filter ([11, 13]) we can catch ridges and valleys from the fingerprint. The FingerCode is computed as the average absolute deviation from the mean of every sector of each image. Error-correction for the FingerCode would never be efficient enough to recognize a user. In [12], the method proposed uses a secret $d + 1 - letter\ word$, which correspond to the $d + 1$ coefficients of a polynomial p of degree d. The public key will be extract from (F, p), where F is the FingerCode. Then, we choose n random point of p. These points will be hidden like in a fuzzy commitment scheme. To find the polynomial p, each point is decoded. If at least $d + 1$ points are decoded then p can, also, be retrieved.

A method based on the minutiae points and, also, on the pattern of the finger was presented in [4]. All the ridges that cross a line (x, y) where x and y are minutiae points are counted. Then, are presented all the possible combinations of three minutiae points and the ridges crossing that line. Such a combinations' list needs C_n^3 entries, where n is the number of minutiae points. This method is more complex because before all the calculations are done we need to identify the minutiae points and then combine them.

3 Our Method

3.1 System Description

A commercial fingerprint-based authentication system requires a very low False Reject Rate (FRR) for a given False Accept Rate (FAR) where FAR is *the probability that the system will incorrectly identify* and FRR is *the probability of failure in identification*.

Our method is, also, based on the minutiae points of the fingerprints. We can identify at least 40 minutiae points on a fingerprint, depending on its quality. In general, the number of the minutiae points varies from 0 to 100. All the methods mentioned above can be applied to a fingerprint verification. But, for an identification we need an algorithm with a low level of complexity because the data bases used in practice have millions of fingerprints. To reduce the search time and complexity, we first propose to classify the fingerprints, and then, to identify the input fingerprints only in one subset of the data base. To choose the right subset the fingerprint is matched at a coarse level to one of the existing types. After that, it is matched at a finer level to all the fingerprints of the subset. The FBI in the United States recognize eight different types of patterns [5]. For example, we have an input fingerprint and we want to identify it in a data base with 15000 entries. We will take the minim number of minutiae points, 40. If no classification is made we have to do at least $40 \times 15000 = 600000$ operations. But, if we use a classification with eight types (each subset has the same number of fingerprints $15000/8 = 1875$) we will have at least $(8+1875) \times 40 = 75320$ calculations. This is because we will first compare the input fingerprint with each group and after that it will be compared with each element of the chosen group. As we can see, the calculations are reduced to only $12,5\%$. The classification of the fingerprints is preferred to have more than three types of subsets. This is because a higher accuracy is achieved. Such a classification, also, helps to reduce the number of calculations with a higher percentage.

3.2 Fuzzy Mathematical Background

A fuzzy set A in X is characterized by its membership function:

$$\mu_A : X \to [0,1]$$

where $\mu_A(x) \in [0,1]$ represents the membership degree of the element x in the fuzzy set A. We will work with membership functions represented by trapezoidal fuzzy numbers. Such a number $N = (\underline{m}, \overline{m}, \alpha, \beta)$ is defined as

$$\mu_N(x) = \begin{cases} 0 \ \ for \ \ x < \underline{m} - \alpha \\ \frac{x - \underline{m} + \alpha}{\alpha} \ \ for \ \ x \in [\underline{m} - \alpha, \underline{m}] \\ 1 \ \ for \ \ x \in [\underline{m}, \overline{m}] \\ \frac{\overline{m} + \beta - x}{\beta} \ \ for \ \ x \in [\overline{m}, \overline{m} + \beta] \\ 0 \ \ for \ \ x > \overline{m} + \beta \end{cases}$$

The rules are represented by fuzzy implications. Let X and Y be two variables whose domains are U and V, respectively. The rule

$$if \ X \ is \ A \ then \ Y \ is \ B$$

is represented by its conditional possibility distribution ([14], [15]) $\pi_{Y/X}$:

$$\pi_{Y/X}(v,u) = \mu_A(u) \to \mu_B(v), \ \forall u \in U, \ \forall v \in V$$

where $\to$ is an implication operator ([2]) and μ_A and μ_B are the membership functions of the fuzzy sets A and B, respectively. One of the most important implication is Lukasiewicz implication [2], $I_L(x,y) = \min(1 - x + y, 1)$.

3.3 Proposed Fuzzy Logic System

Fuzzy control provides a formal methodology for representing, manipulating and implementing human's heuristic knowledge about how to control a system. In a fuzzy logic controller, the expert knowledge is of the form

IF (*a set of conditions are satisfied*) *THEN* (*a set of consequences are inferred*)
where the antecedents and the consequences of the rules are associated with fuzzy concepts (linguistic terms). The most known systems are: Mamdani, Tsukamoto, Sugeno and Larsen which work with crisp data as inputs. A Mamdani type model which works with interval inputs is presented in [10].

In this paper we use a version of Fuzzy Logic Control (FLC) system from [9] in fingerprints identification. This version is characterized by:

- the *linguistic terms* (or values), that are represented by trapezoidal fuzzy numbers
- *Lukasiewicz implication*, which is used to represent the rules
- the *crisp control action of a rule*, computed by Middle-of-Maxima method
- the *overall crisp control actions*, computed by discrete Center-of-Gravity.

We assume that the facts can be given by crisp data, intervals and/or linguistic terms and a rule is characterized by:

- a set of linguistic variable A, having as domain an interval $I_A = [a_A, b_A]$
- n_A linguistic values $A_1, A_2, ..., A_{n_A}$ for each linguistic variable A
- membership function $\mu^0_{A_i}(x)$ for each value A_i, where $i \in \{1, 2, ..., n_A\}$ and $x \in I_A$.

According to the structure of a FLC, the following steps are necessary in order to work with our system.

Firing levels

We consider an interval input $[a,b]$ with $a_A \leq a < b \leq b_A$. The membership function of A_i is modified ([10]) by membership function of $[a,b]$ as follows

$$\forall x \in I_A, \mu_{A_i}(x) = min(\mu^0_{A_i}(x), \mu_{[a,b]}(x))$$

where

$$\mu_{[a,b]}(x) = \begin{cases} 1 & if \quad x \in [a,b] \\ 0 & otherwise \end{cases}$$

It is obvious that, any t-norm T can be used instead of *min* (see, for instance, [6–8]).

The firing level, generated by the input interval $[a,b]$, corresponding to the linguistic value A_i is given by:

$$\mu_{A_i} = \max\{\mu_{A_i}(x) | x \in [a,b]\}.$$

The firing level μ_{A_i}, generated by a linguistic input value A'_i is

$$\mu_{A_i} = \max\{\min\{\mu^0_{A_i}(x), \mu_{A'_i}(x)\} | x \in I_A\}.$$

The firing level μ_{A_i}, generated by a crisp value x_0 is $\mu^0_{A_i}(x_0)$.

Fuzzy inference

We consider a set of fuzzy control rules

$$R_i : if\ X_1\ is\ A^1_i\ and\ ...\ and\ X_r\ is\ A^r_i\ then\ Y\ is\ C_i$$

where the variables $X_j, j \in \{1, 2, ..., r\}$, and Y have the domains U_j and V, respectively. The firing levels of the rules, denoted by $\{\alpha_i\}$, are computed by

$$\alpha_i = T(\alpha^1_i, ..., \alpha^r_i)$$

where T is a t-norm and α^j_i is the firing level for $A^j_i, j \in \{1, 2, ..., r\}$. The conclusion inferred from the rule R_i, using the Lukasiewicz implication is

$$C'_i(v) = I(\alpha_i, C_i(v)), \forall v \in V.$$

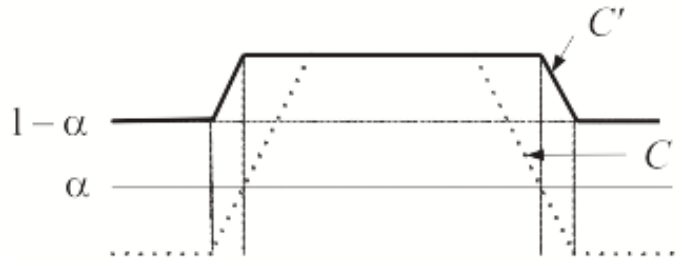

Figure 1: Conclusion obtained with Lukasiewicz implication

Defuzzification

The fuzzy output C_i' of the rule R_i is transformed in a crisp output z_i using the Middle-of-Maxima operator. The crisp value z_0 associated to a conclusion C' inferred from a rule having the firing level α and the conclusion C represented by the fuzzy number $(\underline{m}_C, \overline{m}_C, \alpha_C, \beta_C)$ is:

$$z_0 = \frac{\underline{m}_C + \overline{m}_C + (1-\alpha)(\beta_C - \alpha_C)}{2}$$

The overall crisp control action is computed by the discrete Center-of-Gravity method: if the number of fired rules is N then the final control action is

$$z_0 = \left(\sum_{i=1}^{N} \alpha_i z_i \right) / \sum_{i=1}^{N} \alpha_i$$

where α_i is the firing level and z_i is the crisp output of the i-th rule.

4 An application in fingerprint identification

For the proposed FLC we consider rules with two inputs and one output. The input variables are $\sigma_1 = \{x_i | x_i \in X\}$ and $\sigma_2 = \{x_j | x_j \in X, x_i \neq x_j, \forall i, j\}$ where X is the set defined in Section 2. By σ_1 we represent the values set of basic input data and σ_2 is a user data to be evaluated and authenticated. These values sets will be denoted by S_1 and S_2 respectively. The σ set values denote the optimal points set to be used for the fuzzy matching authentication, according with S output variable. The fuzzy rule-base consists of

R1: If S_1 is *Low* and S_2 is *Very Low* then S is *Low*
R2: If S_1 is *Very Low* and S_2 is *Low* then S is *Low*
R3: If S_1 is *Very Low* and S_2 is *Very Low* then S is *Very Low*
R4: If S_1 is *Very Low* and S_2 is *Very Low* then S is *Low*
R5: If S_1 is *Very Low* and S_2 is *Middle* then S is *Middle*
R6: If S_1 is *Very Low* and S_2 is *Middle* then S is *Low*
R7: If S_1 is *High* and S_2 is *Very High* then S is *High*
R8: If S_1 is *Very High* and S_2 is *High* then S is *High*
R9: If S_1 is *Very High* and S_2 is *Very High* then S is *High*
R10: If S_1 is *Low* and S_2 is *Very High* then S is *Middle*

The value *Very Low* for a variable S_1, S_2 or S represents a minimum degree of trust while the value *Very High* represents the maximum degree. There are five linguistic values for every variable

$$\{Very\ Low,\ Low,\ Middle,\ High,\ Very\ High\}.$$

We consider the universes of discourse $[0, 10]$. The membership functions corresponding to the linguistic values are represented by the following trapezoidal fuzzy numbers:

- for the variable $S1$: $\{(0,1,0,1.5),(3,4,1,0.5),(5,6,1,0.5),(7.5,8.5,3,1),(9.5,10,0.5,0)\}$
- for the variable $S2$: $\{(0,1.5,0,1),(2.5,4,0.5,0.5),(5,6,2,0),(6.5,8,1.5,0),(8.5,10,0.5,0)\}$,
- for the variable S: $\{(0,1,0,1.5),(2.5,4,0.5,0.5),(5,7,2.5,0.5),(8,8.5,2,0.5),(9.5,10,1,0)\}$

We consider the following interval input values: $[1.5,2.2]$ for $S1$ and $[3.2,4.2]$ for $S2$. The positive firing levels corresponding to the linguistic values of the input variable $S1$ are

$$\mu_{VeryLow} = 0.666, \mu_{Low} = 0.2$$

and the positive firing levels corresponding to the linguistic values of the input variable $S2$ are:

$$\mu_{Low} = 1, \mu_{Middle} = 0.6$$

The fired rules and their firing levels, computed with t-norm Product $T(x,y) = xy$, are:

$$R_2 \text{ with firing level } \alpha_2 = 0.666,$$
$$R_5 \text{ and } R_6 \text{ with } \alpha_5 = \alpha_6 = 0.3996.$$

The fired rules give the following crisp values as output:

$$z_2 = 3.25,\ z_5 = 5.3996,\ z_6 = 3.25;$$

then the overall crisp control action is

$$z_0 = 3.836.$$

These values represent the matching approach for every subset points which are candidate to be in the final set, and are computed using a fuzzy merging comparison between selection sets σ_1 and σ_2. The optimal selection set (which has less points) is represented by the output variable σ.

5 Conclusions

Among all the biometric techniques, the identification based on fingerprints is used in the most applications. The uniqueness of the fingerprint can be determinate by the pattern of ridges and the minutiae points. For identifying an input fingerprint, the proposed method uses a fuzzy classification of the data. The proposed system is much more efficient than the FLC presented in [8]. This is because, in order to reduce the necessary points number, we find the minutiae points by using a fuzzy logic reasoning system which compare two points sets matching values. In a practical application, it is recommended to use the proposed system with a set of implications and aggregate the results given by every implication, in order to obtain the overall output; in this way can be obtained a stronger base for more accurate results of our system. We intend to use these results in a future work, by maping their relative placement on the finger, and comparing all its points with the ones of the fingerprints for the right subset.

Bibliography

[1] L. Chmielewski and J. H. Hoepman, Fuzzy Private Matching (Extended Abstract), *ARES '08: Proceedings of the 2008 Third International Conference on Availability, Reliability and Security*, IEEE Computer Society, pp. 327–334, 2008.

[2] E. Czogola and J. Leski, On equivalence of approximate reasoning results using different interpretations of fuzzy if-then rules, *Fuzzy Sets and Systems*, vol. 117, no. 2, pp. 279–296, 2001.

[3] M. Freedman, K. Nissim and B. Pinkas, Efficient private matching and set intersection, *Advances in Cryptology, EUROCRYPT 2004*, Springer-Verlag, pp. 1–19, 2004

[4] R. S. Germain, A. Califano and S. Colville, Fingerprint Matching Using Transformation Parameter Clustering, *IEEE Comput. Sci. Eng.*, vol. 4, no. 4, pp. 42–49, 1997.

[5] M. R. Hawthorne, *Fingerprints. Analysis and Understanding*, CRC Press, 2009

[6] I. Iancu, T-norms with threshold, *Fuzzy Sets and Systems. International Journal of Soft Computing and Intelligence*, vol. 85, no. 1, pp. 83–92, 1997.

[7] I. Iancu, Operators with n-thresholds for uncertainty management, *Journal of Applied Mathematics & Computing*, Springer Berlin, vol. 19, no. 1-2, pp. 1–17, 2005.

[8] I. Iancu, Generalized Modus Ponens Using Fodor's Implication and T-norm Product with Threshold, *International Journal of Computers, Communications & Control (IJCCC)*, vol. 4, no. 4, pp. 330–343, 2009.

[9] I. Iancu, Extended Mamdani Fuzzy Logic Controller, *The Fourth IASTED International Conference on Computational Intelligence CI 2009*, ACTA Press, vol. 5, pp. 143–149, 2009.

[10] F. Liu, H. Geng and Y. Q. Zhang, Interactive Fuzzy Interval Reasoning for Smart Web Shopping, *Applied Soft Computing, Elsevier*, vol. 5, no. 4, pp. 433–439, 2005.

[11] D. G. Radojevic, Fuzzy Set Theory in Boolean Frame, *International Journal of Computers, Communications & Control (IJCCC)*, vol. 3, no. 5, pp. 121–131, 2008.

[12] V. V. T. Tong, H. Sibert, J. Lecoeur and M. Girault, Biometric fuzzy extractors made practical: a proposal based on Finger Codes, *Advances in Biometrics*, Springer Berlin / Heidelberg, pp. 604–613, 2009.

[13] T. Vesselenyi, S. Dzitac, I. Dzitac and M.J. Manolescu, Fuzzy and Neural Controllers for a Pneumatic Actuator, *International Journal of Computers, Communications & Control (IJCCC)*, vol. 4, no. 2, pp. 375–387, 2007.

[14] L. A. Zadeh, A theory of approximate reasoning, *Machine Intelligence 9, Elsevier*, pp. 149–194, 1979.

[15] L. A. Zadeh, Fuzzy sets as a basis for a theory of a possibility, *Fuzzy Sets and Systems*, vol. 100, pp. 9–34, 1999.