On-line signature verification based on *optimal feature representation* and neural-network-driven fuzzy reasoning

Julio Martínez-R., *Member, IEEE*. Rogelio Alcántara-S., *Member, IEEE.*

Abstract— **In this paper we present an innovative approach to on-line handwritten signature verification based on *optimal feature representation* (OFR) and neural-network-driven fuzzy reasoning (NND-FR); OFR is introduced here and is different from feature selection and generation. To create a reference signing model of a person, a set of shape features and dynamic features are extracted from a set of original signatures. Subsequently, for each distinctive feature, an averaged prototype and a consistency function are calculated using genetic optimization, this procedure derived from our concept of *optimal feature representation*. Two NND-FR systems are trained using the prototype and consistency function of each feature. For verification, every feature is compared against its optimal reference using the NND-FR scheme, evaluating separately the form and dynamics of a questioned signature. Our verifier was tested using 1923 signatures from 36 signers in the context of random and skilled forgeries, resulting in a misclassification rate as low as 0.14%.**

Index terms— **feature representation, feature selection, genetic algorithms, neural networks, neural-network-driven fuzzy systems, on-line signature verification**

I. Introduction

Automatic signature verification is an active field of research with many practical applications, ranging from access control to sensitive resources and law enforcement to security and fraud prevention in checks and credit cards [1]. A comprehensive survey of distinctive features in handwritten signatures, classifiers and other techniques related to on-line signature verification can be found in [2], [3], [4], and [5].

Signature verification can be considered a special case of pattern recognition. Like in any pattern recognition problem, in signature verification distinctive features can be extracted from a set of original signatures. These features can be in the form of functions of time or global parameters [5].

The problem of *feature selection*, is related to the fact that the number of features at the disposal of the designer of a classification system is usually very large. This number can easily become of the order of dozens or hundreds.

A subset of features out of the universe of all the avaiable features can be selected according to some criterion, such as statistical hypothesis testing or class separability, thus reducing the quantity of features to be used to a few. *Feature generation,* on the other hand, is the transformation of the selected features into a format that provides, for example, a higher class separation. Feature generation includes linear transforms, such as the Karhunen-Loève transform, discrete Fourier transform, discrete Wavelet transforms, Wavelet packets, and others [6]. An appropiately chosen transform can exploit and remove information redundancies, which usually exist in the set of samples obtained by the measuring devices.

Once a feature has been selected and properly transformed, it must be *represented*, meaning that a reference or prototype must be created from a set of exemplars of the feature. Typically, a feature is represented by its mean or average value.

As we shall show later on, an appropiate representation can be created to ensure the highest between-class separatibity between class ω_0 and class ω_1, which for singature verification represent the genuine class and the forgery class. This is done creating an optimized representation in contrast to a simple average representation. What is remarkable here is a) the optimized representation of features have not been previously exploited, existing only reference to feature selection, feature generation [6], [7] and reestructuration of feature space [8], and b) this representation is quite significant to achieve excelent results in any pattern recognition problem.

In regard to the application of on-line signature verification, shape-related and dynamics-related features are calculated. The results obtained in this stage are used to train two NND-FR systems per signer, in an architecture that resembles the process followed by human forensic examiners [9]. The judgment of genuiness is established by a 2-input/1-output fuzzy classifier, which evaluates the balance between rhythm and form of a questioned signature with respect to the writing habits of a person.

The rest of this paper is organized as follows. Section II describe the concept of optimal feature representation in comparison to variable and feature selection. The

Manuscript received november 20, 2002.

J. Martínez-R. is with the Institute of Technology of Aguascalientes, Departament of Electrical and Electronics Engineering. Aguascalientes, México, and with the Faculty of Engineering, UNAM, Department of Electrical Engineering. (Phone: +52 449 910 50 02 ext. 106, 127; fax: +52 449 970 24 03; e-mail: jcmartin@verona.fi-p.unam.mx).

R. Alcántara-S., is with the Faculty of Engineering, UNAM, Department of Electrical Engineering. (e-mail: rogeras@servidor.unam.mx).

overall approach of our automated signature verificator is stablished in section III. Section IV deals with the NND-FR systems as used by the verification system. The fuzz combiner of shape and dynamics characteristics a questioned signature is presented in section V. Section VI explains the nature of the database collected for signature verification and shows the verification results. Finally, in section VIII some conclusions are drawn.

II. Optimal feature representation

Optimal feature representation (OFR) is a concept that we are inserting to the field of pattern recognition, and is used here for on-line signature verification producing very encouraging results. We describe this new approach in the following paragraphs.

A. Variable and Feature selecction.

Variable selection, refers to the problem of selecting input variables that are most predictive of a given outcome. This problem is found in all machine learning tasks. Feature selection refers to the selection of an optimum subset of features derived from these input variables [10]. Variable selection has been in the interest of many researchers in areas of application for which datasets with tens of hundreds of thousands of variables are avalaible. Given that more than one feature can be extracted from a single variable, the problem of feature selection is still an open an difficult problem.

The definition of the mathematical statement is not wadely agreed. Some interpretations include a) discovering the variables relevant to the concept, and how relevant they are, and b) finding a mininum subset of variables that are useful to the predictor (classifier, regression machine, etc.).

Determining an optimum number of features can be made using methods that asses the quality of feature subsets according to the prediction error of an predictor. These methods are called wrapper methods. Besides, there exist methods that embed feature selection in the learning algorithms.

Some contributions to the art of feature selection are the following. In [11], Vafaie and De Jong, describe an approach to improve the usefulness of machine learning techniques for generating clasiffication of real world data. The heart of their proposal is a feature selection architecture in which a search technique (genetic algorithms) is applied to a feature set, exploring the feature space. Once a feature subset has been selected, a stage based on a given criterion function and a classification process measures the goodness of recognition for the specific feature subset. At the end of this iterative search, a “best feature subset” is selected. In [8] these same authors use genetic algorithms for reestructuring feature space representations. In this case a “best feature set” is obtained applying simple arithmetic operations such as +,-,*,/ to the set of original features. The genes of the genetic algorithm encode certain combinations of arithmetic operations. For example, F1-(F2+F4) can be used to create a new feature, Fx. Fx is evaluated according to an evaluation function, and its discrimination powers are calculated.

In [12], Skalak demonstrates that feature selection can be of great help in reducing computational costs without sacrificing accuracy in pattern recognition problems. In [13], Yu and others use genetic feature selection and fuzzy and crisp nearest neighbor classifiers for hyperspectral satellite imagery. The selected genetic algorithm is a simple one. Once again, feature selection demonstrates to be a powerful tool to improve an automated classification system.

Feature generation, is the transformation of the selected features by linear transforms, such as the Karhunen-Loève transform, discrete Fourier transform, discrete Wavelet transforms, Wavelet packets, and others [6]. An appropiately chosen transform can exploit and remove information redundancies, which usually exist in the set of samples obtained by the measuring devices.

Feature selection and generation are typically the first stages in the design of any pattern recognition system.

B. Feature representation.

So far, the concepts of feature selection and generation have been explained. Now we will introduce the concept of feature representation. In Fig. 1a it can be seen a typical design flow of a pattern recognition system [6]. The design of a pattern recognition system involves, in its early stages, the selection and generation of features. Features can be scalars or vectors.

In the general case *L* scalar features are used and form a feature vector, $\mathbf{x} = [x_1, x_2, ..., x_L]^T$; when the feature is represented by a time series or by a collection of succesive samples, a set of *L* features form a feature matrix, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_L]$; no matter what, each of the feature sets identifies uniquely a pattern. In the sequel, we use interchangeably scalar features and vector features to refer to features.

In Fig. 1b, it is shown an augmented scheme of the design flow. It includes a block named *feature representation*. To understand the functionality of this block lets situate ourselves in the context of a real pattern recognition problem. Without any loss of generality, consider that the stages of feature generation and feature selection have been accomplished. In this way, M (with M<L) features are at the disposal of the designer; suppose now that N samples of the object or phenomenon were used to choose the M features. There exist N rows in a matrix describing N samples or exemplars of the object. Now the question for the

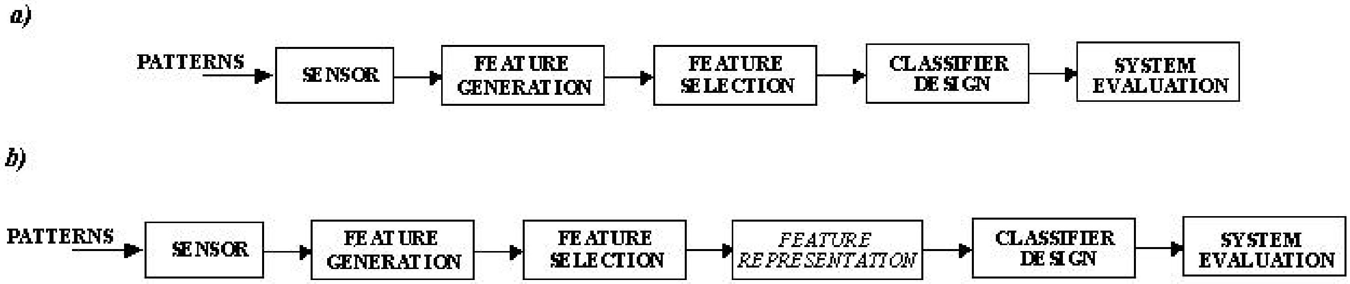

Fig. 1. Insertion of feature representation in the design cycle of a pattern recognition system

designer is: which feature value is the most representative, amongst the N samples, for comparison purposes?. In most cases, the designer of the pattern recognition system decides to use the mean or average value of the feature along the N samples. Assuming N samples and M features, the most representative values of the features can be accomodated in a 1xM vector allocating the mean values of each feature. Figure 2 ilustrates this typical procedure. Although this approach is widely adopted in practice, it is easily demonstrable that such representation does not produce the best between-class and within-class measures in pattern recognition problems, when they are used as reference vectors or class prototypes in two-class or multi-class problems.

What we propose as *optimal feature representation*, is to find a representation capable of producing the largest between-class and an acceptable within-class distance measures. Acceptable here means that the within-class distance is no worse than the within-class distance attainable by a simple mean value representation.

Lets start with some definitions. First, this technique is applicable to both mutli-class (ω_0 ω_1 ω_2 ω_3 ... ω_C) and two-class problems (ω_0 ω_1). A multi-class problem must be converted to a two-class problem for every class, by making $\omega_0 = \omega_c$ and $\omega_1 = \Omega$, where Ω is the set of all the sample feature vectors excluding the samples belonging to class ω_c.

SAMPLE NUMBER	$feature_1$	$feature_2$	$feature_3$	...	$feature_m$
1	x_1	x_2	x_3	...	x_m
2	x_1	x_2	x_3	...	x_m
.	.	.	.		.
.	.	.	.		.
n	x_1	x_2	x_3	...	x_m
avg	$\overline{x_1}$	$\overline{x_2}$	$\overline{x_3}$	...	$\overline{x_m}$

Fig. 2 Typical average representation.

The goal is to find an optimal feature vector representative of class ω_0 whose distance to the closest sample pattern, or feature vector, of class ω_1 is maximum, keeping a reasonable within-class distance. The two characteristics mentioned above are met with the process shown in Fig.3. Note the use of a genetic algorithm. Genetic algorithms (GA's) are adaptive search techniques which have demonstrated substantial improvement over a variety of random and local search methods. The genetic algorithm here is used to compose the feature representation of class ω_0. The composition of the representative of class ω_0 is made averaging *some* out of the N feature vectors. This average representation keeps the within-class distance within reasonable limits. The subset of feature vectors conforming the representative feature vector is selected by the genetic algorithm according to the following general cost function J:

$$J = \min[d(ap,\Omega)], \qquad (1)$$

where: J, is the cost function to be maximized.
d, is a distance measure suitable to the problem to be solved.
ap, is an averaged prototype candidate to be the optimal representation of the feature.
Ω, is the set of all the sample feature vectors in ω_1.

The main problems in applying GA's to any situation

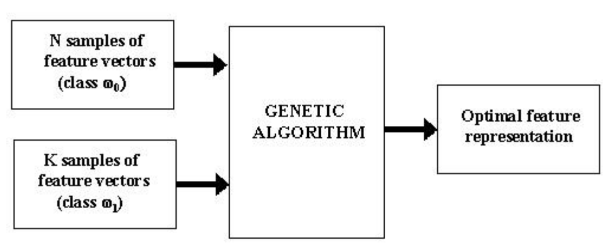

Fig. 3 Optimal feature representation.

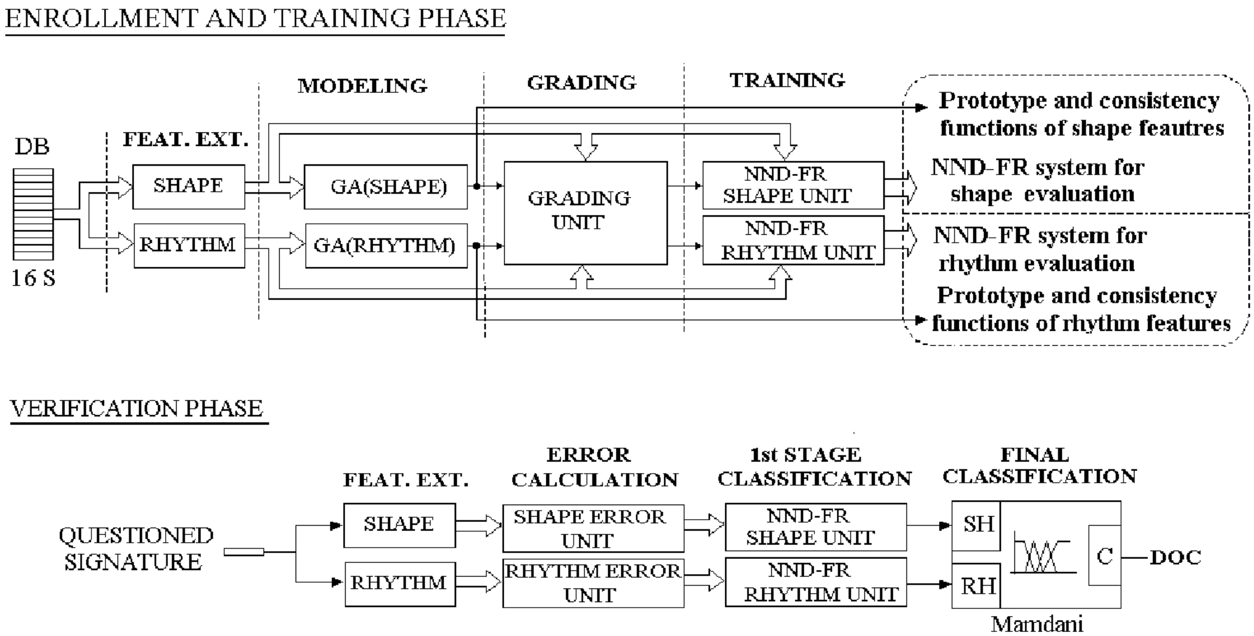

Fig. 4 Architecture and overall approach of the automated verifier with optimal feature representation. The optimal references are calculated in the enrollment and training phase, in the blocks of the modeling stage.

are selecting an appropiate representation and an adequate evaluation function. The evaluation function was stated in (1). The problem of representation here is solved assigning an index number to each sample feature vector of class ω_0. To link the problem to the genetic algorithm, the index is converted to a binary number which is embedded onto the chromosomes in the evolutionary process. Several binary indexes placed side-to-side conform the binary chromosome. A population of chromosomes prepares the genetic algorithm to start the evolutionaty process, and the operations of fitness evaluation, selection, crossover, and reproduction are executed.

III. OFR FOR ON-LINE SIGNATURE VERIFICATION

To test our proposal of OFR, we undertake the problem of on-line signature verification with an approach based on the procedures of an expert examiner of signatures. The expert examiner considers that the signing process is dominated by personalized writing habits [9]. A signature is the result of a complex combination of personalized patterns of shape and rhythm. It is important to remark that it is the final balance of rhythm and form which is critical in signatures comparison, not only the matching of some individual elements [9], or, in our perspective, some individual features. It is important to mention in advance that besides the reference prototype, here we calculate a consistency function and a weighthing factor for each feature.

A. Overall strategy.

Fig. 4 summarizes the procedure of our overall approach. Two phases can be distinguished. In the *enrollment and training* phase, 16 instances of a signature are picked up from an acquired database. Repetition of signatures is allowed if less than 16 exemplars have been donated by the signer; 80 random forgeries (4 genuine from other 20 signers) and 5 synthetic skilled forgeries are also included but not shown in figure 1 for simplicity. The objective of including these few synthetic skilled forgeries is to enable the system to reject forgeries of good quality. More about synthetic skilled forgeries later. Five shape–related and five dynamics-related features are extracted from every signature. Once the two sets of features are calculated the goal is to generate a model for the signer. Morover, an optimized model. The elements of the model are an optimal averaged prototype function *P*, the representative function of class ω_0; a consistency function *C*, and a weighting factor *W*, per feature. To create these elements, a genetic algorithm is used as described in section II. The averaged prototype function must yield a weighted distance or *error* minimum when compared to the respective features derived from genuine signatures (within-class distance), and maximum when compared to the respective features derived from skilled forgeries (between-class distance).

The goal of the grading stage is to grade all signatures, genuines as well as forgeries, in the range 0-1. The distances (or errors) per feature and the grades of every signature are used to train two NND-FR systems, one for shape (form) and one for rhythm (dynamics). As a consequence, each NND-FR system *learns* to grade a questioned signature according to: a) deviations with respect to normal variations of genuine signatures, b) consistency in the signing process of the original signatures, and c) importance or contribution of every feature to a correct verification.

In the *verification* phase, a questioned signature is digitized and its respective shape and rhythm features are computed. The error between every feature and its reference is calculated in the respective error unit (see Fig. 4). The set of shape errors is used to train the NND-FR shape unit. The output of this unit is a grade of shape for the questioned signature. A grade for signature's rhythm is calculated in a similar manner. Finally, a 2D fuzzy system calculates the degree of certainty (DOC) in which a signature should be considered genuine.

In order to reduce the FAR (false acceptance rate) of this system, we use synthetic skilled forgeries. It is equivalent to know in advance the form of potential forgeries, and include this knowledge in the class ω_1 to create a more discriminant reference function. To do so, instead of distorting the image of original signatures (as other researchers do), we distort the waveform of the extracted features to represent the signatures, based on the observation that the features from skilled forgeries are very similar to the features from originals, having random differences in magnitude and phase. To create synthetic features, original features are phase shifted, corrupted by additive noise, and lowpass filtered. Our experiments demonstrated that skilled forgeries are better discriminated if *synthetic skilled forgeries* are used to generate the model functions.

B. Database description and feature extraction.

A signature to be analized is captured using a digitizer tablet and a pen. Data are acquired every 1/128 seconds and are as follows: X,Y coordinates of pen position and absolute pressure, both versus time.

1) *Shape related features.* According to Slyter [9], shape related features should reflect the signing habits of a person in relation to gross forms, variations in signature's design, connective forms and micro-forms [9]. As a consequence, we should derive, from any signature, functions or parameters that reliably reflect the variations and consistency of such elements. This is still an open and difficult problem [4]; however, in [5] it can be found a set of 5 functions that are aimed to characterize the shape of a signature; these functions can discriminate or 'discover' discrepancies in, at least, slant and local shape of a signature, as it is shown in a contrived example given by Nalwa, and are inherently independent of the position, size and orientation of the signature. We adopted these features or characteristic functions as our shape descriptors, given that we are mainly focused in feature representation, not in feature generation. The features are measured over the parameterized length of a polygon fitted to the ordered samples of the on-line signature with respect to a moving coordinate frame and using a gaussian weighting window, and they are listed in table 1. For implementation details please refer to [5]. Original names were preserved for ease of identification.

2) *Rhythm and dynamics related features.* The relevant dynamic features are listed in table 1. Absolute speed is derived from x(t), y(t) data. In order to obtain representations of equal length, the speed function is resampled to have a fixed length of 256. Lets symbolize absolute speed as *Sp(t)*.

Pressure is directly obtained from the digitizer tablet, and the variations in pressure patterns *dP(t)* are calculated by differentiation of absolute pressure. Top *tow(s,t)* and base *bow(s,t)* patterns of writing are obtained sampling the upper and lower envelopes of the signature's digital image. In addition, the description of these patterns is augmented with timing information: to every point belonging to the upper/lower envelopes a time instant is associated to it according to the instant in which the point under consideration was drawn. In this way, if a forger is able to reproduce the visual aspect of a signature with a different trajectory with respect to the original signature this fact becomes evident using this description of the envelopes of a signature. Signature's representation in the features space is shown in table 1.

A total of 5 features are shape-related and 5 are dynamics-related. Once a feature is computed, its length is normalized to a fixed size to all signers, making a point-to-point averaging and comparison across different instances of a signature feasible.

C. Optimal feature representation using GA's.

Genetic algorithms is a tool very little usage in the field of signature verification. In [14], Yang and others used a genetic algorithm to select partial curves along a signature. The features used were length, slope,

Table 1 Signature representation in the features space

	Feature	Meaning	Size
SHAPE	Cx(l)	X coordinate of local center of mass	82
	Cy(l)	Y coordinate of local center of mass	82
	T(l)	Torque exerted about origin of the	82
	S1(l)	Curvature ellipse measure 1	82
	S2(l)	Curvature ellipse measure 2	82
RHYTHM	Sp(t)	Speed as a function of time	256
	P(t)	Pressure as a function of time	256
	DP(t)	Pressure change patterns	256
	Bow(s,t)	Base of writing with timing	84
	Tow(s,t)	Top of writing with timing	84

largest angle, and curvature of partial curves along the signature. In that study, the emphasis was set mostly on the genetic algorithm itself and no verification results were presented. In [15] the authors report basically the same technique, but the optimization is carried out over the virtual strokes (i.e. points of pen-up trajectories) of a signature. Some results were reported. In [16], Wijesoma and others proposed the use of genetic algorithms with no special improvement in the genetic algorithm itself, but with emphasis on the selection of a subset of features out of a set of 24 features representing the shape and dynamics of a signature. Please note that so far the use of genetic algorithms in signature verification have been either to select a subset of features [16] or to segment a signature in partial curves whose features are used for verification [14], [15]. The classifier used in all cases was fuzzy logic. We should now use the genetic algorithm to create a prototype function, a consistency function, and a weighting factor that indicates how much a feature is able to contribute to the purpose of signature verification, in accordance with our optimal feature representation.

Recalling that we consider the problem of signature verification as a problem of pattern recognition with two classes (ω_0, ω_1), the goal of the genetic algorithm here is to produce a prototype function, a consistency function, and a weighting factor for each feature. The class 'genuine' or ω_0 consists of genuine signatures. A prototype function *PF* and its associated consistency function *CF* are constructed averaging a set of N vectors of a feature from N genuine signatures as follows:

$$PF(i) = \frac{1}{N}\sum_{j=1}^{N} F_j(i) \tag{2}$$

$$CF(i) = \frac{1}{\sqrt{N}}\sqrt{\sum_{j=1}^{N}(F_j(i) - PF(i))^2} \tag{3}$$

where PF, prototype function.
CF, consistency function.
N, number of functions to conform the model with $1 \le i \le length(F)$

The 'others' class (ω_1) is created computing the corresponding features from 4 genuine signatures of other 20 signers in the database, plus 5 synthetic skilled forgeries. A synthetic feature is obtained from a genuine feature by exchanging two segments of the vector that contains a feature, adding random noise, and filtering the resulting vector with a LPF IIR filter, with normalized bandpass limit set to 0.8 and a order of 10. Filtering gives some additional phase shifting. Fig. 5 shows genuine and synthetic features. The solid line indicates the feature of an original. A real skilled forgery is drawn in dashed line. Dash-dot line is the synthetic feature that

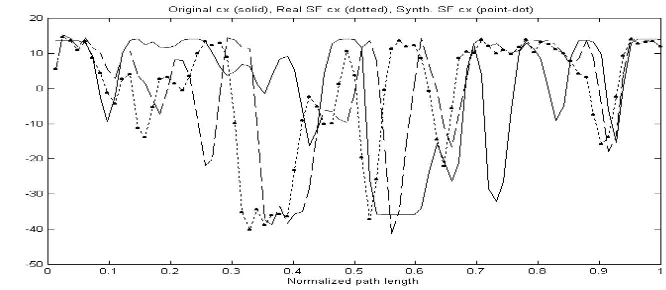

Fig. 5 Solid line: genuine; dashed line: a real skilled forgery; dash-dot line: synthetic skilled forgery in the features space.

acts like a skilled forgery. Please observe the similarity between real forgeries and synthetic skilled forgeries.

At this point two data sets have been described, one for the class 'genuine' (w_0) with N elements and other for the class 'others' (ω_1) with 85 elements. A distance measure between a feature and its corresponding reference is given by (3)

$$d(PF, F) = \sqrt{\sum_{n=1}^{card\{F\}}\left[{(F(n) - FP(n))}\big/{CF(n)}\right]^2} \tag{4}$$

Distance *d* in (4) is a measure of the error in the features space for a single feature between two signatures.

D. Maximizing between-class distance.

From the database, 16 signatures of a signer are picked up to build the model of each feature. Every signature is identified with a four bits binary number. Let be N a number of signatures (4<N<16) that belong at a given moment to the 'genuine' or ω_0 class. A chromosome is coded, using *n* signatures out of the 16 signatures previously chosen, by concatening their respective bit strings, being the subset of *n* signatures randomly selected initially. For example, for *n*=5:

Chromosome codification	Subset of Signatures
'0010|0111|1010|0001|1001'	{2,7,10,1,9}

The repetition of signature's index or identifiers in the bit string is allowed. The initial population of chromosomes is set to the length (measured in bits) of a chromosome plus 4, which is somewhat arbitrary but yields good results. Once the initial population has been created, the evolutionary operators of selection, mating, reproduction, and mutation are carried out over the primitive chromosomes. *Rank-based* selection is used here [17]. To keep only the top performers of each generation the *generational approach* is adopted [17]. Mutation is allowed to occur 4 times during the whole execution of the GA; this selection of mutation occurrence is helpful in jumping out of local minima during the convergence process. The objective function of the GA is stated as follows:

$$t = (\min(\boldsymbol{ied})-\max(\boldsymbol{iad}))/\max(\boldsymbol{ied}) \quad (5)$$

where *ied* –inter-class or between-class distance- and *iad* –intra-class or within-class distance- are the sets of all the distance measures or *errors* between every element of the 'others' and the 'genuine' classes respectively against a reference, as given by (4). The reference is built in every cycle of the GA using (2) and (2), being *n* variable and the reference prototype being created with the signatures whose identifiers are coded in every chromosome. The GA is stopped when the average fitness of the population and the fitness of the fitest chromosome are equal or when the iteration number reachs 45, whatever happens first. The GA is executed 11 times, for *n*=5 to 15, so this process produces 11 prototype and 11 consistency functions (eleven pairs) for a single feature. Every pair of functions are used to evaluate (4) to produce the sets *ied* and *iad*. A pair is chosen to be the prototype and consistency functions according to max(min(*ied*)-max(*iad*)). Fig. 6 shows an example of *ied* and *iad*. Please observe the band separation between the classes 'others' and 'genuine'.

E. Weighting the features.

Lets define the matrix of *shape descriptors* SD = [cx,cy,T,s1,s2] of size 82x5, and *error in shape descriptors* as ESD = (E_{cx},E_{cy},E_T,E_{s1},E_{s2}) of size 5, as descriptors of the behaviour of a signature in the features space. Matrix SD contains the values of the five features and ESD contains the error between each feature and its corresponding optimized model function according to (4). As noted in Fig. 6, there exist a band, or class separation, that is different to each feature of the same signer. Lets consider this class separation as a symptom of how much a feature is relevant to verify the authenticity of a signature and lets assign a weighting factor to each feature as follows:

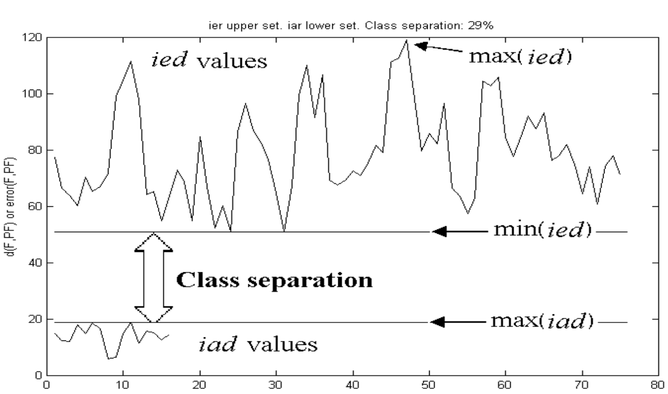

Fig. 6 Class separation as a result of the evolutionary process.

$$wf_f = (\min(\boldsymbol{ied})-\max(\boldsymbol{iad}))/\max(\boldsymbol{ied}) \quad (6a)$$

$$\mathrm{WF} = [\ wf_{cx}\ wf_{cy}\ wf_T\ wf_{s1}\ wf_{s2}\ ] \quad (6b)$$

where *wf* stands for *weighting factor* and WF is a vector of personalized weighting factors. A feature is discarded if (6[a]) results negative. A grade is assigned to a feature considering the standard deviation (σ), mean value (m) of the respective set of *iad* values, and E_f the feature's error with respect to its model, as follows:

$$G_f = \begin{cases} 1 \Rightarrow 0 < E_f \le m - 2\sigma \\ (0.15/_{(max(iad)-m+2\sigma)})(E_f - m + 2\sigma) + 1 \Rightarrow m - 2\sigma < Ef \le max(iad) \\ (0.85/_{(min(ied)-max(iad))})(E_f - max(iad)) + 0.85 \Rightarrow max(iad) < E_f \le min(eid) \end{cases}$$

and putting together all shape related features in a vector G:

$$\mathrm{G} = [G_{cx}\ G_{cy}\ G_T\ G_{s1}\ G_{s2}] \quad (7)$$

a signature receives a grade of shape *GS* according to (8),

$$\mathrm{GS} = \mathrm{G} * \mathrm{WF}^{\mathbf{T}} \quad (8)$$

This is the equation of the Grading Units in Fig. 4. The same process applies to generate a grade of rhythm *GR* for any signature.

To show the efficacy of the genetic algorithm in creating class separation, 241 exemplars are collected including random forgeries, skilled forgeries and genuine signatures. Please note that this collection of signatures includes the signatures used to generate the genuine signature's model as well as other *previously unseen* signatures. For each of these signatures, a 5-dimensional *ESD* vector is calculated so that an 241x5 matrix is created. Lets name this matrix EM or *error matrix*. Applying principal components analysis (PCA) to EM a projection of the original 5-D space onto a 2-D space is made. Fig. 7, left side, shows this projection and discovers an underlying structure of two classes, as expected. Solid diamonds correspond genuine signatures used for training and hollow diamonds are new genuine signatures under test. Intra-personal variability was well absorbed. Some skilled forgeries are pointed. For the same 241 signatures, Fig. 7, right side, shows the evaluation of grade of shape GS, with (7). The left side high values are the grades of the genuine used for training; middle zone values are the grades of random forgeries, the grades for skilled forgeries are pointed and genuine 2 includes grades of genuine signatures under test.

IV. Neural-Network-driven Fuzzy Reasoning (NND-FR) for classification

NND-FR is a family of techniques in which fuzzy reasoning is implemented using neural networks [17]. In

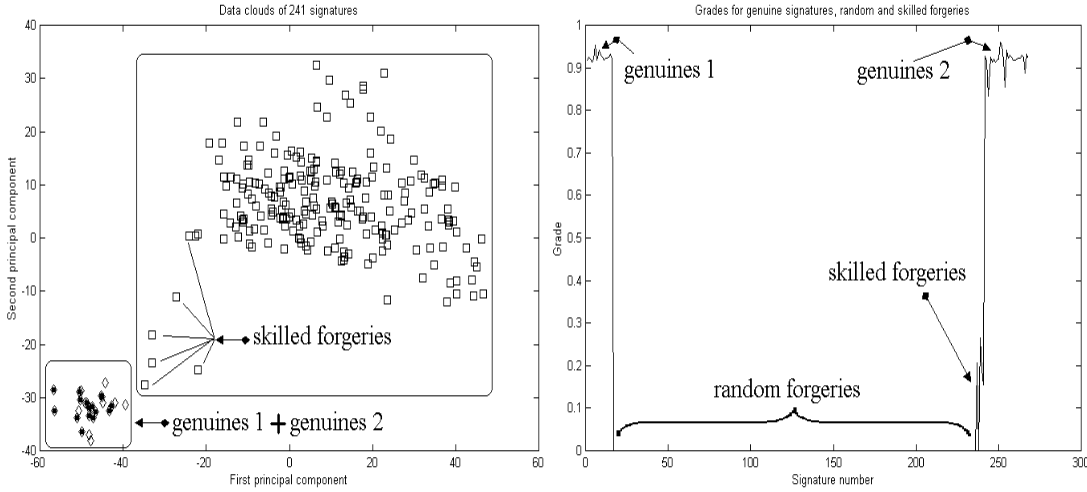

Fig. 7 Left: projection of the two first principal components of features of 241 signatures showing class separability. Right: Grades of the signatures in according to its nature. The group genuines 2 include genuine signatures unseen by the classifier in the training phase.

this work a Sugeno fuzzy system is implemented using neural networks. Fig. 8 shows the block diagram of a NND-FR system. In Sugeno type fuzzy systems rules are typically of the form *if* x_1 *is* A_1 *AND* x_2 *is* A_2 ... *THEN* $y=f(x_1,x_2,...,x_n)$ where f is a function of the inputs $x_1,x_2,...x_n$. The function f is replaced by a neural network, and an induced rule is of the form *if* $(x_1,x_2,...,x_n)$ *is* A^s *THEN* $y^s = NN_s(x_1,x_2,...,x_n)$, where $(x_1,x_2,...,x_n)$ is the vector of inputs, NN_s is a neural network that determines the output y^s (consequent) of the *s*th rule, and A^s is the membership function of the antecedent of the *s*th rule. A^s is generated by a neural network. The topology used here uses 1 fuzzy rule and is illustrated in Fig. 8 for the shape-related features. Note that the inputs to the neural networks are the errors between the features of a questioned signature with respect to the respective models of the genuine signature, defined as ESD. ERD or *error in rhythm descriptors* is used to train the rhythm-related NND-FR system. These two NND-RF systems are those shown in Fig. 4. The training data set of NN_{mem} of the shape-related NND-FR system is given in table 2. Again, consider a similar situation for the rhythm-related NND-FR.

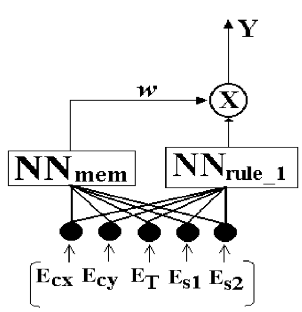

Fig. 8 Topology of the NND-FR

Table 2 Training sets for NND-FR systems

NNmem inputs	**Target**
ESD of genuine	1
ESD of forgeries	0
NNs inputs	**Target**
ESD of genuine	G
ESD of forgeries	G

The error sets ESD and ERD and the grades G in table 2 are those mentioned in subsection III.E, and are derived from the original data sets used to create the reference models.

V. Final decision.

In the *verification* phase (see Fig. 4) a final decision on whether a signature is genuine or a forgery is made by a 2-D fuzzy logic Mamdani system. The universe of discourse of the inputs ranges from 0 to 1, which is the output range of the NND-FR systems. The universe of discourse of the output goes from 0 to 100, and this is the range of values assigned to the output variable DOC –degree of certainty- in which a signature can be considered as genuine. The fuzzy rule base is shown in table 3. It is worth to mention that fuzzy rules are crafted to avoid overemphasize the importance of shape on rhythm and *vice versa*, because, as stated earlier, an expert examiner considers that it is more important the complex balance of rhythm/form than the individual matching of features because it is more difficult to imitate simultaneously both aspects. Membership functions are omitted.

Table 3 Fuzzy rule base. Linguistic variables: VG, very good; G, good; R, regular; B, bad; VB, very bad.

		GRADE OF RHYTHM				
		VG	**G**	**R**	**B**	**VB**
GRADE OF SHAPE	**VG**	VG	VG	G	R	B
	G	VG	G	R	R	B
	R	G	G	R	B	B
	B	R	R	R	B	VB
	VB	R	R	B	VB	VB

VI. SIGNATURES DATABASE AND RESULTS

The database consisted of 923 genuine signatures from 36 signers and 1000 forgeries from 36 forgers. For each signer there are 27 genuine samples and 33 forgery samples. The signatures were collected over a period of 3.5 weeks, thus avoiding the side effects or artifacts in signatures due to exhaustive or continuous efforts. Every signer was allowed to get used to comfortably sign on the tablet before donating any signature. The forgers were advised in advance about all aspects of the signing process of their “victims” and were allowed to practice as much as they considered necessary to forger the victims’ signatures. Every signer had the choice of rejecting a signature if the exemplar was not adequate.

In the training phase, 4 random forgeries, 5 synthetic skilled forgeries and 16 genuine signatures per signer were used. To test the verifier, 210 random forgeries, 30 skilled forgeries, and 27 genuine signatures were used per signer. 7324 verifications were realized. Only 10 classifcation errors occurred. The global **correct verification rate was 99.86%**. FRR (false rejection rate), is the percentage in which the system rejected genuine signatures considering that they are forgeries. It is also known as type I error. For this system and for the database used the FRR was 1.05%. FAR (false acceptance rate) is the percentage in which the system accepted forgeries as if they were genuines. Again, it is also known as type II error. For this system and the database used, it was 0.27%; the average error can be set to 0.66%. In this terms, the efficiency of the system was 99.34%. As can be seen in Fig. 7, if only random forgeries are considered the misclassification rate is 0%.

A question that might arise in the mind of the reader is wether it is or is not a remarkable result in the field of signature verification. Plamondon reports in [4] FRR in values ranging from 1% to 50% with skilled forgeries, and 4 systems with 0%. Nevertheless, it is important to point out that any system achieving 0% in FRR necessarily can not achieve a low value of FAR. This is the case of the 0% FRR reported by Plamondon. With respect to FAR, the range goes from 0.4% to 12%,with 2 with 0%. Again, 0% FAR implies FRR > 0%. In this context, and taking into account that the system was tested with realistic skilled forgeries and that the database used is relatively large, we conclude that the performance is very high.

In relation to the optimal feature representation, although not presented here, in some preliminary results of experiments done without such a representation, it was encountered that exactly the same architecture had a performance very low in comparison to what was attained using OFR. FRR was approximately of 8%, while FAR was 10%. It is quite indicative that OFR is a useful tool to improve the accuracy of any pattern recognition system.

VII CONCLUSIONS

The results presented are very encouraging. As it was shown in this paper, OFR can be added to the typical cycle of design of a pattern recognition system to improve its accuracy to almost 100%. Unfortunately, it must be said that this technique adds computational cost in the training phase. Another drawback of OFR became evident when it was applied to the on-line signaure verification problem. There is no strict control on the within-class distance. We are actually working around this problem and some results indicate that it is posible to control this aspect of the algorithm. It is part of the present and future work related to this subject. Other areas of future work include to test the OFR in signature verification with escalar features, such as global or RMS values of position, speed, acceleration, pressure, and any other variable at the disposal, such the angulation of the pen while the signature is generated.

The approach of the forensic examiner embedded in the architecture of the verifier played a important role too, since it allowed us to asses which specific aspects of the signature failed in the signing process.

The use of weighting factors applied to individual features allowed the system to discover the discrimination power of every feature and its contribution to a correct verification of a questioned signature.

The inclusion of synthetic skilled forgeries in the modeling stage as presented here is novel in this research area. Another new inclusion in this field is the use of top and base patterns of writing associated with the corresponding time instants in which their points were drawn.

REFERENCES

[1] Plamondon R., and Lorette. G., "Automatic signature verification and writer idetification – The state of the art", *Pattern Recognition*, vol. 22, pp. 107-131, 1989.

[2] Plamondon, R., "The design of an on-line signature verification system: from theory to practice", *Progress in Automatic*

Signature Verification, World Scientific Publishing Co. Pte. Ltd., 1994, pp. 795-811.

[3] Plamondon, R. and Srihari, S.N. "Online and off-line handwriting recognition: a comprehensive survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, pp. 63-84, Jan. 2000.

[4] Plamondon R., and Lorette G., "On-line signature verification: how many countries are in the race?", *Proceedings of the 1989 International Carnahan Conference on Security Technology*, pp. 183-191, 1989.

[5] V.S. Nalwa, "Automatic on-line signature verification", *Proceedings of the IEEE*, vol. 85, no. 2, pp. 213-239, Feb. 1997.

[6] Theodoridis, S. and Koutroumbas, K., *Pattern Recognition*, 1st ed. San Diego, Cal. Academic Press, 1999.

[7] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and soft computing*, 1st ed. Upper Sadle River, N.J. Prentice Hall, 1997. 614 pp.

[8] Vafaie, H. and De Jong, K., "Genetic Algorithms as a Tool for Reestructuring Feature Space Representations," Proceeedings of the 1992 International Conference on Tools with AI. Vol. 1. 1-11-1992.

[9] Slyter, A. Forensic Signature Examination. Thomas Publisher. U.S.A. 1995.

[10] Guyon I., and Elisseeff, A. Call for papers and contributions special issue on “Variable and feature selection”. Journal of Machine Learning Research. On-line. www/clopinet.com/isabelle/projects/NIPS2001/call-for-papers.html

[11] Vafaie, H. and De Jong, K. Genetic Algorithms as a Tool for feature selection in Machine Learning. Proceeedings of the 1992 International Conference on Tools with AI, Vol. 1 200-203. 1-11-1992.

[12] Skalak, D. B., "Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms," *International Pattern Recognition* and *Machine Learning Journal.*

[13] Yu, S., De Backer, S., and Schroder, H., "Genetic feature selection combined with composite fuzzy nearest neighbor classifiers for hyperspectral satellite imagery," *Pattern Recognition Letters*, vol. 23, no. 1, pp. 183-190, 2002.

[14] Yang, X., Furuhashi, T., *et al*., "A study on signature verification using a new approach to generic based machine learning", IEEE International Conference on Systems, Man and Cybernetics 1995. Intelligent Systems for the 21st Century, vol. 5, pp. 4383-4386. 1995.

[15] Yang, X., Furuhashi, T., *et al*., "Constructing a High Performance Signature Verification System Using a GA Method", *Proceedings Second New Zeland International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, pp. 170-173, 1995.

[16] Wijesoma, W.S., Ma. Mingming, Sung, E., "Selecting optimal personalized features for on-line signature verification using GA", *2000 IEEE International on Systems, Man, and Cybernetics*, vol. 4, pp. 2740-2745, 2000.

[17] Tsoukalas, L.H., and Uhrig, R. E., "Fuzzy and Neural Approaches in Engineering", John Wiley and Sons, Inc. N.Y. U.S.A. 1997.