Offline Geometric Parameters for Automatic Signature Verification Using Fixed-Point Arithmetic

Miguel A. Ferrer, Jesús B. Alonso, and Carlos M. Travieso

Abstract—This paper presents a set of geometric signature features for offline automatic signature verification based on the description of the signature envelope and the interior stroke distribution in polar and Cartesian coordinates. The features have been calculated using 16 bits fixed-point arithmetic and tested with different classifiers, such as hidden Markov models, support vector machines, and Euclidean distance classifier. The experiments have shown promising results in the task of discriminating random and simple forgeries.

Index Terms—Automatic Signature Verification (ASV), Hidden Markov Models (HMM), Support Vector Machines (SVM), fixed-point arithmetic.

1 Introduction

SIGNATURE verification is an important research area in the field of person authentication. We can generally distinguish between two different categories of verification systems: online, for which the signature signal is captured during the writing process, thus making the dynamic information available, and offline for which the signature is captured once the writing processing is over and, thus, only a static image is available [1], [2].

The objective of the signature verification system is to discriminate between two classes: the original and the forgery, which are related to intra and interpersonal variability [3]. There are three different types of forgeries to take into account. The first, known as random forgery, is usually represented by a signature sample which belongs to a different writer of the signature model. The second, called simple forgery, is represented by a signature sample which has the same shape as the genuine writer's name. The last type is so-called skilled forgery, represented by a suitable imitation of the genuine signature model [4].

Each type of forgery requires a different verification approach [5]. Methods based on the static approach are generally used to identify random and simple forgeries. The reason for this is that these methods have proven to be more suitable for describing characteristics related to the signature shape. A skilled forgery has practically the same shape as the genuine signature. Therefore, methods based on dynamic (online) or pseudodynamic approaches have been shown to be more robust for identifying this kind of forgery [6]. Hybrid systems have also been developed [7].

In this paper, we propose new geometrical features for an offline signature verification system (ASV). The proposed features can be calculated with a fixed-point microprocessor. Therefore, the features can be extracted from inside a personal device such as a smart card. The robustness of the proposed features for random and simple forgeries is tested out with different classifiers, such as hidden Markov models, support vector machines, and Euclidean distance classifier. The last two classifiers can also be programmed using fixed-point arithmetic.

The paper is organized as follows: Section 2 introduces the features proposed. Section 3 is devoted to the classifiers. Section 4 reports the experimental results and the paper ends with concluding remarks.

• *The authors are with the Departamento de Señales y Comunicaciones, Universidad de Las Palmas de Gran Canaria, Campus de Tafira, E35017 Las Palmas de Gran Canaria, Spain.*
E-mail: {mferrer, jalonso, ctravieso}@dsc.ulpgc.es.

Manuscript received 16 July 2004; revised 28 Feb. 2005; accepted 1 Mar. 2005; published online 14 Apr. 2005.
Recommended for acceptance by M. Pietikainen.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0359-0704.

2 Feature Extraction

The geometrical features proposed by this paper are based on two vectors which represent the envelope description and the interior stroke distribution in polar and Cartesian coordinates.

2.1 Outline Detection and Representation

The outline is calculated by means of morphological operations as is shown in Fig. 1: First, we apply a dilatation in order to reduce the signature variability and, afterward, a filling operation is applied to simplify the outline extraction process. When several objects are detected after filling, a horizontal dilatation is performed until all the objects are connected.

The outline is represented as a sequence of its Cartesian coordinates $(X_t, Y_t)_{t=1}^{T}$, T being its length. This sequence follows the contour counterclockwise and starts in the point $(X_1, Y_1) = C_x, \max(Y_t \mid_{X_t = C_x})$, (C_x, C_Y) being the geometric center of the outline (see Fig. 2).

2.2 Feature Vector Based on Polar Coordinates

To represent the signature outline in polar coordinates, it is decimate selecting T_r equidistant samples of the envelope $(X_{t\cdot p}, Y_{t\cdot p})_{t=1}^{T_r}$ (being $p = fix(T/T_r)$ and fix rounds to the nearest integers toward zero) and represent each sample as a three components feature vector, which are: the derivate of the radius, its angle, and the number of black pixels that the radiuses cross when sweeping from one selected point to the next. The latter components have been obtained with an algorithm designed for a fixed-point microprocessor.

The radius function r_t, $t = 1, 2, \ldots, T_r$, is calculated as the number of pixels from the geometric center (C_x, C_y) to each outline selected point $(X_{t\cdot p}, Y_{t\cdot p})_{t=1}^{t_r}$ as:

$$\begin{aligned} d_1 = X_{t\cdot p} - C_x, d_2 = Y_{t\cdot p} - C_y \\ r_t = \max(d_1, d_2) + \min(d_1, d_2)/4. \end{aligned} \tag{1}$$

The radius sequence is normalized to $r_1 = 127$. And, the first component of the feature vector, the derivate of the radius, is obtained as $\Delta r_t = r_{t+1} - r_t$, $t = 1, 2, \ldots, T_r$ being $r_{T_r+1} = r_1$. We do not use the radius because the probability density functions of the radius of different signatures are very similar. The density of the radius derivate is more discriminative. In our experiment, we verified that a $T_r = 64$ value is a good trade-off between the recognition ratio and computational requirements.

The second component of the feature vector is the angle of each selected contour sample, which is calculated by means of the arctan function implemented through lookup table (see (2)):

$$\theta_t = \arctan\left(X_{nT/T_r} / Y_{nt/T_r}\right), t = 1, 2, \ldots, T_r. \tag{2}$$

The third component contains the number of black pixels of the signature strokes that the radius crosses when sweeping from θ_t to θ_{t+1} normalized to maximum value equal to 1 in order to increase the stroke thickness independence. This new component, denoted as A_t, $t = 1, 2, \ldots, T_r$, completes the information about the outline covered by the first and second components with a rough idea about the distribution of the signatures' strokes inside the outline.

Therefore, the sequence of feature vectors based on polar coordinates that characterize the signature is $p_t = [\Delta r_t, \theta_t, A_t]$,

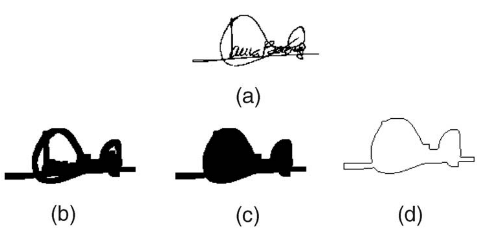

Fig. 1. (a) Original. (b) Dilated. (c) Filled. (d) Outline of the signature.

$t = 1, 2, \ldots, T_r$. Experimentally, we chose $T_r = 64$. A example can be seen in Fig. 3.

2.3 Feature Vector Based on Cartesian Coordinates

The second feature vector is also based on the envelope and the signature strokes density parameterization, but, in this case, in Cartesian coordinates. The envelope is divided through the geometric center into top and bottom halves. We measure the height of the top half at T_h equidistant points, obtaining the sequence uh_t, $t = 1, 2, \ldots, T_h$. We also measure the bottom half height at the same T_h points, getting the sequence lh_t, $t = 1, 2, \ldots, T_h$.

Next, we divide the envelope in two halves once again, this time the left and right-hand sides through the geometric center. T_w equidistant measures of the width of the right and left-hand sides are taken, obtaining the sequences rw_t, $t = 1, 2, \ldots, T_w$, and lw_t, $t = 1, 2, \ldots, T_w$.

The above sequences are combined, obtaining $T_h + T_w$ long sequences as follows:

$$ur_t = \begin{cases} uh_t & t = 1, \ldots, T_h \\ rw_{t-T_h} & t = T_h + 1, \ldots, T_h + T_w \end{cases}$$

and

$$ll_t = \begin{cases} lh_t & t = 1, \ldots, T_h \\ lw_{t-T_h} & t = T_h + 1, \ldots, T_h + T_w. \end{cases}$$

Both sequences characterize the signature envelope shape. We have used $T_h = 42$ and $T_w = 22$ in our experiments, looking for a trade-off between recognition ratio and computational load.

Once both sequences have been obtained, the feature vector sequence is composed of four dimensional vectors. The first component of the feature vector is the sequence ur_t, $t = 1, 2, \ldots, T_h + T_l$, the second component is the values of ll_t, $t = 1, 2, \ldots, T_h + T_l$. The third feature vector component is the value of the index t in order to help the HMM synchronism. The fourth component is the sequence tr_t, $t = 1, 2, \ldots, T_h + T_l$, defined as

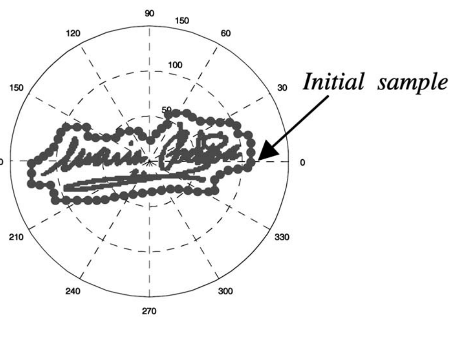

Fig. 2. Original signature, its outline, and its selected samples.

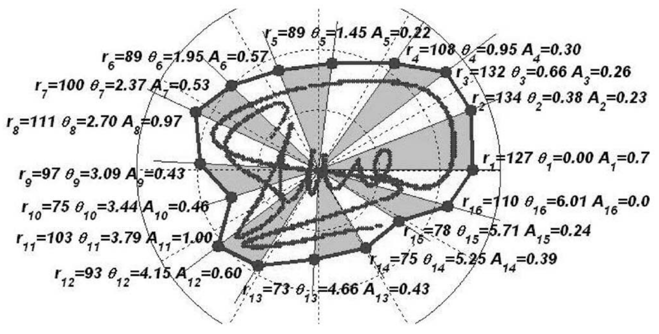
r_5=89 θ_5=1.45 A_5=0.22
r_6=89 θ_6=1.95 A_6=0.57
r_4=108 θ_4=0.95 A_4=0.30
r_3=132 θ_3=0.66 A_3=0.26
r_7=100 θ_7=2.37 A_7=0.53
r_2=134 θ_2=0.38 A_2=0.23
r_8=111 θ_8=2.70 A_8=0.97
r_1=127 θ_1=0.00 A_1=0.7
r_9=97 θ_9=3.09 A_9=0.43
r_{16}=110 θ_{16}=6.01 A_{16}=0.0
r_{10}=75 θ_{10}=3.44 A_{10}=0.46
r_{15}=78 θ_{15}=5.71 A_{15}=0.24
r_{11}=103 θ_{11}=3.79 A_{11}=1.00
r_{14}=75 θ_{14}=5.25 A_{14}=0.39
r_{12}=93 θ_{12}=4.15 A_{12}=0.60
r_{13}=73 θ_{13}=4.66 A_{13}=0.43

Fig. 3. The signature and its envelope with the values of r_t, θ_t, and A_t associated to each selected sample on the envelope with $T_r = 16$.

$$tr_t = \begin{cases} th_t & t - 1, \ldots, T_h \\ tw_{t-T_h} & t = T_h + 1, \ldots, T_h + T_w, \end{cases}$$

where th_t, $t = 1, 2, \ldots, T_h$, contains the number of transitions (black to white or white to black) originated by the signature strokes when we go from the bottom side to the top side of the signature envelope following the vertical lines used to work out uh_t or lh_t. Similarly, tw_t, $t = 1, 2, \ldots, T_w$, contains the number of transitions when we go from the right-hand side to the left-hand side of the signature envelope following the horizontal lines used to work out rw_t or lw_t. These sequences are gathered in the sequence of vectors $c_t = [ur_t, ll_t, t, tr_t]$, $t = 1, 2, \ldots, T_h + T_l$. We have used $T_h = 42$ and $T_w = 22$ sequence. These sequences are illustrated in Fig. 4.

3 Classifiers

To take advantage of both feature vectors, we have to define a fusion technique. The fusion can occur at the feature vector level, at the match score level, or at the decision level [8]. With each classifier, we have made the fusion at the level which reported the best results.

3.1 The HMM Signature Model

In this case, a discrete HMM is chosen to model each signer's features [9]. This avoids making assumptions on the form of the underlying distribution, especially when there is a limited amount of data, as is the case in signature verification. In this case, the best results have been obtained with the fusion at the score level, so a signature is modeled by two left-to-right HMM, one per p_t sequence and another one for the c_t sequence. The number of states in each signer's HMMs signature is 35. The topology only authorizes transitions between each state to itself and to its immediate right-hand neighbors. The classification (evaluation), decoding, and training problems are solved with the Forward-Backward algorithm, the Viterbi algorithm, and the Baum-Welch algorithm. The initialization method is the equal-occupancy method [10]. The K-Means algorithm is used during training to create the multilabeling VQ used by us, which makes a soft decision about which code words are closest to the input vector [11]. Therefore, our VQ generates an output vector whose components indicate the relative closeness of the 10 closest code words to the input.

Additionally, our VQ permits multiple observations training because an incoherence can arise if one assumes the independence property among the different components of the input D-dimensional feature vector [12]. The components of the feature vector p_t are considered independent, so the three sets of observation symbols contain 32 symbols. The four components of the feature vectors c_t are considered in three independent groups: $[ur_t, ll_t]$, t, and tr_t. Experimentally, we have determined 32 symbols for the first group and 16 symbols for the second and third groups.

Fig. 4. The signature, its envelope, and sequences th_t, uh_t, lh_t, tw_t, rw_t, lw_t with $T_h = 8$ and $T_w = 5$.

This number of symbols was judged a good trade-off between the number of centers describing the space of the signers and the number of observation vectors collected in the training database of the signers.

To verify a signature, the log likelihood of the two HMM that model the signature is obtained. The fusion of both scores can be done by regarding the problem as a classification or a combination problem [8]. With regard to this latter point of view, several strategies for combining multiple classifiers have been suggested. As it is well-known that a simple sum rule is sufficient to obtain a significant improvement in the matching performance of a multimodal biometric system [8], the log likelihoods of each HMM are added up to obtain the final score. The usual normalization techniques necessary to consolidate the scores are not necessary because both HMM have the same number of states and both feature vectors are of the same length. If the final score is greater than a threshold, the signature is accepted.

The HMM software used is the GPDShmm toolbox which is freely available from http://www.gpds.ulpgc.es/download/index.htm [13].

3.2 Support Vector Machine Signature Model

We have also used for modeling a signature the technique of Support Vector Machines (SVM) [14] because it provides generally better generalization performance when the amount of data is small and because of its robustness to the quantization effects with respect to fixed-point math implementations [15].

Roughly, the principle of SVMs relies on a linear separation in the feature space where the data have been previously mapped in order to take into account the eventual nonlinearities of the problem. In order to achieve a good level of generalization performance, we maximize the distance (margin) between the separator hyperplane and the data. In the Structural Risk Minimization principle, Vapnik has proven that maximizing the margin means, in fact, minimizing the VC-dimension of the linear separator model, which has been shown to be a good way to reduce the generalization risk. To generalize the linear case, one can project the input space into a higher-dimensional space in the hope of a better training class separation. In the case of SVM, this is achieved by using the so-called kernel trick. In essence, it replaces the inner product used to calculate the distance between the input and the separator hyperplane with a kernel function K. Among the commonly used kernel functions are the polynomial kernel and the RBF kernel.

The software used to train and test the model is the SVMlight, which can be downloaded free of charge from http://www.kernel-machines.org/software.html. The gamma variable is optimized experimentally for each experiment [16]. The input vector sequence is accomplished by concatenating two compatible feature sets $[p_t, c_t]$, $t = 1, 2, \ldots, 64$. In this case, we have built the fusion at the data or feature level instead of at the score level, as it was in the HMM classifier case, because, experimentally, it has reported better results.

To verify a sequence, the SVM of the signature calculates the distance of the input sequence to the separator hyperplane. If the distance is greater than a threshold, the signature is accepted.

3.3 Euclidean Distance-Based Signature Model

Looking for a simple method to verify the signatures, we propose using a verifier based on Euclidean distance. Its advantages come from the minimum time it takes and the reduced requirements it needs to accept or reject the input signature. The signatures are represented by the feature vector sequence $[p_t, c_t]$, $t = 1, 2, \ldots, 64$. The sequence of vectors that model the signature is obtained as the average of the training sequences of the genuine signature. Dynamic time wrapping algorithms were checked and discarded because they also reduce interclass variability. To verify a given signature, the mean Euclidean distance between the input sequence and the average sequence that modeled the signature is calculated. If the distance is greater than a threshold, the input signature is accepted. The Euclidean distance is calculated as in (1).

4 The Evaluation Protocol

4.1 Signature Database

The database we built and used in the experiments of this paper contains data from 160 individuals: 24 genuine signatures for each individual plus 30 forgeries of his/her signature. The repetitions of each genuine signature and forgery specimen were collected using black or blue ink on white A4 sheets of paper featuring two different box sizes: the first box is 5 cm wide and 1.8 cm high and the second box is 4.5 cm wide and 2.5 cm high. Half of the genuine and forgery specimens were written in each type of boxes. The 24 genuine specimens of each signer were collected in single day writing sessions.

The forgeries were produced under the following conditions: The forger imitates a genuine signature from the static image of the genuine signature (scanned at 300dpi) and the forger is allowed to practice writing the signature for as long as s/he wishes. Each forger has to imitate three signatures of five signers in a single day writing session. The genuine signature shown to each forger is chosen randomly from the 24 genuine ones. Therefore, for each genuine signature, there are 30 simple forgeries made by 10 forgers from 10 different genuine specimens. The papers with signatures were acquired with a standard scanner with 75 dpi resolution in an 8-bit gray scale image. A sample of a genuine signature and simple forgeries can be seen in Fig. 5.

All the signature images in black and white and noise cleaned were saved in PNG format. A freely distributed version of the described database is available from: http://www.gpds.ulpgc.es/download/index.htm.

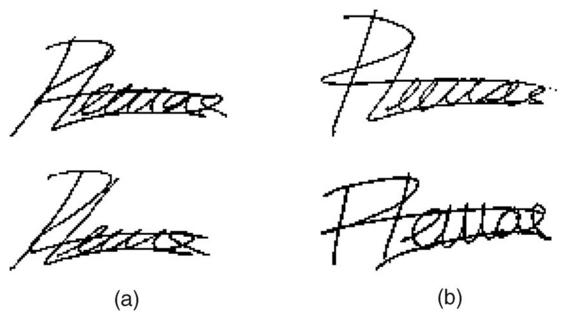

Fig. 5. (a) Genuine signatures. (b) Simulated simple forgeries.

4.2 Threshold Decision

For each signer, classifier learning is done only on genuine signatures. The classifier approximates, during the verification phase, the score of the signature. Signature verification for signer i was performed using a decision threshold $\tau(i)$. A decision is taken about whether the supposed signature of signer i is authentic or forged by comparing the score to its decision threshold.

Two types of error are characteristic of the signature verification problem: Error I or false rejection rate (FRR), which is when an authentic signature is rejected, and Error II or false acceptance rate (FAR), which is when a forgery is accepted. The level of these two error rates depends on the decision threshold chosen; we look for a decision threshold whose value is such as to realize the Equal Error Rate EER (FAR = FRR).

The decision threshold consists of two terms: $\tau(i) = l(i) + \bar{x}$, where $l(i)$ is the decision threshold obtained at the point where the FRR curve estimated with the genuine signatures used for training and the FAR curve estimated with some forgeries cross, and $\bar{x}$ is an offset common to all the signers because the FRR curve obtained with the training sequences is biased (see Fig. 6). To estimate the FAR curve, with some forgeries, we use, in the random forgeries experiment, the other training genuine signatures and, in the simple forgeries experiment, we estimate the FAR curve with three simple forgeries discarded for test purposes.

The offset $\bar{x}$ is obtained as follows: We estimate the FRR curve with genuine signatures for training, the FRR curve with genuine signatures for testing, and the FAR curve for forgeries for 10 randomly selected signers of the database. These signers are

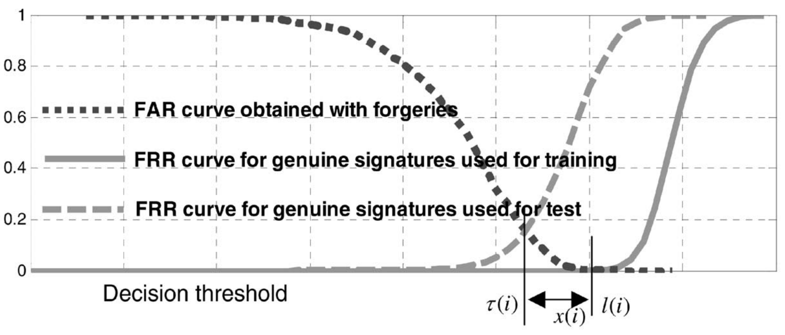

Fig. 6. FRR curves with training and test sequences and FAR curve for simple forgeries.

TABLE 1
Signature Verification Result (Percentage) for Random and Simple Forgeries with the HMM

		Random forgeries		Simple forgeries	
		Error Type I	Error Type II	Error Type I	Error Type II
Number of training genuine signatures	4	$4.3_{1.8}$	$3.8_{1.6}$	$17.3_{5.2}$	$14.9_{3.9}$
	8	$2.5_{1.9}$	$2.4_{1.2}$	$13.4_{2.6}$	$14.9_{3.5}$
	12	$2.2_{1.4}$	$3.3_{1.1}$	$14.1_{3.7}$	$12.6_{3.0}$

TABLE 2
Signature Verification Result (Percentage) for Random and Simple Forgeries with the SVM Trained with 12 Samples

		Random forgeries		Simple forgeries	
		Error Type I	Error Type II	Error Type I	Error Type II
Kernel	Lineal	$4.27_{2.1}$	$3,71_{1.3}$	$21.06_{3.8}$	$18.54_{3.1}$
	Polynomial	$3,65_{2.1}$	$3,15_{1.7}$	$15.41_{2.2}$	$15.64_{1.8}$
	RBF	$3.23_{0.8}$	$2.65_{1.3}$	$15.41_{1.4}$	$13.12_{1.7}$

TABLE 3
Signature Verification Result (Percentage) for Random and Simple Forgeries with the Euclidean Distance Trained with 12 Samples

		Random forgeries		Simple forgeries	
		Error Type I	Error Type II	Error Type I	Error Type II
Number of training genuine signatures	8	$6.16_{0.36}$	$5.92_{0.45}$	$17.29_{0.85}$	$18.25_{0.78}$
	12	$5.56_{0.34}$	$5.13_{0.41}$	$16.21_{0.67}$	$15.66_{0.68}$
	16	$5.61_{0.65}$	$4.96_{0.58}$	$16.39_{1.18}$	$15.50_{1.18}$

removed for test purposes. For each signer, the offset is obtained (see Fig. 6). The common offset is worked out as:

$$\bar{x} = \frac{1}{10} \sum_{i \in 10 \; signers \; selected} x(i).$$

This procedure improves the similar one proposed in [17].

4.3 Experiments and Results

With a view to demonstrating the performance of the proposed geometric parameters worked out using fixed-point arithmetic, we provide results with the different classifiers. Two experiments have been carried out: the first with random forgeries and the second one with simple forgeries. The genuine and forgery (for threshold calculation) training samples will be chosen randomly from the database set and the test will be performed with the other genuine and forged samples. In order to obtain reliable results in each studied case, the training and test procedure was repeated 10 times with different randomly chosen training and test sets. The average and variance of the obtained ratios are given. The numbers with subscripts on the tables represents the average and the standard deviation. As 10 signatures are chosen to obtain the offset, the results will be given by averaging the results of the remaining 150 signatures.

Tables 1, 2, and 3 show the results with the HMM, SVM, and Euclidean distance classifiers. As can be seen, the best results are obtained with the HMM classifier and the worst with the Euclidean distance classifier. Nevertheless, the Euclidean distance classifier is trained more rapidly and we can implement it in a fixed-point microprocessor very easily. If the Euclidean classifier uses the Mahalanobis distance, the results are improved by a half point. The EER point estimation is considered good because the Error Types I and II are very similar.

5 Conclusion and Further Research

This paper proposed a method to calculate geometric features of a signature in fixed-point arithmetic for offline verification. The proposed features have been checked with different classifiers, such as Hidden Markov Models, Support Vector Machines, and the Euclidean distance verifier. The results show that HMM works slightly better than SVM and the distance Euclidean verifier, but, bearing in mind that the SVM and Euclidean distance-based verifiers can be programmed in a fixed-point microprocessor, the results encourage us to follow the SVM research line in order to built a smart card capable of detecting a simple forgery.

Obviously, the chosen parameters do not take into account all the characteristics used by a forensic document examiner. For instance, features reflecting the pseudodynamics of the signature are missing [18], etc. In a necessary further analysis, our work will focus on the use of such parameters with fixed-point arithmetic.

Other further research concerns the database. Automatic signature verification is highly sensitive to signature stability and the variability of the signatures depends on country, age, time, habits, psychological or mental state, and physical and practical conditions [19]. Since our database has collected the signatures of each writer with his/her ballpoint pen in a daily session, some causes of variability have not been taken into account in this paper.

Acknowledgments

This paper has been produced with the support of a TIC2003-08956-C02-02 grant from the Spanish Government.

References

[1] F. Leclerc and R. Plamondon, "Automatic Signature Verification: The State of the Art 1989-1993," *Int'l J. Pattern Recognition and Artificial Intelligence,* special issue on signature verification, vol. 8, no. 3, pp. 643-660, June 1984.
[2] R. Plamondon and S.N. Srihari, "Online and Offline Handwriting Recognition: A Comprehensive Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22 , no 1, pp. 63-84, Jan. 2000.
[3] M.C. Fairhurst and E. Kaplani, "Perceptual Analysis of Handwritten Signatures for Biometric Authentication," *IEE Proc. Vision, Image and Signal Processing,* vol. 150, no. 6, pp. 389-394, Dec. 2003.
[4] J. Edson, R. Justino, F. Bortolozzi, and R. Sabourin, "An Off-Line Signature Verification Using HMM for Random, Simple and Skilled Forgeries," *Proc. Sixth Intl Conf. Document Analysis and Recognition,* pp. 1031-1034, Sept. 2001.
[5] J. Edson, R. Justino, F. Bortolozzi, and R. Sabourin, "The Interpersonal and Intrapersonal Variability Influences on Off-Line Signature Verification Using HMM," *Proc. XV Brazilian Symp. Computer Graphics and Image Processing, 2002,* pp. 197-202 Oct. 2002.
[6] J. Edson, R. Justino, A. El Yacoubi, F. Bortolozzi, and R. Sabourin, "An Off-Line Signature Verification System Using HMM and Graphometric Features," *Proc. Fourth Int'l Workshop Document Analysis Systems (DAS 2000),* pp. 211-222, Dec. 2000.
[7] A. Zimmer and L.L. Ling, "A Hybrid On/Off Line Handwritten Signature Verification System," *Proc. Seventh Int'l Conf. Document Analysis and Recognition 2003,* vol. 1, pp. 424-428, Aug. 2003.
[8] A. Ross and A.K. Jain, "Multimodal Biometrics: An Overview," *Proc. XII European Signal Processing Conf.,* pp. 1221-1224, Sept. 2004.
[9] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition.* Prentice Hall, 1993.
[10] M.A. Ferrer, I. Alonso, and C. Travieso, "Influence of Initialization and Stop Criteria on HMM Based Recognizers," *Electronics Letters of IEE,* vol. 36, pp. 1165-1166, June 2000.
[11] J. Hernando, C. Nadeu, and J.B. Mariño, "Speech Recognition in a Noisy Environment Based on LP of the One-Sided Autocorrelation Sequence and Robust Similarity Measuring Techniques," *Speech Comm.,* vol. 21, pp. 17-31, 1997.
[12] L. Xiaolin, M. Parizeau, and R. Plamondon, "Training Hidden Markov Models with Multiple Observations—A Combinatorial Method," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22 , no. 4, pp. 371-377, Apr. 2000.
[13] S. David, M.A. Ferrer, C.M. Travieso, J.B. Alonso, "gpdsHMM: A Hidden Markov Model Toolbox in the Matlab Environment," *CSIMTA, Complex Systems Intelligence and Modern Technological Applications,* pp. 476-479, Sept. 2004.
[14] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schälkopf, "An Introduction to Kernel-Based Learning Algorithms," *IEEE Trans. Neural Networks,* vol. 12, no. 2, pp. 181-201 Mar. 2001.
[15] D. Anguita, A. Boni, and S. Ridella, "A Digital Architecture for Support Vector Machines: Theory, Algorithm, and FPGA Implementation," *IEEE Trans. Neural Networks,* vol. 14, no. 5, pp. 993-1009, Sept. 2003.
[16] L.E. Martínez, C.M. Travieso, J.B. Alonso, and M.A. Ferrer, "Parameterization Method of a Forgery Handwritten Signature Verification System Using SVM," *Proc. 38th IEEE Int'l Carnahan Conf. Security Technology,* pp. 193-196, Oct. 2004.
[17] M. Fuentes, S. Garcia-Salicetti, and B. Dorizzi, "On Line Signature Verification: Fusion of a Hidden Markov Model and a Neural Network via a Support Vector Machine," *Proc. Eighth Int'l Workshop Frontiers in Handwriting Recognition, 2002,* pp. 253-258, Aug. 2002.
[18] J.-J. Brault and R. Plamondon, "Segmenting Handwritten Signatures at Their Perceptually Important Points," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 9, pp. 953-957, Sept. 1993.
[19] R. Plamondon, "The Handwritten Signature as a Biometric Identifier: Psychophysical Model and System Design," *Proc. European Convention Security and Detection, 1995,* pp. 23-27, May 1995.