

# Simplified Stroke-based Approach for Off-line Signature Recognition

*A. Pérez-Hernández, A. Sánchez and J.F. Vélez*

Departamento de Informática, Estadística y Telemática  
Universidad Rey Juan Carlos  
Campus de Móstoles  
28933 Móstoles (Madrid), SPAIN

alex@radiotbo.com, {an.sanchez, jvelez}@escet.urjc.es

## Abstract

This paper proposes a simple adaptive off-line signature recognition method based on the feature analysis of extracted significant strokes for a given signature. Our system correctly decides on the majority of tested patterns, which include both simple and skilled forgeries. The presence of possible doubtful signatures (those ones on which is difficult to decide) is also considered. Experimental results have shown a good trade-off between response time and reasonable accuracy of recognition results.

## 1. Introduction

Signatures are a special case of handwriting subject to intra-personal variation and inter-personal differences. This variability makes it necessary to analyse signatures as complete images and not as collection of letters and words [5]. Human signatures provide secure means for authentication and authorisation in legal and banking documents; therefore the need of research in efficient automatic solutions for the involved signature recognition and verification problems has increased [10]. In the signature recognition or identification problem, a given signature is searched in the database to establish the signer's identity. Signature verification problem is concerned to determine if a particular signature is authentic or a forgery. Techniques for solving both the recognition and verification problems can be classified as on-line and off-line [1]. In the first ones, data are obtained using an electronic tablet and other devices and in the second ones, images of signatures written on a paper are scanned and dynamic information is not available.

Many approaches for the automatic off-line verification problem have been reported in the literature [1][6][8][11]. In general, the proposed techniques use either a type of features (global, local, statistical, geometric, etc) or a combination of different types of features, extracted from the signature images. In particular, Fang et al [4] use a stroke-based method that approximates the strokes in the signature skeleton by fitting a set of short lines with similar lengths. However, the off-line signature recognition problem has received little attention despite the fact that it is of interest in

areas like law enforcement and commerce. The major previous work is [9] which uses a set of topologic features to characterize each signature.

This paper introduces a simple and efficient off-line approach that can be applied to signature recognition. The technique is based on an approximate feature extraction of relevant component strokes for a given test signature. Each extracted stroke is represented by a set of only three features: its two endpoints and its global orientation. This information is efficiently compared with the corresponding patterns stored in the signature database, using an Euclidean feature matching approach, to find the most similar one to the query. The corresponding degree of similarity is also computed.

Our approach also considers the practical involved problems and requirements described in automatic verification systems, such as: lack of training samples (we only use two signatures per writer to train the system), variability of signature patterns (due to intra-personal or inter-personal variations), presence of forgeries (simple and skilled ones), robustness to moderate noise, and an acceptable response time.

The proposed system can also be adapted for different Internet applications (i.e. e-commerce). It could be possible to recognize a registered user for Internet purchases using his/her signature. A client-server solution is now needed. The signature scanning and some preprocessing to extract the component signature strokes can be performed at the client's side, and the recognition task using the database of signatures is preformed at the server's side.

The paper is organized as follows. Section 2 offers a high-level description of the proposed off-line signature recognition system. In section 3, the signature preprocessing stage is described. Section 4 explains how the signature features (strokes) are extracted and the related problems to perform this task. Section 5 outlines the signature recognition method based on stroke matching. The signature database and the verification experiments are described in Section 6. Finally, Section 7 presents the conclusions.

## 2. Proposed off-line signature recognition system

The overall architecture of our signature recognition system appears in Figure 1. The input to the system is a scanned query image signature. The output is the identity of the signer (if the signature is recognized with an acceptable confidence degree in the database), a negative response about the presence of the signer in the database, or a classification of this signature as doubtful (thus requiring a more complex analysis).

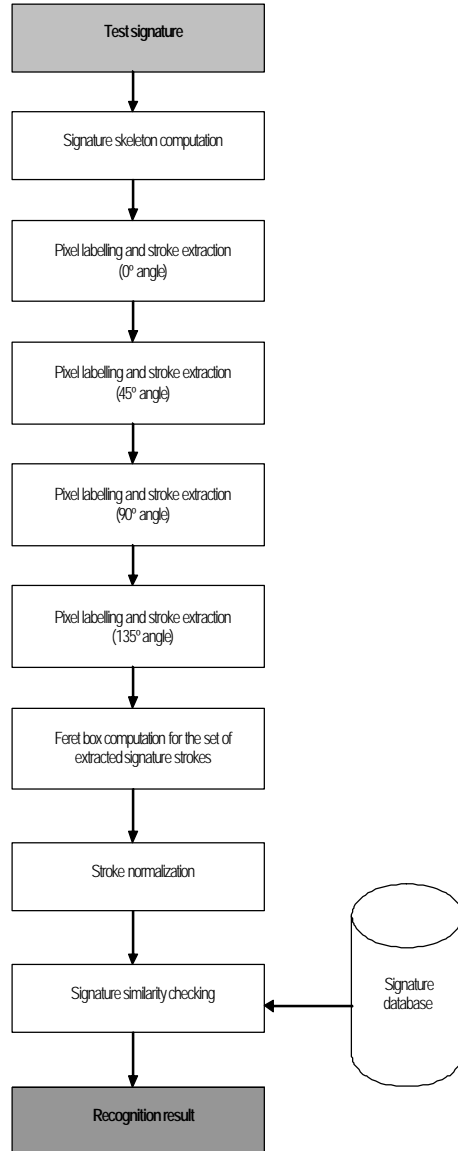


Figure 1: Block diagram of the proposed signature recognition system.

General processing stages of the proposed recognition system are:

- *preprocessing*: where a signature image is scanned, and the corresponding signature skeleton is obtained.
- *feature extraction*: which requires determining the orientations of component signature pixels (angles of 0°, 45°, 90° or 135°, respectively), the signature stroke extraction through pixel tracking, and a final stroke normalization process; and,
- *recognition*: performed by a simple stroke distance matching method.

These steps are explained with more detail in the next sections.

## 3. Preprocessing stage

Initially, grey-level signature images were scanned with a resolution of 200 dpi, converted to binary images by thresholding and stored in BMP format.

To reduce the impact of pen thickness used when signing, and to simplify the structural shape of signatures, they are thinned to obtain the corresponding signature skeletons. This process is carried out by the application of a typical thinning method proposed by Zhang and Suen [12]. This algorithm repeatedly deletes contour pixels by respecting certain conditions until a 1-pixel wide 8-connected skeleton is obtained. The algorithm alternates between two phases, each selecting pixels for deletion based on different criteria. Figure 2 shows the result of Zhang and Suen's method on a sample signature.

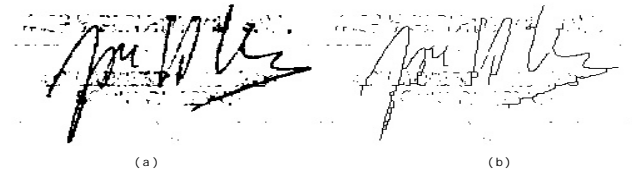


Figure 2: (a) Original noisy signature and (b) its corresponding skeleton.

As it is visible in the previous example, this signature skeleton extraction algorithm is robust to the presence of a moderate noise.

## 4. Feature extraction

This task requires an initial pixel labelling process according to some predetermined orientations. Then, a pixel-tracking algorithm using the estimated prearranged pixel orientations is applied. As a result of this feature extraction, the set of strokes for a given signature is obtained. Finally, a stroke normalization process is carried out prior to the recognition stage.

### 4.1. Pixel labelling.

The initial pixel labelling process considers four predetermined directions: 0°, 45°, 90° and 135°, respectively. This task is performed by the application of

the respective 2x2 convolution masks (associated to these directions) to each pixel in the signature skeleton. The obtained visual results for a random test signature are shown in Figure 3. As it can be noticed, a same pixel can be initially labelled as belonging to more than one possible direction. These labelling conflicts are solved during the tracking stage.

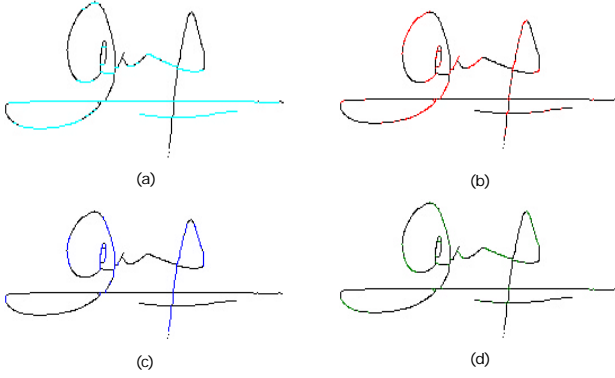


Figure 3: Result of pixel labelling process on a sample signature for the respective directions (angles) of: (a) 0°, in cyan; (b) 45°, in red; (c) 90°, in blue; and (d) 135°, in green.

We limit the possible pixel directions to the above four ones. This simplification has a practical advantage because it permits a certain degree of flexibility when signing with respect to slight stroke variations and it also makes possible a faster signature processing.

#### 4.2. Pixel tracking.

The aim of pixel tracking process is to extract the component strokes of a given signature. The proposed tracking method is similar to the iterative version of connected components algorithm [7][2] for binary images. This algorithm has been adapted to the specific aspects of our signature problem. As we consider four predominant angles, the pixel tracking algorithm is independently applied four times for each set  $a_i$  of labelled pixels (where  $i \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ ) in the four different orientations.

A systematic priority pixel tracking ordering is needed. Signature images are scanned by rows and the following traversal ordering has been used:

$$\{SW, W, S, SE, NW, E, N, NE\}$$

where the capital letters  $N$ ,  $S$ ,  $E$  and  $W$ , respectively represent the north, south, east and west directions, and a pair of letters represents the corresponding group direction (i.e.  $SW$  represents the south-west).

Consequently, each of the four considered sets of pixels  $a_i$  will adopt its corresponding traverse ordering list as shown in Table 1. Note that 0°-list requires two traverse orderings depending on whether a stroke is scanned from left to right (direction  $E$ ) or from right to left (direction  $W$ ). Another remarkable detail is that 45° and 135° angles only consider seven searching directions

instead of eight. This is to avoid possible traverse conflicts between opposite  $NE$  (in 45°) and  $SW$  (in 135°) directions.

Table 1. Traverse ordering sequence for pixels in lists  $a_i$  ( $i \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ )

Angle	Traverse ordering sequence
0° (W)	{W, SW, NW, N, S, NE, SE, E}
0° (E)	{E, NE, SE, N, S, SW, NW, W}
45°	{SW, W, S, SE, NW, E, N}
90°	{S, SE, SW, E, W, NE, NW, N}
135°	{SE, E, S, SW, NE, W, N}

Some problems could happen during the pixel tracking process. For example, it is possible to have intermediate unlabeled pixels that clearly are part of a given stroke (where the surrounding pixels are labelled as belonging to this stroke). We solved this problem by searching in the neighbourhood of border-labelled pixels (at a distance of about 6 pixels) other pixels labelled as border. If this happens, then the intermediate pixels are also classified as being part of the same stroke.

Another source of conflicts is stroke-crossing pixels. Due to the previous stage of signature skeleton construction, a “step effect” in some stroke-crossing pixels is possible (see Figure 4). This problem can be solved through a systematic search procedure in a small interest region around the crossing point.

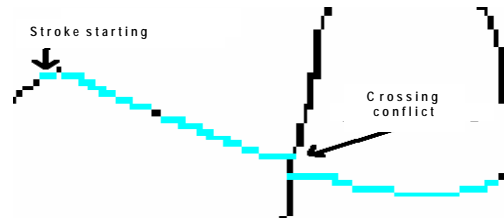


Figure 4: Stroke crossing conflict.

#### 4.3. Stroke normalization.

The application of the proposed pixel tracking method produces a list of strokes describing a signature. Each component stroke of this list is represented by three elements: its original angle of inclination or orientation with respect to the paper (codified with a colour code as represented in Figure 3), the source (upper) stroke endpoint and the destination (lower) stroke endpoint. Therefore, the set of strokes of a given signature will be ordered from left to right and from top to down as appearing in the image. This property can reduce the complexity of comparing two signatures using their corresponding stroke lists.

Signature strokes are later normalized to make invariant the recognition task with respect to signature sizes. Normalization is accomplished by computing the Feret box [2] of the set of extracted strokes for a given signature. The Feret box consists of the smallest rectangle (oriented according to the co-ordinate axis) that

encloses all the signature strokes. This bounding structure can be computed with a complexity of  $O(n)$ , where  $n$  represents the number of extracted strokes. After obtaining the Feret box for a signature, stroke points are transformed according to the following algorithm pseudocode:

```

for each  $s_i \in S$  do
  for each  $p(x,y) \in s_i$  do
     $x' = \left\lfloor \frac{p.x - p_1.x}{d_x} \times 100 \right\rfloor$ 
     $y' = \left\lfloor \frac{p.y - p_2.y}{d_y} \times 100 \right\rfloor$ 
  end
end;

```

where:  $s_i$  represents any component stroke of signature  $S$ ,  $p(x,y)$  is any point belonging to  $s_i$ ,  $p_1$  and  $p_2$  are respectively the upper leftmost and lower rightmost points of the computed Feret box,  $d_x$  and  $d_y$  are respectively the horizontal and vertical dimensions of the Feret box, and  $x'$  and  $y'$  are the respective new transformed co-ordinates corresponding to point  $p(x,y)$ . The notation  $p.x$  ( $p.y$ ) refers to the  $x$ -value ( $y$ -value) of the point  $p$ . Finally, the multiplications by 100 when computing the respective values of  $x'$  and  $y'$  are performed because all the signatures have been normalized to fit into a region of  $100 \times 100$  points.

Figure 5 illustrates the complete normalization stage. As it can be noticed, this process will produce some inevitable distortions in the strokes that could be contracted or stretched. Note that these distortions will produce a lack of invariance with respect to the orientations of the signature strokes. However, this effect will not affect significantly to the result of signature recognition task because the original inclination angles of component strokes are used for recognition and not the corresponding ones of normalized strokes.

## 5. Recognition using stroke distance matching

After applying the complete preprocessing and feature extraction stages, each signature is now represented as an ordered list of strokes, which are independent from the original signature size and thickness. The recognition process consists of two phases: training and test, respectively.

The aim of the training stage is to experimentally adjust two system parameters: 1) the distance threshold value  $d$  among two compared signature stroke endpoints to decide if the stroke appears in the compared signature; and 2) the smallest valid stroke-length  $ml$ , which is necessary to avoid that inherent noise or slight signature variations could change the recognition results. These two experimental parameters are obtained as result of the system training and will be used during the tests. In this way, a linear separation of the involved signature training classes is achieved.

During the recognition phase, a given signature is compared with all stored signatures (database) to retrieve the most similar one to the test signature according to some similarity or distance measure. Different methods to compare pairs of ordered lists of features (strokes) are available [3]. Feature matching methods assume that each pattern is represented by a (weighted or non-weighted) set of independent features. Symbolic matching methods also consider the relations among features and the patterns are represented as graphs. To take advantage of the proposed simplified signature representation, a simple Euclidean stroke matching method is used. Thus, each list component stroke of a test signature is compared with the strokes of a database signature by only using the two endpoints and the original angles of both strokes. The decision of whether or not a signature stroke appears in another signature (with which is compared to) is next explained. If other stroke in the compared signature with the same angle (color) exists and the minimum distance between the corresponding stroke endpoints is smaller than the distance threshold value  $d$ , then a signature stroke appears in the compared signature.

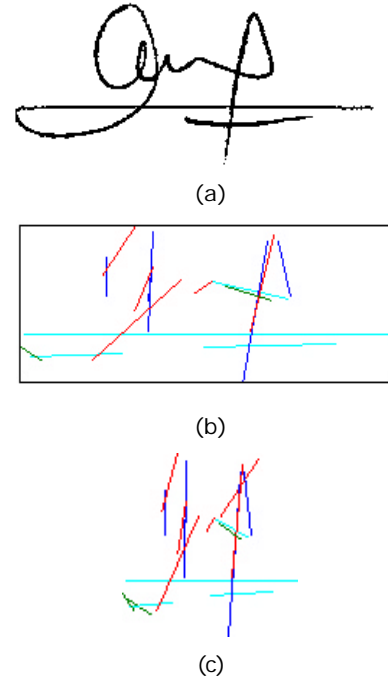


Figure 5: Normalization stage: (a) original image signature, (b) stroke extraction and Feret box computation (rectangle drawn with solid black lines), and (c) stroke normalization result.

Simple heuristics are used to simplify and make more efficient the stroke comparison task between two signatures. First, if the number of strokes is quite different (i.e. a difference greater than 13), then both signatures are considered very different and no stroke matching is computed (the similarity degree is directly set to 0.0). Second, the reference database signatures are stored as stroke lists and not as their corresponding

image bitmaps. This will reduce the actual memory and time system requirements.

A simple and efficient method to compute the similarity between two compared signatures is given by the ratio of strokes in the first signature that are present in the second signature, and vice-versa. The stroke-based similarity degree  $sim_{t,j}$  between a test  $t$  and a database signature  $j$  is computed as:

$$sim_{t,j} = \min(\frac{mat_{t,j}}{s_t} \times 100, \frac{mat_{j,t}}{s_j} \times 100) \quad (1)$$

where:  $mat_{t,j}$  represents the number of stroke coincidences (or matchings) between the test  $t$  and a database signature  $j$ ,  $mat_{j,t}$  represents the number of matchings between a database signature  $j$  and the test signature  $t$ , and  $s_t$  and  $s_j$  are respective number of component strokes in the test and in a given database signature.

## 6. Experimental results

This section describes the performance of our signature recognition system. A prototype recognition system has been implemented using Borland Delphi 6 on an AMD Athlon 4 processor at 1.2 GHz. Signatures were produced using different types of pens, scanned with a resolution of 200 dpi and stored in BMP format. Figure 6 shows some used training signatures corresponding to three different writers.



Figure 6: Some sample training signatures used in the experiments.

The results presented in this section were based on two disjoint sets of signatures: training and test sets. Training set consisted on twenty signatures (two samples of ten different writers). As previously explained, the training phase allowed to experimentally determine the best values of parameters:  $d$  (minimal distance threshold value among two compared signature stroke endpoints) and  $ml$  (smallest valid stroke length). Figure 7 shows the best computed confusion matrix, as result of the system training, to achieve linear separation of the involved signature training classes (using a classification threshold of 51%). The associated experimentally-computed training parameter values (measured in number of pixels) were  $d=20$  and  $ml=15$ , respectively.

	1	1	1	2	2	2	2	3	3	3	3	4	4	4	5	5	5	6	6	6	7	7	7	8	8	8	9	9	9	10	10	10	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 7: Confusion matrix as result of the training stage.

Recognition results are based on a set of 134 signatures. This sample has been partitioned into two disjoint sets: 50 signatures are in the reference database set and 84 in the test set. Signatures of the reference database were collected under no constraints. Test set includes 50 original signatures (corresponding to writers present in the reference database) and 34 forgeries (28 are simple forgeries and 6 are skilled forgeries) for evaluating the system. Recognition rates are controlled by a threshold parameter (percentage)  $RT$  that serves to achieve a balance among accepted, rejected and difficult-to-classify (or doubtful) signatures. This last signature group is not “decided” by our system but it is passed to a more accurate automatic recognizer or to a human expert. Tables 2, 3 and 4 present the experimental results for the different groups of test signatures using the respective values of  $RT$  equal to 51%, 53%, and 55%. For each of these three  $RT$  values, the interval of doubtful patterns has been set to  $[RT-1, RT+1]$  in the experiments

Table 2. Recognition rates for test signatures with  $RT=51$ .

	Correct (%)	FAR (%)	FRR (%)	Doubtful (%)
Genuine	100.0	0.0	0.0	0.0
Simple forgeries	96.4	3.6	0.0	0.0
Skilled forgeries	66.6	33.4	0.0	0.0
Total signatures	96.4	3.6	0.0	0.0

Table 3. Recognition rates for test signatures with  $RT=53$ .

	Correct (%)	FAR (%)	FRR (%)	Doubtful (%)
Genuine	98.0	0.0	0.0	2.0
Simple forgeries	96.4	0.0	0.0	3.6
Skilled forgeries	66.8	16.6	0.0	16.6
Total signatures	95.2	1.2	0.0	3.6

Table 4. Recognition rates for test signatures with  $RT=55$ .

	Correct (%)	FAR (%)	FRR (%)	Doubtful (%)
Genuine	96.0	0.0	2.0	2.0
Simple forgeries	100.0	0.0	0.0	0.0
Skilled forgeries	83.4	16.6	0.0	0.0
Total signatures	96.4	1.2	1.2	1.2

According to experimental results we conclude that our system can be easily adapted to the requirements of a practical recognition application by appropriately setting the value of the parameter  $RT$  and the related interval of doubtful signatures.

It is also interesting to remark that our approach has been tested with different types of forgeries [5]: random forgeries, produced without knowing either the name of the signer nor the shape of the signature; simple forgeries, produced knowing the name of the signer but without an example of his/her signature; and skilled forgeries, produced by looking at an original instance of the signature and attempting to imitate it. Figure 9 illustrates the differences between a simple forgery (correctly classified by the system as rejected) and a skilled forgery (incorrectly classified by the system, and false positive).



Figure 9: (a) Genuine signature, (b) simple forgery, and (c) skilled forgery.

The average recognition time for each of the 84 test signatures was 276 ms, and the recognition time interval ranged from 62 ms (best recognition time) and 547 ms (worst recognition time) using the considered database of 50 signatures.

The lack of a standard international signature database is a big problem for a fair performance comparison as pointed out in reference [4]. Moreover, the signature recognition problem has not been sufficiently studied in the literature. We have only used the results by Sethi and Han [9], based on a set of geometric and topological signature features, for comparison since they use a similar number of test patterns. In their approach, the average recognition rate (also depending on a threshold value) is about 90%. For our data, recognition rate is around 95% and we also have tested with skilled forgeries.

## 7. Conclusions

A novel high-level adaptive off-line signature recognition method based on the analysis of extracted signature strokes is proposed. This approach can be used as a front-end recognition filter which decides on the most easy-to-analyze signature patterns and would filter the most difficult ones to a more sophisticated automatic

recognition system (or to a human expert). The implemented system also provides a good trade-off between short response time (the average recognition time for a signature included in the test set is 276 ms), and reasonable correct recognition results (more than 95% of the patterns in the experiments). Robustness on moderate capture noise and invariance to geometrical transformations is also achieved (with respect to the signature size and signature displacements).

Our future work will include an analysis (and addition) of new computed stroke features to improve the recognition results. Other source of improvement when comparing two signatures is to take advantage of the local ordering in the list of signature strokes as well as the classification of the signatures in the database by their number of strokes. This would reduce the actual recognition time. Finally, we also propose the adaptation of our signature recognition method to possible Internet applications as explained in Section 1.

## 8. References

- [1] Bajaj, R., and Chaudhury, S., "Signature Verification using Multiple Neural Classifiers", *Pattern Recognition*, Vol. 30, No. 1, pp. 1-7, 1997.
- [2] Costa, L. da F., and Cesar, R.M., *Shape Analysis and Classification*, CRC Press, 2001.
- [3] Duda, R.O., Hart, P.E., and Stork, D.G., *Pattern Classification*, 2<sup>nd</sup> Edition, John Wiley & Sons, 2001.
- [4] Fang, B., et al, "Off-line signature verification by the tracking of feature and stroke positions", *Pattern Recognition*, Vol. 36, pp. 91-101, 2003.
- [5] Justino, E.J., Bortolozzi, F., and Sabourin, R., "The interpersonal and intrapersonal variability influences of offline signature verification using HMM", *Proc. XV Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2002)*, Fortaleza (Brazil), 2002.
- [6] Mizukami, Y., et al, "An off-line signature verification system using an extracted displacement function", *Pattern Recognition Let.*, Vol. 23, pp. 1569-1577, 2002.
- [7] Parker, J.R., *Algorithms for Image Processing and Computer Vision*, John Wiley & Sons, 1997.
- [8] Ramesh, V.E., Narasimha Murty, M., "Off-line signature verification using genetically optimized weighted features", *Pattern Recognition*, vol. 32, pp. 217-233, 1999.
- [9] Sethi, I.K., and Han, k., "Use of Local Structural Association for Retrieval and Recognition of Signature Images", *Proc. SPIE*, Vol. 2420, pp. 125-136, 1995.
- [10] Suen, C.Y., Lu, Q., and Lam, L., "Automatic recognition on handwritten data on cheques - Fact or fiction?", *Pattern Recognition Letters*, Vol. 20, pp. 1287-1295, 1999.
- [11] Vélez, J.F., Sánchez, A., and Moreno A.B., "Robust Off-line Signature Verification using Compression Networks and Positional Cuttings", *Proc. 2003 IEEE Workshop on Neural Networks for Signal Processing*, Vol. I, pp. 627-636, 2003.
- [12] Zhang, T.Y., Suen, C.Y., "A fast parallel algorithm for thinning digital patterns", *Communications of the ACM*, Vol. 27, pp. 236-239, 1984.