Cursive Character Recognition by combining Describing Features, Slalom Method and Daubechies Wavelet

L. K. Toscano, J. H. Sossa, R. Barron, G. Sanchez

Centro de Investigación en Computación – IPN
Av. Juan de Dios Batiz, esquina con Miguel Othón de Mendizábal
Ciudad de México, 07738, México.
Contact: likatome@hotmail.com and hsossa@cic.ipn.mx

(Paper received on July 10, 2007, accepted on September 1, 2007)

Abstract. In many applications, characters are written directly by a person. In these cases, the main problem is the wide variability of these characters. In a strict way we should consider them as random signals. To handle this variability many Artificial Intelligence based tools have been proposed, among them neural networks, support vector machines. Following a different strategy, in this paper, Splines are used to adjust to a finite set of samples obtaining as a result a representative pattern of the trace as an optimal set of the nodes of the Spline. We then use wavelets to decompose the initial samples of the trace into its components: approximation and detail. In this paper we consider only the approximation part due to for recognition purposes it is relevant. As we will see an enhancement is obtained by normalizing the optimal nodes. In the experiment section we show the entire enhancement obtained.

1 Introduction

The handwriting character movement is caused by an act of intentional muscular force and joints elasticity. In order to recognize handwritten characters, it is necessary to extract the features related to this movement. Conventional manuscript character recognition is based on the feature extraction from the character shape under analysis. These features can be the lines inclination, the relative position of each line, the length of the different parts of the line, and so on [1]. For example this approach can be used to recognize efficiently non-cursive characters. However, for cursive characters, this approach is not well suited.

One reason why we humans are able to read and understand cursive characters (very aerodynamic or deformed) is because somehow we have the ability to mentally trace several times the letter in the order it was written. When a person writes a character, generally realizes 4 steps, which are:

(a) In the mind, the person imagines the character symbol that he wants to write,
(b) His brain transmits the movement order to his muscle and joints,
(c) He realizes a series of movements according to character writing order,
(d) The image character is made in consequence of three steps (a) to (c).

© H. Sossa, R. Barrón and E. Felipe (Eds.)
Special Issue in Neural Networks and Associative Memories
Research in Computing Science 28, 2007, pp. 69-80

The character generation process is made from the step (a) to (d), while the recognition process can be performed in inverse order, its means from step (d) to (a). However, the inverse process for character recognition is quite difficult.

As it is known, character recognition can be done off-line and on-line [1]. On-line character recognition requires the physical presence of the person that writes the character. Features used in this case are the pen pressure, the trace speed, the trace directions sequence and others [1]. On-line character recognition demands the realization of steps (c) to (a), while off-line character recognition is equivalent to the inverse realization of all steps from (d) to (a). Thus off-line character recognition is a part of the complete inverse on-line manuscript character recognition process. In the past time several manuscript character on-line recognition systems have been proposed. For example, in [2] the authors used a neural network to recognize cursive isolated characters. In [3] a study where the Laplace transform and a second order lineal model that takes the writing velocity as a variable control was used to synthesize the inverse process.

In [4], the authors combined well-known HHMs and dynamic programming for the cursive characters recognition. Character segmentation and its recognition are performed by this combination, 91% efficiency recognition for the English characters was obtain.

In this paper, the main proposal is focused on the inverse writing realization steps from (c) to (a). Steps (c) and (b) are based on the approximation given by a Spline function which it is possible to obtain the movement order, required to perform the character trace using a digitizing tablet. The process from step (b) to (a) is carried out likening the generated models for each character; finally a tree layer neural network is used to train the feature vectors from the optimal knots.

2 Proposed System

The proposed system consists in a feature extraction and recognition stages respectively. In the feature extraction stage, the optimal knots sequence for each character are obtained, which are significant points of the handwritten character. Then using these points, the handwritten character shape can be reconstructed [4]. In this stage, a natural Spline function (SLALOM method) and Steepest Descent method are applied repeatedly until with 20 knots be sufficient to reconstruct the character with minimum error. These 20 optimal knots are used as a feature vector in the recognition stage, using a three-layer Backpropagation neural network. Figure 1 shows the proposed system structure.

2.1 Data acquisition

Data from each character is obtained by means of a digitizing tablet. For writing an ergonomic pencil Intous 2 of Wacom was used. With this pencil people was asked to write the characters on the tablet. This allows knowing the order of articulation of each character. From the tablet we can get an image of the written character and the data as how the character was written. Figure 2 shows, for example, the data acquired for character "h" and its signal in the *x* and *y* axes.

Online handwritten character acquisition from the digitizer tablet

↓

Signal Approximation

↓

Size, Position and Time Normalization

↓

Optimal knots Detection (Slalom, Steepest Descent)

↓

Character Recognition using ANN (Backpropagation)

Fig. 1. Proposed system structure

2.2 Database Construction

To build the database of characters to be recognized, the 26 letters of the English alphabet were used. For each character 50 samples were obtained. In the case of this research, all samples were obtained from one writer. The database contains thus 1,300 samples. From the total of samples, 910 were used to obtain the describing models and 390 were used for testing. In other words, from the 50 samples of each character, 35 were used for model construction and 15 for testing. Figure 3 shows one sample of characters: "a", "n", "m" and "o".

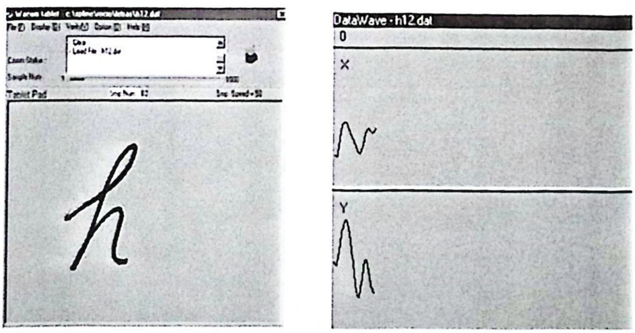

Fig. 2. Captured data from the trace of character "h". (a) Trace. (b) Obtained signals in the *x* and *y* axes.

2.3 Signal Approximation

When the signal of the character is acquired, it is accompanied with some high frequency noise produced by the small vibrations introduced by the movement of the hand. To reduce this kind of noise a Daubechies 1 wavelet was applied. Only the approximation part was taken. Figure 4 (a) shows the original trace of a sample of character "e" and the one obtained (Figure 4 (b)) by wavelet processing the character as explained.

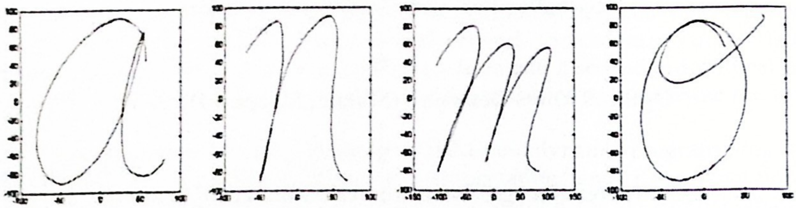

Fig. 3. A sample of the trace of characters: "a", "n", "m" and "o".

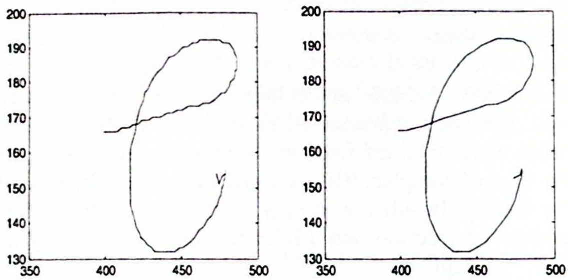

Fig. 4. (a) Original trace of character "e" and (b) its corresponding trace after wavelet approximation.

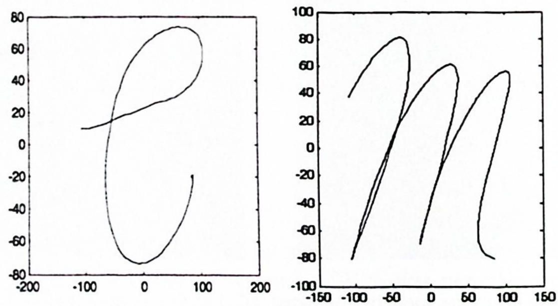

Fig. 5. Traces of letters "e" and "m" after filtering and normalization.

2.4 Post-Filtering and Normalization

To yet reduce high frequency components introduced by the vibration of the hand, a low pass Butterworth filter of order 5 was applied to the character traces in both the x and the y axes. After low pass filtering the character, it is normalized in position, size and time. Normalization in position and size is performed by an affine transformation of the character. Normalization in time is done by interpolation and decimation. Figure 5 shows the traces of the characters "e" and "m" after filtering normalization.

2.5 Feature Extraction

Feature extraction, in the content of this work consists on obtaining the optimal nodes from the signals in the x and y axes of the traced characters. To accomplish this, the SLALOM method well described in [5] was used. The general schema of the application of the SLALOM method to obtain the optimal points of the characters trace is shown in Fig. 6.

SLALOM METHOD

The Slalom method is one type of natural Spline function which must satisfy the following two conditions:

1. The difference between the Spline function $g(x)$ and a given f_i (i=1...M) must be smaller than a previously determined value, δ.

$$|g(x) - f_i| \leq \delta \qquad i=1,2,....,M \tag{1}$$

2. The Spline function g(x) must be a smooth function, that does not need to cross over for every given points f_i (i=1...M).

Figure 5 shows the smooth Spline function g(x) generated by Slalom method and sampling points f_i (i=1...M).

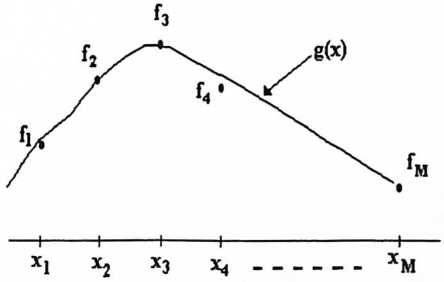

Fig. 6. Smooth function.

To satisfy the two conditions given above, an error function J[g] must be minimized. We can define the first derivative J'[g] of the error function J[g] as Eq. (2).

$$J'[g] = \int \left(\frac{d^2}{dx^2} g(x) \right)^2 dx + \alpha \sum_{i=1}^{M} (g(x_i - f_i)^2 dx \qquad (2)$$

and the first and second derivative of g(x) correspondent to i+1th knot in discrete way, which is written as:

$$\dot{g}_{i+1} = \frac{g_{i+1} - g_i}{x_{i+1} - x_i} = \frac{g_{i+1} - g_i}{\Delta} \qquad (3)$$

$$\dot{g}_{i+1} = \frac{1}{\Delta} \cdot \left(\frac{g_{i+1} - g_i}{\Delta} \frac{g_i - g_{i-1}}{\Delta} \right) = \frac{g_{i+1} - 2g_i + g_{i-1}}{\Delta^2} \qquad (4)$$

where Δ is the interval between the i-th and the i+1-th knots. Supposing that intervals between two consecutive knots is equal to 1, we have

$$\dot{g}_{i+1} = g_{i+1} - 2g_i + g_{i-1} \qquad (5)$$

Next, Eq. (2) can be rewrite using as,

$$J'[g] = \sum_{j=2}^{N-1} (g_{j+1} - 2g_j + g_{j-1})^2 + \alpha \sum_{i=1}^{M} (g_{j_i} - f_i)^2 \qquad (6)$$

where N is the samples number and M is the number of knots, g_i is the i-th value from Spline function g(.) and g_{ji} is the equivalent value of g(.) of the i-th knot.

The minimization problem of J'[g] can be solve as follows.

$$\frac{\partial J'}{\partial g_k} = 0, \qquad k = 1 \ldots N \qquad (7)$$

By substituting Eq. (6) into Eq. (7), we can get,

$$\frac{\partial J'}{\partial g_1} = 2(g_1 - 2g_2 + g_3) + 2\alpha\delta_{1,\Omega}(g_1 - f_1) = 0$$

$$\frac{\partial J'}{\partial g_2} = 2(-2g_1 + 5g_2 - 4g_3 + g_4) + 2\alpha\delta_{2,\Omega}(g_2 - f_2) = 0$$

$$\frac{\partial J'}{\partial g_3} = 2(g_1 - 4g_2 + 6g_3 - 4g_4 + g_5) + 2\alpha\delta_{3,\Omega}(g_3 - f_3) = 0$$

$$\vdots$$

$$\frac{\partial J'}{\partial g_{N-2}} = 2(g_{N-4} - 4g_{N-3} + 6g_{N-2} - 4g_{N-1} + g_N) + 2\alpha\delta_{N-2,\Omega}(g_{N-2} - f_{N-2}) = 0$$

$$\frac{\partial J'}{\partial g_{N-1}} = 2(-2g_{N-3} + 5g_{N-2} - 4g_{N-1} + g_N) + 2\alpha\delta_{N-1,\Omega}(g_{N-1} - f_{N-1}) = 0$$

$$\frac{\partial J'}{\partial g_N} = 2(g_{N-2} - 2g_{N-1} + g_N) + 2\alpha\delta_{N,\Omega}(g_N - f_N) = 0 \tag{8}$$

to get the g_k, for $k = 1, 2, \ldots, N$, the following lineal equations must be resolved

$$\begin{bmatrix} 1 & -2 & 1 & & & & & \\ -2 & 5 & -4 & 1 & & & & \\ 1 & -4 & 6 & -4 & 1 & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & & 1 & -4 & 6 & -4 & 1 \\ & & & & 1 & -4 & 5 & -2 \\ & & & & & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ \vdots \\ g_{N-2} \\ g_{N-1} \\ g_N \end{bmatrix} = \begin{bmatrix} \alpha\delta_{1,\Omega}(f_1 - g_1) \\ \alpha\delta_{2,\Omega}(f_2 - g_2) \\ \alpha\delta_{3,\Omega}(f_3 - g_3) \\ \vdots \\ \alpha\delta_{N-2,\Omega}(f_{N-2} - g_{N-2}) \\ \alpha\delta_{N-1,\Omega}(f_{N-1} - g_{N-1}) \\ \alpha\delta_{N,\Omega}(f_N - g_N) \end{bmatrix} \tag{9}$$

where δ is the sampling space and $\delta_{j,\Omega}$ satisfies,

$$\delta_{j,\Omega} = \begin{cases} 0 & j \notin \Omega \\ 1 & j \in \Omega \end{cases} \tag{10}$$

$\delta_{j,\Omega} = 0$, when j is different from any knot position and $\delta_{j,\Omega} = 1$, when j corresponds with the some knot position.

After the Slalom method is applied and optimal knots are obtained, the error E, can be obtained by Eq. (11).

$$E = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - g(x_i))^2 \tag{11}$$

To minimize the error E, the steepest descent method is applied. The steepest descent method is the simplest gradient methods used to minimize a given error function. Then the optimal knot's position update is given by Eq. (12).

$$x_{k+1} = x_k - \lambda_k \nabla f(x_k) = x_k - \lambda_k g(x_k) \tag{12}$$

After the adjustment of all knots position is performed, we analyze the distance of all consecutive knots.

Optimal node initialization. To the signals, both in the *x* and *y* axes, of the trace of a character a second derivative is applied. The use of the second derivative allows determining the change of velocity at the moment the trace is done. However at it is known the application of the second derivative amplifies the small changes in the signal, giving as a result a noisy signal. To reduce the influence of this introduced noise, the derived signals (in both axes) are smoothed again by means of a low pass filter. Figure 7 shows the processed signals of letter "m" and the second derivative of these signals, respectively. The derived signals are divided into segments according to the number of zero crossings. In this case, the value of "0" in the second derivative means that the change of velocity is zero. For each segment, the data with maximal absolute value are considered as the initial nodes. Figure 8 shows the initial nodes of letters "m" and "e". These nodes are not optimal due to errors between signals are too big. To reduce the magnitude of this error we take the initial nodes f_i and we get a smooth and continuous function $g(t)$ by means of SLALOM method [5].

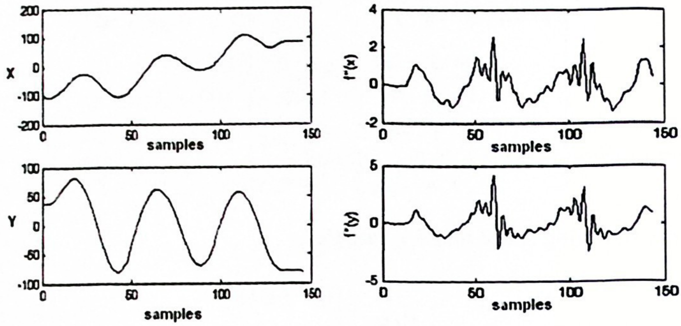

Fig. 7. (a) Signals corresponding to letter "m" after preprocessing and (b) low pass filtered of the second derivative of the same letter.

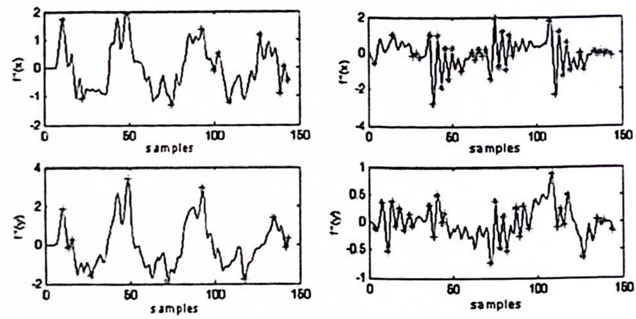

Fig. 8. (a) Initial nodes of letter "m" and (b) initial nodes of letter "e".

Obtaining the optimal nodes. By applying again SLALOM method we obtain the intermediate nodes generated by a smooth and continuous function. Figure 9 shows the nodes obtained by taking the first term of the SLALOM method along with the reconstructed signal. As we can observe from Fig. 9, the obtained points are redundant due to each node represent the same position. By eliminating these redundant nodes and by applying the step and decent method to include the second term of equation (9) we can thus obtain the optimal set of nodes.

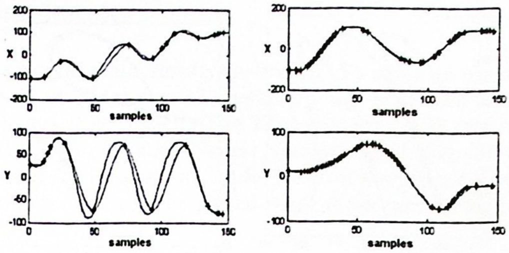

Fig. 9. Original signals (line "-") and reconstructed signals (line "-.") from the obtained nodes ("*") by means of the SLALOM method [5]. (a) letter "m" and (b) letter "e".

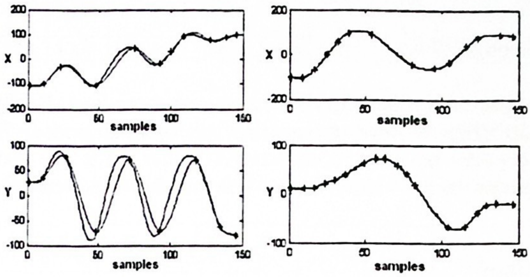

Fig. 10. Original signals (line "-") and reconstructed signals (line "-.") from the obtained nodes ("*"), (a) letter "m" and (b) letter "e".

Optimal node adjustment. The adjustment of the number and positioning of the optimal nodes is performed by using again the step and descent method. Fig. 10 shows the results of the elimination of the redundant nodes, while Fig. 11 shows the results of the adjustment. These operations are applied iteratively while arrives to desired number of nodes (20). Figure 12 shows the optimal nodes (both in the *x* and *y* axes), the original trace and the reconstructed trace from this set of nodes.

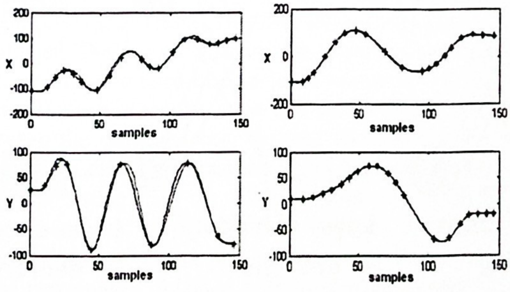

Fig. 11. Adjusted nodes by step and descent method. Original signals (line "-") and reconstructed signals (line "-.") from the obtained nodes ("*"), (a) letter "m" and (b) letter "e".

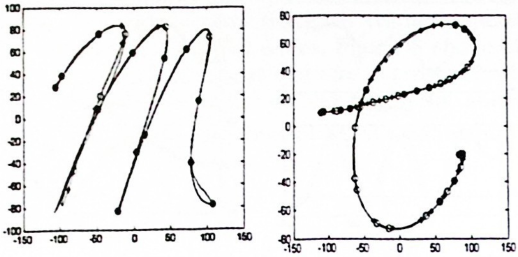

Fig. 12. Original traces and reconstructed traces from the optimal nodes. "*"means optimal node in the x axe, and "o" means optimal node in the y axe.

2.6 Character Recognition

As mentioned before, the proposed system uses one three-layer Backpropagation type neuronal networks (Fig 13) for processing the optimal nodes. For this features processing the networks have 40 input data (20 nodes for x axe and 20 nodes for y axe), with 40 neurons in the hidden layer. This number was determined after several test.

Learning Algorithm

The learning algorithm used for updating the system coefficient matrix is the very well known backpropagation algorithm.

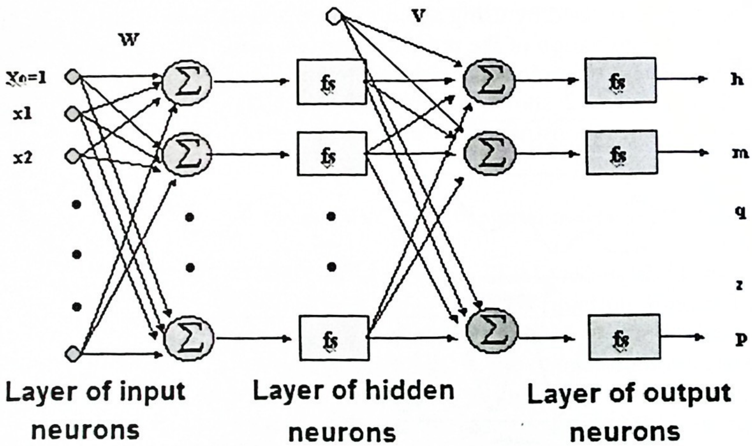

Fig. 13. The structure of the neural network used to recognize the characters.

3 Experimental Results

In this section the experimental results are shown. To evaluate the proposed system, a data base composed of 3,900 cursive characters was generated consist of 50 cursive characters of 26 letters from 3 users. The 2730 characters (35 characters of 26 letters) were used to train Backpropagation Neural Network 1170 characters (15 characters of 26 letters) were used for evaluation. Table 1 shows the proposed system recognition rate using the training data set. The global recognition rate of the proposed system is 99.81%.

Table 1. Recognition rate with characters used during training.

a	b	c	d	e	f	g	h	i	j	k	l	m
100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	99.04%	100%	99.04%
n	o	p	q	r	s	t	u	v	w	x	y	z
100%	99.04%	98.09%	100%	99.04%	100%	100%	100%	100%	100%	100%	100%	100%

Table 2 shows the recognition rate was carried out with characters not used during training. The overall recognition rate of the proposed system is 97.8%.

Table 2. Recognition rate with characters not used during training.

a	b	c	d	e	f	g	h	i	j	k	l	m
95.5%	95.5%	100%	97.7%	100%	100%	100%	93.3%	100%	100%	93.3%	100%	95.5%
n	o	p	q	r	s	t	u	v	w	x	y	z
97.7%	95.5%	95.5%	100%	93.3%	100%	95.5%	100%	100%	100%	97.7%	100%	97.7%

Classification percentages for a similar methodologies described in the literature are from 85% and 98% [1]-[4]. Comparing the proposed system with others similar systems has a similar recognition rate for training and testing data set..

4 Conclusions

In this paper a new methodology for the recognition of cursive manuscript characters has been presented. The SLALOM method was used to obtain the optimal knots of each character. These optimal knots are considered as the describing features of each character, which were used like an input vector in a Backpropagation Neural Network. Computer evaluation results show that the proposed system provides a good recognition rate when the same database is used for training and testing, as well as when both databases are different, obtaining a 99.34 % recognition rate for training data set and 97.43 % testing data set. These results can be considered quite good thinking that the characters recognized are cursive and have a certain shape deformation degree. The recognition percentage using the proposed system is good enough against the percentages obtained with similar proposals systems described in the literature where the results are around 85% to 98% recognition rate.

Acknowledgements. This research was supported by CONACYT and SIP under grants 46805 and 20071438, respectively.

References

1. R. Plamondon, S. N. Srihari, "On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey", IEEE Trans. on Pattern Analysis and Machine Intell., vol. 22, no. 1, 2000, pp. 63-84.
2. S. Manke y U. Bodenhausen, "A Connectionist Recognizer for On-line Cursive Handwriting Recognition", Proceedings of the Int. Conf. Acoustic, Speech and Signal Processing, ICASSP '94, vol. 2, 1994, pp.633-636.
3. R. Plamondon y F. J. Maarse, "An Evaluation of Motor Models of Handwriting", IEEE Trans. System, Man, and Cybernetics, vol.19, no. 5, 1989, pp. 1060-1072.
4. B. Sin, J. Ha, S. Oh y J. Kim "Network-Based Approach to Online Cursive Script Recognition", IEEE Trans. System, Man, and Cybernetics, vol. 29, no. 2, 1999, pp. 321-328.
5. Y. Isomichi, "Inverse-Quantization Method for Digital Signals and Images", IEICE Transactions, Vol. J64-A, no. 4, pp. 285-292, 1981.
6. T. Huang y M. Yasuhara, "A Total Stroke SLALOM Method for Searching for the Optimal Drawing Order of Off-line Handwriting", Proc. IEEE Int. Conf. on System, Man and Cybernetics, vol. 3, 1995, pp. 2789-2794.
7. Hailong Liu; Xiaoqing Ding, "Handwritten character recognition using gradient feature and quadratic classifier with multiple discrimination schemes", Proceedings Eighth International Conference on Document Analysis and Recognition, Vol. 1 pp 19 – 23, 2005.
8. Teng Long; Lian-Wen Jin; Li-Xin Zhen; Jian-Cheng Huang "One stroke cursive character recognition using combination of directional and positional features", IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 5, pp. 449-452, 2005.