A new fuzzy geometric representation for on-line isolated character recognition

Jean-François Hébert, Marc Parizeau
Computer Vision and Systems Laboratory
Dept. of Electrical and Computer Engineering
Laval Univ., Ste-Foy (Qc), Canada, G1K 7P4
{jfhebert,parizeau}@gel.ulaval.ca

Nadia Ghazzali
Dept. of Mathematics and Statistics
Laval Univ., Ste-Foy (Qc), Canada, G1K 7P4
ghazzali@mat.ulaval.ca

Abstract

This paper introduces a new fuzzy representation for isolated character description. This representation maps a character from its original sequence of 2D coordinates into a fuzzy vector space that can thereafter serve as input to any artificial neural network classifier. Recognition experiments on isolated digits extracted from the UNIPEN database are then conducted to evaluate the performances of the proposed representation using an hybrid Kohonen-Perceptron (KP) neural network.

1 Introduction

Artificial neural networks are known to be good classifiers [6]. They are thus used more and more frequently for handwriting recognition problems [1, 3, 9] although, as classifiers, they are usually restricted to an input vector space of fixed dimension. Moreover, this important constraint is usually not respected in commonly used representations for on-line cursive characters [10], that are most often based on a sequence of points of variable length.

This paper describes a new representation for on-line handwritten characters based on fuzzy vectors to represent a fixed number of regions that can lead to a fixed dimension input vector space for any type of classifier that requires such a fixed dimension input vector space. In contrast to other approaches [1, 3], the resulting fuzzy vector space is of lower dimensionality.

The organization of the paper is as follows. Section 2 first describes in detail the proposed fuzzy representation. Then, Section 3 presents briefly the KP neural network used in the following section to evaluate and compare the performance of the proposed representation. Finally, recognition experiments are conducted in Section 4, on handwritten digits found in the international UNIPEN database [4].

2 Proposed representation

The process starts by segmenting the on-line handwriting into a sequence of elementary "strokes" corresponding to circular arcs. The algorithm used was described recently by Li & al. [8]. Figure 1 gives some examples of isolated digits with their stroke reconstruction.

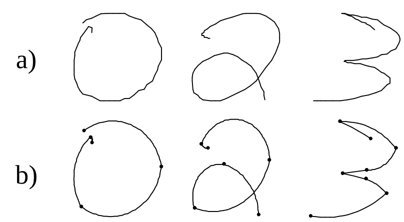


Figure 1. *a) 3 digits; b) their stroke reconstruction.*

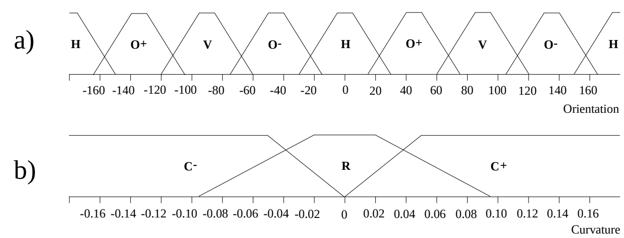


Figure 2. *a) Sets* $\mathbf{H}$,$\mathbf{V}$,$\mathbf{O+}$ *and* $\mathbf{O-}$ *for parameter* θ*; b)* $\mathbf{R}$, $\mathbf{C+}$ *and* $\mathbf{C-}$ *for parameter* c.

More formally, a character $\mathcal{C}$ is thus a sequence of strokes $\mathcal{C} = s_1, s_2, \ldots, s_q$, where any stroke $s = (p_0, p_1, l, c)$ is a circular arc described by four parameters: p_0 and p_1 are respectively the starting and ending points of the arc, l is its curvilinear length, and c its curvature. Another useful stroke parameter is the orientation angle θ of vector (p_0, p_1).

2.1 Fuzzy vector space

The next step of the process is to decomposed the character into a fixed number of (possibly overlapping) regions. For each of these regions, a fuzzy vector is then extracted through fuzzyfication of all strokes found in that region. The complete fuzzy vector space is finally constructed by simple concatenation of regional fuzzy vectors.

The strokes are fuzzyfied using the fuzzy sets shown in Figure 2. The four fuzzy sets defined in Figure 2a) are used to fuzzify the orientation angle $\theta \in [-180, 180]$, whereas the three fuzzy sets in Figure 2b) are used for the fuzzification of

S1: *Fit character $\mathcal{C}$ into a $n \times m$ rectangular grid G.*

S2: *For each $g_{i,j} \in G$, $1 \le i \le n$; $1 \le j \le m$, determine the set of crossing strokes $\mathbf{S}_{i,j} = \{s | s \in g_{i,j}\}^a$. For all strokes $s \in \mathbf{S}_{i,j}$, form the fuzzy vector $\widetilde{s}$:*
$<\mu_{\mathbf{H}}(\theta), \mu_{\mathbf{V}}(\theta), \mu_{\mathbf{O}+}(\theta), \mu_{\mathbf{O}-}(\theta), \mu_{\mathbf{R}}(c), \mu_{\mathbf{C}+}(c), \mu_{\mathbf{C}-}(c)>$

S3: *For each region $g_{i,j}$, determine the fuzzy vector $\widetilde{g}_{i,j}$:*
$\widetilde{g}_{i,j} = <\nu^{i,j}_{\mathbf{H}}, \nu^{i,j}_{\mathbf{V}}, \nu^{i,j}_{\mathbf{O}+}, \nu^{i,j}_{\mathbf{O}-}, \nu^{i,j}_{\mathbf{R}}, \nu^{i,j}_{\mathbf{C}+}, \nu^{i,j}_{\mathbf{C}-}>$
where $\nu^{i,j}_{\mathbf{F}}$ is defined for fuzzy set $\mathbf{F}$ as:

$$\nu^{i,j}_{\mathbf{F}} = \begin{cases} \max\limits_{s \in S_{i,j}} \mu_{\mathbf{F}}(\theta_s) & \text{if } \mathbf{F} \in \{\mathbf{H}, \mathbf{V}, \mathbf{O}+, \mathbf{O}-\} \\ \max\limits_{s \in S_{i,j}} \mu_{\mathbf{F}}(c_s) & \text{if } \mathbf{F} \in \{\mathbf{R}, \mathbf{C}+, \mathbf{C}-\} \end{cases}$$

S4: *Construct fuzzy representation $\widetilde{\mathcal{C}}$ by concatenation of all $\widetilde{g}_{i,j}$.*

[a] A stroke s belongs to region $g_{i,j}$ if s crosses $g_{i,j}$, i.e. if s is completely included in $g_{i,j}$ or if at least one edge of $g_{i,j}$ is crossed by s.

Figure 3. *Basic fuzzy representation algorithm.*

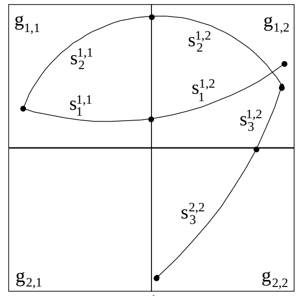
curvature $c \in [-\infty, \infty]$. Therefore, in regard to a specific region, a stroke can be considered more or less horizontal ($\mathbf{H}$), vertical ($\mathbf{V}$), oblique with a positive slope ($\mathbf{O}+$), or oblique with a negative slope ($\mathbf{O}-$) depending on the stroke orientation θ. Similarely, a stroke can be more or less rectilinear ($\mathbf{R}$), have a positive curvature ($\mathbf{C}+$) or a negative curvature ($\mathbf{C}-$). The overlapping between adjacent fuzzy sets in Figure 2 implies that a stroke may belong to two distinct fuzzy sets. For example, a stroke such that $15 \le \theta \le 30$ is considered both horizontal and oblique with a positive slope. And since only one fuzzy vector is associated to each region of a character, we deal with the occurrence of several strokes in a same region by considering the maximum grade of membership in each fuzzy set. This is useful to reveal the occurrence of several strokes having different orientation and curvature values in a region. Figure 3 gives a step by step description of the basic fuzzification process.

2.2 Regional sub-stroke decomposition

One problem with the previous basic representation is that stroke orientation is global and thus does not give precise information about the local orientation of strokes within each region. A first refinement to the representation is called sub-stroke decomposition because each stroke s_k is decomposed in as many sub-stroke $s^{i,j}_k$ as there are fuzzy regions crossed by s_k. Figure 4 illustrates this idea with the sub-stroke decomposition of a '9' digit.

2.3 Regional length normalization

The next refinement of the representation is to take into account the length of strokes. Instead of fuzzyfying the length of strokes (θ and c are not related to scale) and thus add several dimensions to the representation (i.e. each additionnal fuzzy set adds $n \times m$ dimensions), we propose to apply a weight factor w_s on each fuzzified stroke $\widetilde{s}$ such that $\widetilde{s}' = w_s \widetilde{s}$. In practice, the ratio $w_s = l_s / l_d$ where l_s is the arc length of s and l_d is the diagonal length of the region works well.



a)

$\nu_{\mathbf{H}}$:	0.99	$\nu_{\mathbf{H}}$:	0.28
$\nu_{\mathbf{V}}$:	0	$\nu_{\mathbf{V}}$:	0.37
$\nu_{\mathbf{O}+}$:	0.81	$\nu_{\mathbf{O}+}$:	0.32
$\nu_{\mathbf{O}-}$:	0	$\nu_{\mathbf{O}-}$:	0.51
$\nu_{\mathbf{R}}$:	1	$\nu_{\mathbf{R}}$:	1
$\nu_{\mathbf{C}+}$:	0	$\nu_{\mathbf{C}+}$:	0
$\nu_{\mathbf{C}-}$:	0.40	$\nu_{\mathbf{C}-}$:	0.40
$\nu_{\mathbf{H}}$:	0	$\nu_{\mathbf{H}}$:	0
$\nu_{\mathbf{V}}$:	0	$\nu_{\mathbf{V}}$:	0
$\nu_{\mathbf{O}+}$:	0	$\nu_{\mathbf{O}+}$:	0.92
$\nu_{\mathbf{O}-}$:	0	$\nu_{\mathbf{O}-}$:	0
$\nu_{\mathbf{R}}$:	0	$\nu_{\mathbf{R}}$:	1
$\nu_{\mathbf{C}+}$:	0	$\nu_{\mathbf{C}+}$:	0
$\nu_{\mathbf{C}-}$:	0	$\nu_{\mathbf{C}-}$:	0.14

b)

Figure 4. *a) Regional sub-stroke decomposition; b) resulting fuzzy representation.*

2.4 Global characteristics

The last refinement is to introduce global characteristics into the representation in order to better discriminate similar symbols. Specifically, two types of global characteristics are included into the final representation: relative horizontal and vertical densities, and height over width character ratio.

3 KP neural network

In order to evaluate the performance of the proposed fuzzy representations, recognition experiments on isolated digits were conducted using a hybrid Kohonen-Perceptron (KP) network [2]. The KP network combines a Kohonen self-organizing feature map [7] (the K-net) with a multilayer feedforward Perceptron network [5] (the P-net).

The purpose of this network collaboration is to benefit from the well-known modeling capacities of the K-net to help determine which cluster (group of neurons) of the P-net must contribute more than others when producing the KP-net outputs. Thus, when operating in recognition mode, an input vector is first presented to the K-net which calculates for each of its neurons a contribution value based on the Euclidean distance between the input exemplar and the neuron weight vectors. All these neuronal contributions are then passed on to the P-net which uses them to balance the activation signals of the output neurons.

In learning mode, the K-net and the P-net can be trained separately, starting with the K-net. Training algorithms are Kohonen's algorithm for the K-net and a modified back-propagation algorithm for the P-net. Note that the K-net must be trained first since its neuronal contributions are used by the P-net for propagating the input vector and then for back-propagating the output errors. The back-propagation algorithm is modified to take into account the neuronal contributions while adapting neuron weight vectors of the P-net.

4 Experimental results

Experimental results on UNIPEN [4] isolated digits (0-9) are now presented. Section 1a of UNIPEN Train-R01/V07 (15953 digits) made up the training data set, while Section 1a of UNIPEN DevTest-R02/V02 (8598 digits) formed the

Table 1. *Results for the testing data set. Recognition rates are given in percentage according to best of first (*1-h*), second (*2-h*) and third (*3-h*) hypotheses. Dimensionality of representation (*Dim*), number of training epochs for P-net (*Epoch*), and resulting root mean square error (*RMS*) are also given.*

Repr.	Grid	Dim	Epoch	RMS	1-h	2-h	3-h
Bin	8×6	48	400	0.07	89.5	94.7	96.2
	10×8	80	400	0.07	89.1	94.2	95.9
	14×10	140	300	0.06	88.8	93.5	95.5
Basic	3×2	42	300	0.07	90.6	95.7	97.2
R2	3×2	42	400	0.06	91.6	96.0	97.3
R3	3×2	42	300	0.05	94.6	97.3	98.1
Final	3×2	49	600	0.04	95.3	97.9	98.6
	4×3	93	300	0.04	96.1	98.2	99.1
	5×4	151	300	0.04	96.3	98.4	99.0

testing data set. The reader should note that, except for some empty segments, none of the digits found in these databases were removed, even though many of them are either mislabeled or very badly written.

Recognition rates on the test data set using various representations are given in Table 1. The *Bin* representation corresponds to a binary pixel matrix indicating whether the character drawing crosses or not the pixel. This representation is used as a performance reference. The *Basic* representation is the initial proposition described in section 2.1, whereas representations *R2* and *R3* integrate respectively the refinements of Sections 2.2 (sub-stroke decomposition) and 2.3 (length normalization). The *Final* representation integrates all refinements. For all recognition experiments, the same KP network structure was used (i.e. a 15×15 Kohonen map linked with a $k \times 225 \times 10$ multilayer Perceptron; k being the dimensionality of the representation). In all recognition experiments, the K-net was trained using the same parameters. The parameters of the P-net were also fixed except for the number of training epochs in order to avoid over-training. For testing, an input exemplar is said to be recognized according to the best hypothesis (*1-h*) if the maximum argument of its observed outputs matches with the maximum argument of its desired outputs. If the match rather coincides with the second or third maximum value, then the digit is said to be recognized according to second or third best hypotheses (*2-h* or *3-h*).

Results given in Table 1 clearly indicate that the combination of various levels of analysis improves the fuzzy representation. Indeed, for the best hypothesis, the recognition rate obtained with the *Final* 3×2 representation is 4.5% over the recognition rate obtained with the *Basic* representation, and almost 6% over the rate obtained with the 8×6 *Binary* representation. Furthermore, if one considers the top three hypotheses, results as high as 99.1% can be obtained.

Finally, in order to evaluate how much these results could still be improved upon, we conducted a reading experiment with 10 human subjects. They were presented in random order the digits that were not correctly recognized (*1-h*) by our network, and were asked to classify them knowing that they were digits. On average, these humans commited 27.3% of error which corresponds to around 2.7% of possible improvements over our best result. But of course, there is no guarantee what so ever that our human subjects could recognize perfectly all the digits that our network recognized, thus these 2.7% of improvements are only hypothetical!

5 Conclusion

This paper has introduced an algorithm for extracting a new fuzzy representation for isolated characters recognition. This representation has been shown to give excellent results (up to 96.1% for first choice; and up to 99.1% for top-three choices) on handwritten digits extracted from the international UNIPEN database of on-line handwriting. These results were obtained using an hybrid Kohonen-Perceptron neural network as classifier.

Acknowledgments: This research was supported in part by NSERC grants to M. Parizeau and N. Ghazzali, and in part by an FCAR grant to M. Parizeau.

References

[1] Y. Bengio, Y. LeCun, C. Nohl, and C. Burges. LeRec: A NN/HMM Hybrid for On-Line Handwriting Recognition. *Neural Computation*, 7(5), 1995.

[2] M. Guillot and R. Azouzi. Improving On-Line Adaptation in Neurocontrol using a Combination of Self-organizing Map and Multilayer Feedforward Network. *Appears in Intelligent Engineering Systems Through Artificial Neural Networks, edited by C.H. Dagli*, 4:915–922, 1995.

[3] I. Guyon, P. Albrecht, Y. LeCun, J. Denker, and W. Hubbard. Design of a Neural Network Character Recognizer for a Touch Terminal. *Pattern Recognition*, 24(2):105–119, 1991.

[4] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet. UNIPEN Project of On-line Data Exchange and Recognizer Benchmarks. *ICPR*, 2:29–33, 1994.

[5] S. Haykin. *Neural Networks – A Comprehensive Foundation*. IEEE Press, 1995.

[6] L. Holmström, P. Koistinen, J. Laaksonen, and E. Oja. Neural and Statistical Classifiers - Taxonomy and Two Case Studies. *IEEE Trans. on Neural Networks*, 8(1):5–17, 1997.

[7] T. Kohonen. The Self-Organizing Map. *Proc. of the IEEE*, 78(9):1464–1480, 1990.

[8] X. Li, M. Parizeau, and R. Plamondon. Segmentation and Reconstruction of On-Line Handwritten Scripts. *Pattern recognition*, 31(6), 1998.

[9] L. Schomaker. Using stroke- or character-based self-organizing maps in the recognition of on-line, connected cursive script. *Pattern Recognition*, 26(3):443–450, 1993.

[10] C. Tappert, C. Suen, and T. Wakahara. The State of the Art in On-Line Handwriting Recognition. *IEEE Trans. on Pattern and Machine Intelligence*, 12(8):787–808, 1990.