ON-LINE CHARACTER RECOGNITION USING HISTOGRAMS OF FEATURES AND AN ASSOCIATIVE MEMORY

N. Mezghani, A. Mitiche

INRS-EMT
800, de la Gauchetière Ouest, Suite 6900
Montréal QC, H5A 1K6 Canada.

M. Cheriet

École de Technologie Supérieure
1100 rue Notre-Dame Ouest
Montréal QC, H3C 1K3 Canada

ABSTRACT

The purpose of this study is to investigate a new representation of shape and its use in handwritten on-line character recognition. This representation is based on the empirical distribution of features such as tangents, and tangent differences at distant points along the character signal. Recognition is carried out by an associative memory trained using this representation and the Hellinger distance which measures distance between distributions. We report on extensive experiments that show the pertinence of the representation and the superior performance of the scheme.

1. INTRODUCTION

Due to the increasing popularity of hand-held computers, digital notebooks, and advanced cellular phones, automatic on-line recognition of handwritten text has been gaining more interest lately. The traditional methods of human-machine communication, such as keyboards and pointing devises, are often inconvenient to use when the size of the machine is so small as to fit a palm. In this context, handwriting recognition is a very attractive input method.

The most challenging issues in handwriting recognition is the vast variation in personal writing styles. A recognition system should be to some extent insensitive to such variations and still be able to distinguish different but sometimes similar looking characters. Current systems usually have problems handling such variations.

Several methods have been developed for on-line character recognition system [1]. Research has concentrated on classifiers rather than representation, although a good representation is as important as a good classifier.

Various classifiers have been investigated, the most successful being neural networks (TDNN [2], MLP [3]) for reasons such as simplicity of conception, speed of execution, and performance. Hidden Markov models and hybrid HMM/NN have also been used [4, 5, 6, 7]. Finally, nearest neighbor classifiers using elastic matching [8, 9] and weighted elastic matching [10] have been considered in some character recognition applications.

As representation, stroke structures have been the most extensively used [11, 2]. Also topological and geometrical features such as Fourier descriptors have been proposed in [12, 13]. In other studies there were efforts to model pen-tip movement [14, 15] to extract features such as curvilinear and angular velocities.

In this study we propose a new representation of character shape and investigate its use in on-line Arabic character recognition by a Kohonen associative memory. A Kohonen memory is a hight performance classifier which requires light training efforts and has attractive properties such as good generalization. As representation we investigate statics based on features empirical distributions (histograms). The features considered are tangent and tangent differences at regularly spaced points along the character signal.

This paper is organized as follows: section 2 discusses the proposed representation. Section 3 explains the associative Kohonen memory. In section 4 we presents the database and in section 5 we discusses some experimental results.

2. FEATURE REPRESENTATION

Feature extraction and selection are important in achieving high recognition. We investigate a representation based on feature empirical distributions (histograms). The features considered are tangents, and tangent differences at regularly spaced points along the character signal.

Let $\Gamma(s)$ be the arc length parametric representation of the curve of a character, and N equidistant points on $\Gamma(s)$ $((x_0, y_0)(x_1, y_1)...(x_k, y_k).....(x_{N-1}, y_{N-1}))$

Let $\phi^{(0)}$ be the feature the measurements of which are the tangent angles defined by:

$$\theta_k = \arctan(\frac{y_{k+1} - y_k}{x_{k+1} - x_k}) \quad k \in \{0, 1, ..., N\} \qquad (1)$$

For $\alpha \in \mathbb{N}$, $1 \leq \alpha \leq N-1$, let $\phi^{(\alpha)}$ be the feature the measurements of which are the tangent angle differences defined by:

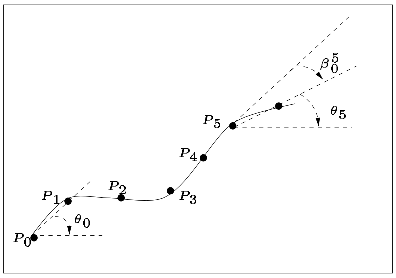

Fig. 1. Tangent angle and tangent angle difference.

$$\beta_k^{(\alpha)} = \theta_{(k+\alpha) mod N} - \theta_k \quad k \in \{0, 1, ..., N\} \tag{2}$$

Therefore, for each shape Γ, we have a set of features $\Phi = \{\phi^{(\alpha)}, \alpha = 0, 1, 2, ..., N-1\}$. The tangent angle ($\phi^{(0)}$) is invariant under translation and scaling and the tangent angle difference is invariant to rotation as well.

Now, we compute statistics for each feature $\phi^{(\alpha)}$, namely the empirical distribution (histograms) of features. In the continuous case the histogram of $\phi^{(\alpha)}$ on shape $\Gamma(s)$ is defined as:

$$H^{(\alpha)}(\Gamma, z) = \int \delta(z - \phi^{(\alpha)}(s)) ds \quad \alpha = 0, 1, ..., N-1 \tag{3}$$

In the above definition, z is a continuous variable for the feature, e.g $H^{(0)}(\Gamma, 0)$ is the number of points on Γ that have zero angle. δ is the Dirac delta function with unit mass at zero and $\delta(x) = 0$ for $x \neq 0$. For discretized variables we have:

$$H^{(\alpha)}(\Gamma, z) = \frac{1}{N} \sum_{j=1}^{N} \delta(z - \phi^{(\alpha)}(s_j)) \tag{4}$$

In practice, we descretized a histogram $H^{(\alpha)}(\Gamma, z)$ into m bins as shown in figure 2. Therefore, for each feature $\phi^{(\alpha)}$ the histogram is an m-dimensional vector:

$$H^{(\alpha)}(\Gamma) = (H_1^{(\alpha)}, H_2^{(\alpha)}, ...H_m^{(\alpha)})$$

Finally, we compose a vector of representation using histograms of all the features:

$$\mathcal{H} = (H^{(0)}, H^{(1)}, ...H^{(N-1)}) \tag{5}$$

3. THE KOHONEN MEMORY

The Kohonen memory (also called Kohonen self organizing feature map) used to represent all points in a source into a lesser number of points in a target space, in such a way that neighboring memory points have neighboring values to preserve topological ordering.

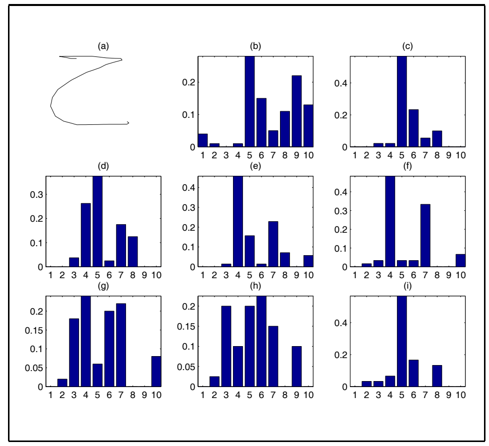

Fig. 2. (a) The "Ha" letter (b) Histogram of tangent angle $H^{(0)}$ (c) Histogram of tangent angle differences $H^{(10)}$ (d) Histogram of tangent angle differences $H^{(20)}$ (e) Histogram of tangent angle differences $H^{(30)}$ (f) Histogram of tangent angle differences $H^{(40)}$ (g) Histogram of tangent angle differences $H^{(50)}$ (h) Histogram of tangent angle differences $H^{(60)}$ (i) Histogram of tangent angle differences $H^{(70)}$

The memory is organized as a two-dimensional array as shown in figure 3. $X = (x_1, x_2, ...x_I)$ is the input. Memory node j stores weight vector $W_j = (w_{1j}, w_{2j}, .., w_{Ij})$ determined during training.

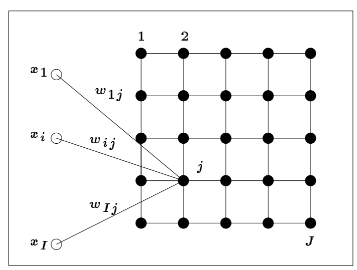

Fig. 3. Kohonen memory of J nodes.

The Euclidean distance has been traditionally used in the Kohonen memory algorithm. However, we use the Hellinger distance. The Hellinger distance is a more appropriate measure because it measures distance between distributions.

1. *Initialize weights* W_j^0 to small random values, $j \in [1, J]$.
2. *Get new input* $X^n = (x_1^n, ..., x_I^n)^T$, and compute the Hellinger distances to all weight vectors $W_j^n = (w_{1j}^n, ..., w_{Ij}^n)^T$ according:

$$d(X, W_j) = \sum_{i=0}^{I} (\sqrt{x_i^n} - \sqrt{w_{ij}^n})^2 \tag{6}$$

3. *Find node* j^* with smallest distance.

4. *Update weights*:

$$w_{ij}^{n+1} = w_{ij}^{n} + \epsilon_n h_n^{j,j^*}(x_i^n - w_{ij}^n) \tag{7}$$

$$\epsilon_n = \epsilon_i(\frac{\epsilon_f}{\epsilon_i})^{\frac{n}{n_{max}}}, \quad \sigma_n = \sigma_i(\frac{\sigma_f}{\sigma_i})^{\frac{n}{n_{max}}} \tag{8}$$

$$h_n^{j,j^*} = exp - \frac{||j - j^*||^2}{2\sigma_n^2} \tag{9}$$

In training, a set of vectors $X = (x_1, x_2, .., x_I)^T$ is input repeatedly to the J nodes of memory containing each a weight vector $W_j = (w_{1j}, w_{2j}, ..., w_{Ij})^t$, initially consisting of random values. Nodes respond to the input vector according to the distance between the input vector and the node's weight vector. The node with weight closest to the input is determined and weights at all nodes are updated. Function h^{j,j^*} defines the influence of node j^* on node j during update at j. It depends on parameter σ which decreases with iterations between the value σ_i and σ_f. ϵ scales weight change and varies with iterations from ϵ_i to ϵ_f. Parameters σ_i, σ_f, ϵ_i and ϵ_f must be chosen appropriately to obtain convergence and topological ordering. The result of training is a set of weight vectors $W_j = (w_{j1}, w_{j2}, .., w_{jJ})$, stored at nodes $j = 1, ..., J$ to represent the Kohonen map contents. Once the memory is trained, observed vector X is classified by presenting it and assigning to it the class of the node whose content is closest.

4. THE DATABASE

Data collection was done using a digital Wacom Graphire tablet, with a resolution accuracy of 23 points/cm. It has a sampling frequency of 100 points/sec. Arabic has 28 letters in the alphabet based on 18 distinct shapes that vary according to their connection to preceding or following letters (Fig.4). Using a combination of dots and symbols above and below these shapes, the full complement of 28 consonants can be constructed. Our system recognizes 17 distinct classes of shapes because the "Fa" letter and "Qaf" letter are the same expect for their position with respect to the base line. The database contains 432 samples of each character, written by 18 writers without constraint, leading to a wide variety of size and orientation.

The on-line signal is smoothed and resampled to have equidistant points . Smoothing consists of averaging a points with its neighbors, we used the 3-points average. Resampling is a processing step implemented in almost every on-line handwritten system. In general, points recorded during writing are equidistant in time but not in space. Hence, the number of captured points varies depending on the velocity of writing. To normalize the number of points, the sequence

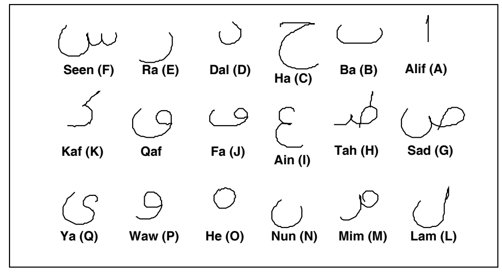

Fig. 4. The 18 shapes of Arabic isolated characters and their assigned labels.

of captured points is replaced with a sequence of points having the same spatial distance. Therefore, all characters will have the same number of points.

5. EXPERIMENTAL RESULTS

The database is divided into two distinct sets. The training set contained 4896 samples and the testing set 2448 samples (which corresponds to 288 samples of each character for training and 144 samples of each character for testing).

Using all the features $\phi^{(\alpha)}$, $\alpha \in \{0, 1, ..., N-1\}$, results in a vector of significantly high dimension (e.g for $N = 100$ and $m = 10$, the dimension is 1000). Therefore, we use a subset of these features that we have choose experimentally. We tested these using a memory of 400 nodes (this is a sufficient size, also determined experimently) and a number of iterations of 80. Table 1 shows the recognition rates for α's multiples of 10. The tendency is for the rate, as a function of α, to grow to maximum and then decrease.

Index feature α	0	10	20	30	40
Recognition rate τ	68.99	66.54	70.38	75.32	77.04
Index feature α	50	60	70	80	90
Recognition rate τ	74.95	69.32	65.77	50.16	-

Table 1. Recognition rates vs. index features (rates obtained for each histogram taken individually)

Following these first experiments, we retained the features for $\alpha = 0, 10, 20, 30, 40$, to compose the vector of representation which, in this case, is of dimension 50. With this vector of representation we obtained a superior recognition rate of 94.56%. The size of the memory and the number of iteration have also been selected experimentally. The retained number of nodes is 400 and the number of iterations 80 (Fig. 5).

As one can notice, the recognition rate obtained by the resulting vector of representation is far superior than any histogram taken individually.

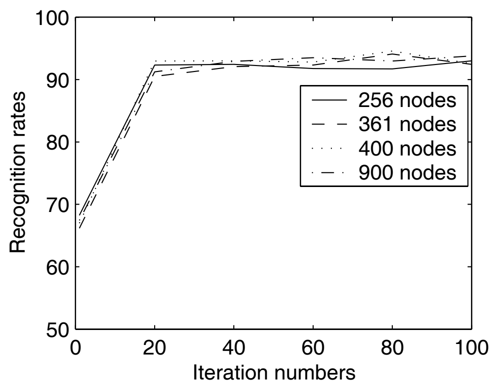

Fig. 5. Recognition rate vs. the number of iterations

We also measured the various execution times. Table 2 shows that the recognition time is relatively short. This is very important for an on-line recognition system. The training time is long, but is not a disadvantage because training is done only one time.

steps	Training	Labeling	Recognition
Execution time per character	1h 5mn	8.47s	1.05s

Table 2. Execution time vs. steps in Kohonen map

6. CONCLUSION

The aim of this paper was to develop a new representation of shape and to use it in handwritten on-line Arabic character recognition. We investigated statistics of feature based on histograms of tangent angles and tangent angle differences. Recognition was carried out by an associative memory. Experimental results show the high performance of the proposed scheme and the pertinence of the representation.

7. REFERENCES

[1] C.C. Tappert, C.Y. Suen, and T. Wakahara, "The state of the art in on-line handwriting recognition," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 8, pp. 787–808, August 1990.

[2] G. Seni, *Large Vocabulary Recognition of On-line Handwritten Cursie Words*, Ph.D. thesis, the state university of New York at Buffalo, August 1995.

[3] T. Klassen, "Towards neural network recognition of handwritten arabic letters," M.S. thesis, Dalhousie University, Halifax, 2001.

[4] J. Hu, A.S. Rosenthal, and M.K. Brown, "Combining high-level features with sequetial local features for on-line recognition," *Proceedings of Italian Image Processing confrence, Florence*, pp. 647–654, September 1997.

[5] J. Hu, M.K. Brown, and W. Turin, "Hmm based on-line handwriting recognition," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, pp. 1039–1045, October 1996.

[6] K.S. Nathan, H.S.M Beigi, H.Subrahmonia, G.J Clary, and H. Maruyama, "Real-time on-line unconstrained handwritting using statistical methods," *Proceedings of ICAASSP'95*, vol. 4, pp. 2619–2623, May 1995.

[7] M. Schenkel, I. Guyon, and D. Henderson, "On-line cursive script recognition using time delay neural networks and hidden markov models," *Machine Vision and Applications*, pp. 215–223, 1995.

[8] X. Li and D.-Y. Yeung, "On-line handwritten alphanumeric character recognition using dominant point strokes," *Pattern recognition*, vol. 30, no. 1, pp. 675–684, June 1998.

[9] H. Sakoe S. Uchida, "Handwritten character recognition using elastic matching based on a class-dependent deformation model," *International Conference on Document Analysis and Recognition*, vol. 1, pp. 163–168, August 2003.

[10] P. Scattolin and A. Krzyzak, "Weighted elastic matching method for recognition of handwritten numerals," *Proc. Conf. Vision Interface*, pp. 178–185, 1994.

[11] T.S. Al-Sheikh and S.G El-Taweel, "Real-time arabic handwritten character recognition," *Pattern recognition*, vol. 23, no. 12, pp. 1323–1332, 1990.

[12] N. Mezghani, A. Mitiche, and M. cheriet, "Reconnaissance en-ligne de caractères arabes manuscrits par un rèseau de kohonen," *Proc. Vision Interface 2002*, pp. 186–191, 2002.

[13] S. Manke, M. Finke, and A. Waibel, "Npen++: A writer independent, large vocabulary on-line cursive handwriting recognition system," *Proceedings of the ICDAR 95*, 1995.

[14] R. Plamondon, "Handwriting generation: the delta lognormal theory," *Proceedings of the Fourth International Workshop on Frontiers in Handwriting Recognition*, pp. 1–10, 1994.

[15] R. Plamondon, "A model-based segmentation framework for computer processing of handwriting," *Proceedings of the Eleventh IAPR International Conference on Pattern Recognition*, pp. 303–307, 1992.