Complementary Features Combined in an HMM-based System to Recognize Handwritten Digits

Alceu de S. Britto Jr
Pontifícia Universidade Católica do Paraná (PUCPR) - Brazil
Universidade Estadual de Ponta Grossa (UEPG) - Brazil
alceu@ppgia.pucpr.br

Robert Sabourin
École de Technologie Supérieure (ÉTS) - Canada
Robert.Sabourin@etsmtl.ca

Flavio Bortolozzi
Pontifícia Universidade Católica do Paraná (PUCPR) - Brazil
fborto@ppgia.pucpr.br

Ching Y. Suen
Centre for Pattern Recognition and Machine Intelligence (CENPARMI) - Canada
suen@cenparmi.concordia.ca

Abstract

In this paper we combine complementary features based on foreground and background information in an HMM-based classifier to recognize handwritten digits. A zoning scheme based on column and row models provides a way of dividing the digit into zones without making the features size variant. This strategy allows us to avoid the digit normalization, while it provides a way of having information from specific zones of the digit. Recognition rates around 98% have been achieved using 60,000 digit samples of the NIST SD19 database.

1. Introduction

Many approaches to solving the handwritten numeral recognition problem have been proposed in recent years due to its numerous possible applications. Drawing up a taxonomy of these approaches is difficult, since their methodologies overlap. However, research in this field has basically considered investigating: a) feature extraction methods; b) classification methods; and c) system architectures based on different strategies, such as combinations of multiple classifiers, the use of multiple templates, and the use of verification modules.

The investigation of feature extraction methods has gained considerable attention since a discriminative feature set is considered the most important factor in achieving high recognition performance. In [1] a survey of feature extraction methods for off-line recognition of segmented characters is presented. The authors describe important aspects that must be considered before selecting a specific feature extraction method. Another interesting work of shape analysis techniques can be found in [2].

In general, the feature extraction methods for numeral recognition reported in the literature have been based on two types of features, statistical and structural. The statistical features are derived from statistical distributions of points, such as zoning, moments, projection histograms or direction histograms [3-5]. Structural features are based on topological and geometrical properties of the character, like strokes and their directions, end-points, or intersections of segments and loops [6-9].

Many researchers have explored the integration of structural and statistical information to highlight different character properties, since these types of features are considered to be complementary. In [9] structural and statistical information is integrated into a classifier based on Hidden Markov Model (HMM). The authors use state-duration adapted transition probability distribution and macro-states to overcome the weakness of the HMMs in modeling structural features. Both statistical and structural features are extracted from chain code (locations, orientations and curvatures). The recognition rate is 96.16% in 2,711 digit samples extracted from the CEDAR database.

Another multifeature-based system is proposed in [10]. In this work, a combination of seven different families of features is proposed in order to arrive at a complete character description. These features are divided into global features (invariant moments, projections and profiles) and local features (intersections with straight lines, holes and concave arcs, extremities, end-points and junctions). A set of 53,324 digits extracted from the NIST database is used to test the system. The recognition, rejection and substitution rates are 90.82%, 8.93% and 0.25% respectively.

In [11] a MLP-based classifier based on concavity features achieved a recognition rate of over 99.7% in 60,000 samples of handwritten digits of the NIST SD19 database.

In this paper we combine features extracted from the foreground and the background of digit images in an HMM-based system. The challenge is to achieve recognition rates close to that presented in [11] with MLPs, however, using an HMM-based system. HMM have been used with success to model isolated characters, which are applied to recognize words or numeral strings. The reason is that HMM can model specific handwriting knowledge related to the interaction between adjacent characters in words or numeral strings easier than MLPs. Moreover, HMM has been successfully used to provide implicit segmentation-based methods to recognize words and numeral strings as in [12]. With such an approach it is possible to avoid a priori segmentation of the string or word into characters.

This paper is organized as follows. Section 2 describes in details our feature extraction method. The foreground and background features are described. Section 3 describes the HMM-based classifier. Combining column and row-based models provides an implicit zoning scheme. Section 4 presents the experiments undertaken to develop the system and also some experiments to evaluate the recognition performance of our classifier. Finally, in Section 5 we present conclusion and future works.

2. Feature extraction method

The extraction method consists of scanning the digit image from left-to-right (column-based features) and from bottom-to-top (row-based features). Foreground and background information are combined in a vector of 47 features: 34 foreground plus 13 background features.

2.1 Foreground features (FF)

The FF vector consists of local and global features calculated taking into account the foreground pixels of the image columns or rows. The local features are based on transitions from background to foreground pixels and vice versa.

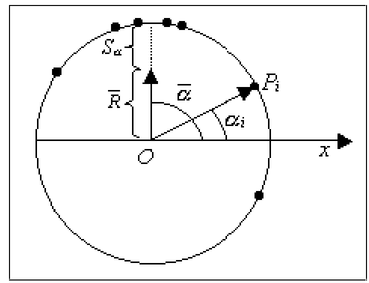

Figure 1. Circular mean direction $\overline{\alpha}$ and variance S_α for a distribution $F(\alpha_i)$

For each transition, the mean direction and corresponding variance are obtained by means of statistic estimators. These estimators are more suitable for directional observations, since they are based on a circular scale. For instance, given the directional observations $\alpha_1 = 1^o$ and $\alpha_2 = 359^o$, they provide a mean direction ($\overline{\alpha}$) of 0^o instead of 180^o calculated by conventional estimators. Let $\alpha_1, \ldots, \alpha_i, \ldots, \alpha_N$ be a set of directional observations with distribution $F(\alpha_i)$ and size N. Figure 1 shows that α_i represents the angle between the unit vector $\overline{OP_i}$ and the horizontal axis, while P_i is the intersection point between $\overline{OP_i}$ and the unit circle. The cartesian coordinates of P_i are defined as:

$$(\cos(\alpha_i), \sin(\alpha_i)) \quad (1)$$

The circular mean direction $\overline{\alpha}$ of the N directional observations on the unit circle corresponds to the direction of the resulting vector $(\overline{R})$ obtained by the sum of the unit vectors $(\overline{OP_1}, \ldots, \overline{OP_i}, \ldots, \overline{OP_N})$. The center of gravity $(\overline{C}, \overline{S})$ of the N coordinates $(cos(\alpha_i), sin(\alpha_i))$ is defined as:

$$\overline{C} = \frac{1}{N}\sum_{i=1}^{N} cos(\alpha_i) \quad (2)$$

$$\overline{S} = \frac{1}{N}\sum_{i=1}^{N} sin(\alpha_i) \quad (3)$$

These coordinates are used to estimate the mean size of $\overline{R}$, as:

$$\overline{R} = \sqrt{\left(\overline{C}^2 + \overline{S}^2\right)} \quad (4)$$

Then, the circular mean direction can be obtained by solving one of the following equations:

$$cos\left(\overline{\alpha}\right) = \frac{\overline{C}}{\overline{R}}, \qquad sin\left(\overline{\alpha}\right) = \frac{\overline{S}}{\overline{R}} \quad (5)$$

Finally, the circular variance of $\overline{\alpha}$ is calculated as:

$$S_\alpha = 1 - \overline{R} \qquad 0 \le S_\alpha \le 1 \quad (6)$$

To estimate $\overline{\alpha}$ and S_α for each transition of a numeral image, we have considered $\{0^0, 45^0, 90^0, 135^0, 180^0, 225^0, 270^0, 315^0\}$ as the set of directional observations, while $F(\alpha_i)$ is computed by counting the number of successive black pixels over the direction α_i from a transition until the encounter of a white pixel. In Figure 2 the transitions in a column of numeral 5 are enumerated from 1 to 6, and the possible directional observations from transitions 3 and 6 are shown.

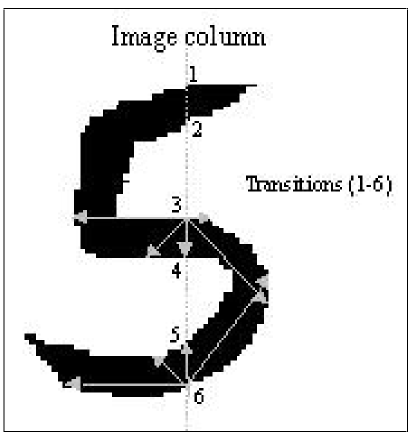

Figure 2. Transitions in a column image of numeral 5, and the directional observations to estimate the mean direction for transitions 3 and 6

In addition to this directional information, we have calculated two other local features: a) relative position of each transition, taking into account the top of the digit bounding box, and b) whether the transition belongs to the outer or inner contour, which shows the presence of loops in the numeral image. Since for each column we consider 8 possible transitions, at this point our feature vector is composed of 32 features.

The global features are based on vertical projection (VP) of black pixels for each column, and the derivative of VP between adjacent columns. This constitutes a total of 34 features normalized between 0 and 1.

2.2 Background features (BF)

The BF vector is based on concavity information (see Figure 3). These features are used to highlight the topological and geometrical properties of the digit classes. Each concavity feature represents the number of white pixels that belong to a specific concavity configuration.

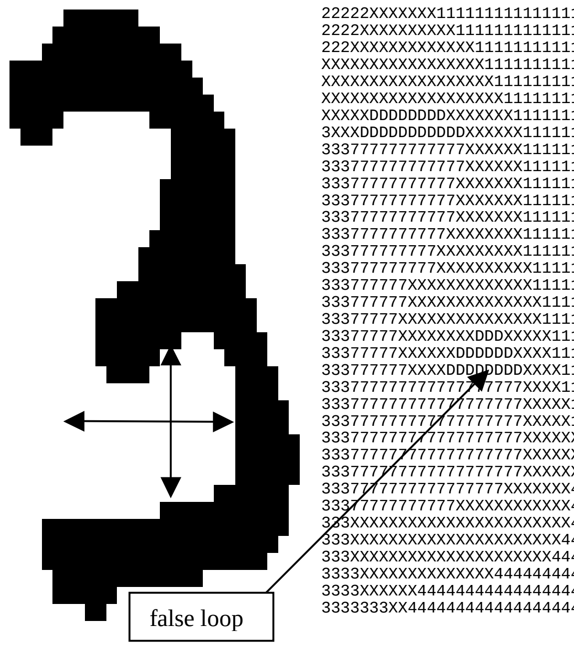

Figure 3: Example of concavity features

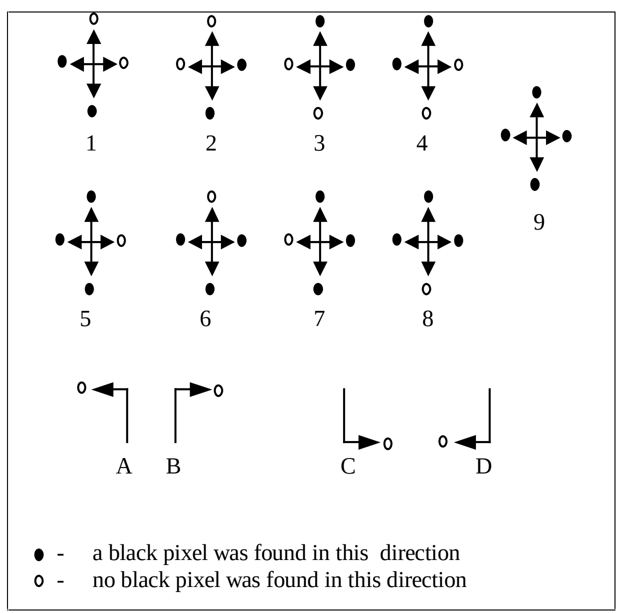

Figure 4 – Concavity and false loop configurations

The label for each white pixel is chosen based on the Freeman code with four directions. Each direction is explored until the encounter of a black pixel or the limits imposed by the digit bounding box. A white pixel is labeled if at least two consecutive directions find black pixels. Thus, we have 9 possible concavity configurations. Moreover, we consider four more configurations, in order to detect more precisely the presence of loops.

The total length of this feature vector is then 13. The concavity vector is normalized between 0 and 1, by the total of the concavity codes computed for each column or row of the digit image. Figure 4 shows the 9 concavity configurations and also 4 configurations for false loops.

2.3 Column and row-based features

The feature vector composed of foreground and background features is extracted from each column and row of the digit image. Each feature vector is mapped to one of 256 possible discrete symbols available in a codebook previously constructed by using the K-means algorithm [16]. Thus, the output of the feature extraction method consists of two sequences of discrete observations for each digit: column-based and row-based sequences.

3. Hidden Markov models

In the proposed classifier each digit class is represented by two numeral HMMs: one based on columns ($\lambda_c^0, \lambda_c^1, \ldots, \lambda_c^9$) and other based on rows ($\lambda_r^0, \lambda_r^1, \ldots, \lambda_r^9$) of the digit image. These column- and row-based models provide a way of combining foreground and background features in the zoning scheme as shown in Figure 5.

The topology of the numeral models is defined taking into account the recognition of handwritten text. This means a left-right model with number of states defined as described by Wang in [13], which defines the possible number of states (N) of the HMMs taking into account durational statistics calculated from the training database. First the mean length μ and the variance σ^2 of all observation sequences in the training set are collected and they together define the possible N for each numeral HMM:

$$\frac{\mu(\mu-1)+\sigma^2}{\mu-1+\sigma^2} < N < \mu+1-\sqrt{2\sigma^2+1} \qquad (7)$$

Table 1 presents the range (minimum and maximum number of states) for each numeral model calculated on the training set (50,000 isolated digits – 5,000 per class). In addition, the mean length value is also calculated. The final number of states of each digit model was experimentally defined as that corresponding to the minimum value in Table 1.

Table 1. Minimum, maximum and mean number of states by digit class

Numeral model	Column based models			Row based models		
	Min	Mean	Max	Min	Mean	Max
0	13	18	24	14	21	28
1	5	6	7	16	24	32
2	14	22	30	16	24	32
3	14	20	26	20	28	36
4	15	22	28	18	28	39
5	13	21	29	19	27	35
6	15	20	25	18	27	36
7	15	20	25	18	27	36
8	14	17	24	20	29	38
9	16	20	25	21	31	41

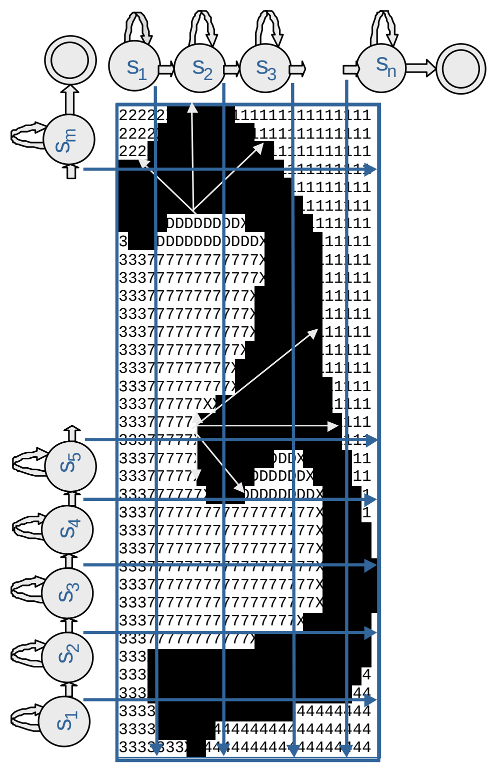

Figure 5. Implicit zoning scheme provided by combining column and row-based models.

4. Experimental results

The experiments undertaken during the course of development of the proposed method were done using isolated numerals from the NIST SD19. We use 50,000 numeral samples for training, 10,000 for validation and 10,000 for testing. A final experiment was done using a more robust protocol based on 195,000 samples for training, 28,000 for validation and 60,000 for testing. In all the experiments a zero-level rejection was used.

4.1 Evaluation of the number of states

The gap between the number of states usually found in the literature for HMMs used to represent characters (5 or 6 states) [14,15] and those estimated using the scheme presented in Section 3 is very large (see Table 1). For this reason, we decide at this time to evaluate, for the column-based models, configurations with 6, 8 and 12 states. Table 3 shows the recognition results considering different number of states for the column and row numeral models.

Table 3 - Experiments considering different number of states in the numeral HMMs

	Column models		Row models	
Number of states	Validation (%)	Testing (%)	Validation (%)	Testing (%)
6	97.63	94.55	95.65	92.27
8	97.78	94.90	-	-
12	97.89	95.26	-	-
Minimum values	98.01	95.51	97.56	95.16
Mean values	97.54	94.61	97.40	95.02

The best results were obtained by using the minimum values presented in Table 1. The maximum values were not evaluated since we have observed a loss in terms of recognition rates for the mean values.

4.2 Evaluation of the codebook sizes

The codebook size was experimentally optimized. We have evaluated codebooks composed of 64, 128, 192, 256 and 320 entries. The codebook composed of 256 entries provided the best results (see Table 4). The recognition rate of the row models considering a codebook with 64 entries were not calculated, since we have observed that 256 entries provided better results (based on the column-based models).

Table 4 – Experiments using different codebook sizes

	Column models		Row models	
Codebook size	Validation (%)	Testing (%)	Validation (%)	Testing (%)
64	95.40	92.94	-	-
128	97.89	95.26	97.56	95.16
192	98.24	96.23	98.16	96.63
256	98.44	96.54	98.40	97.09
320	98.32	96.44	98.30	96.92

4.3 Combination of column and row models

The experiments have shown that combining column- and row-based models to represent each digit class provides an interesting recognition performance. In Table 5, experiment (a), only the column-based model using the foreground feature (FF) vector was evaluated. In the experiment (b), we observed a significant improvement in the recognition performance when we combine foreground and background features in the column-based model. Similar results were obtained in the experiment (c) for the row-based models. Finally, the experiment (d) has shown that column and row-based models are really complementary. The models were combined by summing the *log* of their final probability calculated using Viterbi's algorithm.

Table 5. Combination of column and row models

		Valid.(%)	Testing (%)
(a)	Column (FF vector)	96,79	94,00
(b)	Column (FF + BF vectors)	98.44	96.54
(c)	Row (FF + BF vectors)	98.40	97.09
(d)	Combination (column and row models used in (b) and (c), respectively)	99.00	98.02

Table 6 shows the confusion matrix related to the experiment (a) presented in Table 5. We can observe many confusions between digit classes: 0-6, 2-7, 3-5, 4-9, 6-0, 8-6, 9-0 and 9-4.

Table 6. Confusion matriz: column model (FF vector)

	0	1	2	3	4	5	6	7	8	9
0	929	0	1	0	9	0	20	0	5	8
1	0	980	1	1	0	0	0	2	0	1
2	9	9	969	4	3	1	1	74	5	7
3	0	0	21	980	0	75	3	16	2	4
4	6	5	1	3	950	2	8	13	0	30
5	7	1	0	7	0	897	26	0	2	2
6	29	0	2	0	6	0	909	0	1	0
7	0	4	1	3	13	0	0	874	0	2
8	4	0	4	2	2	18	33	12	981	15
9	16	1	0	0	17	7	0	9	4	931

Table 7. Confusion matrix: combination of column and row models (FF + BF vectors)

	0	1	2	3	4	5	6	7	8	9
0	988	0	0	0	0	0	3	0	2	0
1	0	986	0	0	0	0	1	0	0	0
2	1	8	993	3	0	0	0	21	0	0
3	1	0	1	995	0	7	0	12	1	2
4	2	2	1	0	983	0	0	2	0	23
5	0	1	0	1	0	971	24	1	0	1
6	6	1	0	0	2	0	966	0	1	0
7	0	1	4	0	1	0	0	961	0	1
8	2	1	1	1	0	12	6	2	995	9
9	0	0	0	0	14	10	0	1	1	964

Table 7 shows the final confusion matrix related to the experiment (d) presented in Table 5. We can observe that the use of complementary information (foreground/background features + column/row models) has shown to be a promising way to reduce those confusions shown in Table 6. However, there is still some confusion between classes 2-7, 4-9, and 9-4.

In a final experiment we have used a more robust experimental protocol, in which the database is composed of 195,000 samples for training, 28,000 for validation and 60,000 for testing. The recognition rate for the testing set was 97,9%.

5. Conclusion

In this paper we have combined complementary features extracted from both foreground and background of digit images. These features were combined in an HMM-based classifier composed of 20 models: 10 column-based and 10 row-based. A zoning scheme based on column and row models provides a way of dividing the digit into zones without making the features size invariant. The experiments have shown that HMMs can provide high recognition performance (98%) close to those provided by the use of neural networks (99%). This is very important since HMMs have shown to be more appropriate to model handwriting knowledge related to the interaction between adjacent characters in words or numeral strings. Further work can be done in order to develop new features based on structural information.

6. References

[1] Trier O.D., Jain A.K. and Taxt T. Feature Extraction Methods for Character recognition - a Survey. *Pattern Recognition*, Vol 29, No. 4, pp. 641-662, 1996.

[2] Loncaric S. A Survey of Shape Analysis Techniques. Pattern Recognition, Vol. 31, No. 8, pp. 983-1001, 1998.

[3] Kimura F. and Shridhar M. Segmentation-recognition algorithm for handwritten numeral strings. Machine Vision Applications, No. 5, pp. 199-210, 1992.

[4] Gader P.D. and Khabou M. A. Automatic Feature Generation for Handwritten Digit Recognition. IEEE Transactions on Pattern Analysis and Machine Intellingence, Vol. 18, No. 12, pp. 1256-1261, December 1996.

[5] Cheung K. and Yeung D. A Bayesian Framework for Deformable Pattern Recognition with Application to Handwritten Character Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 12, December 1998.

[6] Pavlidis T. A vectorizer and feature extractor for document recognition. Comput. Vision and Graphic Image Process, 35, 111-127, 1986.

[7] Hirano T., Okada Y. and Yoda, F. Structural Character Recognition Using Simulated Annealing. Proceedings of the Fourth International Conference on Document Analysis and Recognition (ICDAR'97), Vol. 2, pp. 507-510, 1997.

[8] Lee L.L and Gomes N.R. Disconnected Handwritten Numeral Image Recognition. Proceedings of the Fourth International Conference on Document Analysis and Recognition (ICDAR'97), Vol. 2, pp. 467-470, 1997.

[9] Cai J. and Liu Z. Integration of Structural and Statistical Information for Unconstrained Handwritten Numeral Recognition. Proc. of the Fourteenth International Conference on Pattern Recognition (ICPR'98), Vol. I, pp. 378-380, 1998.

[10] Heute L., Paquet T., Moreau J.V., Lecourtier, Y. and Olivier, C. A structural/statistical feature-based vector for handwritten character recognition. Pattern Recognition Letters, 19 (1998), pp. 629-641.

[11] Oliveira L. S., Sabourin R., Bortolozzi F., and Suen C.Y. Feature Selection Using Multi-Objective Genetic Algorithms for Handwritten Digit Recognition, 16th International Conference on Pattern Recognition (ICPR 2002), vol I pp. 568-571, Quebec City, CA, August 11-15, 2002.

[12] Britto Jr, A.S., Sabourin R., Bortolozzi F. and Suen C.Y. A String Length Predictor to Control the Level Building of HMMs for Handwritten Numeral Recognition, 16th International Conference on Pattern Recognition (ICPR 2002), vol. IV pp.31-34, Quebec City, CA, August 11-15, 2002.

[13] Wang X. Durationally constrained training of HMM without explicit state duration PDF. Inst. of Phonetic Sciences, University of Amsterdam, Proceedings 18, pp. 111-130, 1994.

[14] Procter S. and Elms A. J. The recognition of handwritten digit strings of unknown length using hidden Markov models. Proceedings of the Fourteenth International Conference on Pattern Recognition (ICPR'98), pp. 1515-1517, 1998.

[15] Rabiner, L. R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, Vol. 77, No. 2, pp.257-286, Feb. 1989.

[16] Makhoul J., Roucos S., and Gish H. Vector Quantization in Speech Coding. Proceedings of the IEEE, vol. 73, pp. 1551-1558, Nov. 1985.