A Two-Stage HMM-Based System for Recognizing Handwritten Numeral Strings

Alceu de S. Britto Jr [a,b], Robert Sabourin [c,d], Flavio Bortolozzi [a], and Ching Y. Suen [d]

[a] Pontifícia Univ. Católica do Paraná , R. Imaculada Conceição, 1155 - Curitiba (PR) 80215-901 -Brazil
[b] Univ. Estadual de Ponta Grossa, Pr. Santos Andrade s/n, Ponta Grossa (PR) 84100-000 - Brazil
[c] École de Technologie Supérieure, 1100 Rue Notre Dame Ouest - Montreal (QC) H3C1K3 - Canada
[d] Centre for Pattern Recognition and Machine Intelligence , 1455 de Maisonneuve Blvd. West, Suite GM 606 - Montreal (QC) H3G 1M8 – Canada

Abstract

In this paper we propose a handwritten numeral string recognition method composed of two HMM-based stages. The first stage uses an implicit segmentation strategy based on string contextual information to provide multiple segmentation-recognition paths. These paths are verified and re-ranked by using a Verification stage based on a digit classifier. It allows the use of two sets of features and numeral models: one taking into account of both segmentation and recognition aspects in an implicit segmentation based strategy, and another considering just recognition aspects of isolated digits. The two system stages have shown to be complementary in the sense that the Verification stage has shown to be a promising idea to deal with the loss in terms of recognition performance brought by the necessary tradeoff between segmentation and recognition carried out in the first system stage.

1. Introduction

Over the past years the recognition of handwritten numeral strings has gained special attention due to its important role in a large number of potential applications such as reading zip codes on envelopes, courtesy amounts on checks, and data filled on tax and census forms. From the literature different approaches have been proposed to deal with this challenging problem. More conventional methods [1,2], such a segmentation-based approach, rely on the prior segmentation of strings into digits. In these methods the segmentation process is strongly based on heuristics and done before the recognition of each digit. This separation between segmentation and recognition has frequently become unreliable when the digits are touching or overlapping each other, broken or noisy. Moreover, the use of heuristics necessary to drive a blind search for numerals in the string reduces the accuracy of these methods.

Another approach is called segmentation-free, which has shown advantages in dealing with broken and touching numerals. Notice that 'segmentation-free' does not mean that there is no segmentation. This name is used in the sense that the segmentation process is held by an isolated digit classifier. These methods require either an implicit or an explicit segmentation strategy. In the first strategy, the recognition is performed simultaneously with the segmentation process [3,4], while in the second one the recognition is performed after the segmentation process, in order to search for the best way to assemble primitive segments or digit candidates to form the string [5-7].

The proposed method can be categorized as a segmentation-free approach that avoids a prior segmentation of the string into digits by using an implicit segmentation strategy. In this method the challenge consists of finding the best compromise between segmentation and recognition. To deal with this problem we propose a two-stage system. It allows the use of two sets of features and numeral models: one taking into account of both segmentation and recognition aspects, and another considering just the recognition aspects.

The general architecture is shown in Figure 1. The String Context-Based Stage (SCB) finds the N best segmentation-recognition paths for a given numeral string. For this purpose, a dynamic programming is used to match numeral Hidden Markov Models (HMMs) against to a given string. The 10 numeral HMMs ($\lambda_c^0, \lambda_c^1, ..., \lambda_c^9$) used in this stage are trained on isolated digits, but considering contextual information regarding string slant and intra-string size variations. In addition, features extracted from the foreground pixels of the string image columns contemplate both segmentation and recognition processes.

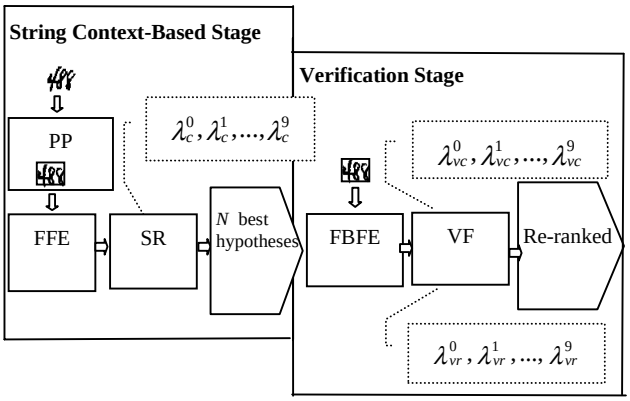


Figure 1: System architecture

The objective of the Verification Stage is to re-rank the *N* best segmentation-recognition paths provided by the first system stage using a powerful isolated digit recognizer. This stage consists of an HMM-based digit classifier trained on isolated digits without taking into account of string contextual information. A new set of features based on foreground and background information is used in order to improve the recognition performance of the numeral HMMs. Moreover, 10 additional numeral HMMs ($\lambda_{vr}^0, \lambda_{vr}^1, ..., \lambda_{vr}^9$) based on the rows of the numeral images are combined with the column-based models ($\lambda_{vc}^0, \lambda_{vc}^1, ..., \lambda_{vc}^9$) during the digit recognition process.

This paper is divided into 5 sections. Section 2 describes each module of the first system stage. Section 3 presents the Verification stage. Experimental results and conclusion are shown in Sections 4 and 5, respectively.

2. String Context-Based Stage

The SCB stage is based on an implicit segmentation strategy. The general objective is to provide the *N* best segmentation-recognition paths for a given numeral string. To this end, it is composed of three modules: Preprocessing (PP), Foreground Feature Extraction (FFE) and Segmentation-Recognition (SR). The numeral HMMs are trained on isolated digits, but considering contextual information (CI) regarding string slant and intra-string size variation.

2.1 Preprocessing Module

The string slant is corrected in order to reduce the script variability. The method proposed in [8] has also shown to be really helpful in alleviating overlapping between adjacent digits which interfere the columns of pixels extracted from them. The smooth method described in [9] is used, before and after the slant correction, in order to reduce possible artifacts on the string contour.

2.2 Foreground Feature Extraction

The objective of the FFE method is to contemplate both segmentation and recognition in an implicit segmentation strategy. It consists of scanning the numeral string from left-to-right, while local and global features are calculated taking into account the foreground pixels of the image columns. The local features are based on transitions from background to foreground pixels and *vice versa*.

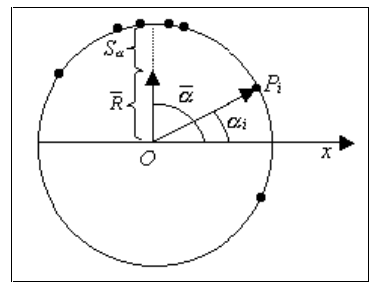


Figure 2. Circular mean direction $\overline{\alpha}$ and variance S_α for a distribution $F(\alpha_i)$

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° instead of 180° calculated by conventional estimators. Let $\alpha_1, ..., \alpha_i, ..., \alpha_N$ be a set of directional observations with distribution $F(\alpha_i)$ and size *N*. Figure 2 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:

$$(\cos(\alpha_i),\ \sin(\alpha_i)) \tag{1}$$

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 $(\overline{OP_1}, ..., \overline{OP_i}, ..., \overline{OP_N})$. The center of gravity $(\overline{C}, \overline{S})$ of the *N* coordinates $(cos(\alpha_i), sin(\alpha_i))$ is defined as:

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

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

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

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

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

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

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

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

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 3 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.

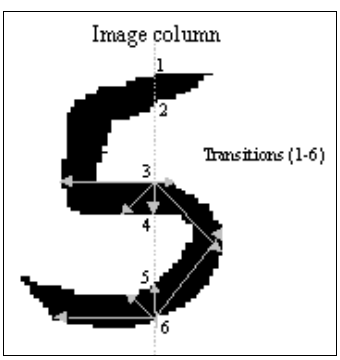


Figure 3. Transitions in a column image of numeral 5, and the directional observations to estimate the mean direction for transitions 3 and 6

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 horizontal projection (HP) of black pixels for each column, and the derivative of HP between adjacent columns. This constitutes a total of 34 features extracted from each column image and normalized between 0 and 1.

2.3 Segmentation-Recognition Module

The SR module matches numeral HMMs against the preprocessed string using the Level Building Algorithm [3,4]. The HMMs ($\lambda_c^0, \lambda_c^1, ..., \lambda_c^9$) representing numeral classes are trained using a special data set extracted from the original NIST SD19. To create this data set we developed an automatic process based on the digit classifier proposed in [10]. In this process, a handwritten numeral string is selected and segmented into digits when all of its components are recognized as isolated digits. Moreover, the string recognition result must correspond to that labeled by NIST. The objective is to obtain a data set in which the isolated digits have a link to their original strings. This allows the use of contextual information during training of our numeral models. The general idea is to keep the same experimental conditions during system training and testing.

We use this special data set to create training and validation sets composed of 50,000 and 10,000 isolated digits, respectively. The contextual information used during training concerns string slant and intra-string size variation. The contribution to string recognition by using the slant estimated from the original string in order to correct the isolated digits for training of an implicit segmentation based system is presented in [8]. The intra-string size variation is shown in Figure 4, which corresponds to the distances of each digit to the top and base line of the string bounding box.

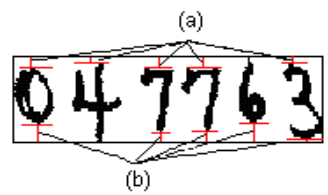


Figure 4. Intra-string size variation: (a) distance from the top of the bounding box; (b) distance from the base of the bounding box

To deal with this problem we use the string height as contextual information. The extraction of features from each training sample is done taking into account the string height instead of the digit height. On the other hand, the inter-string size variation is treated with size-invariant features.

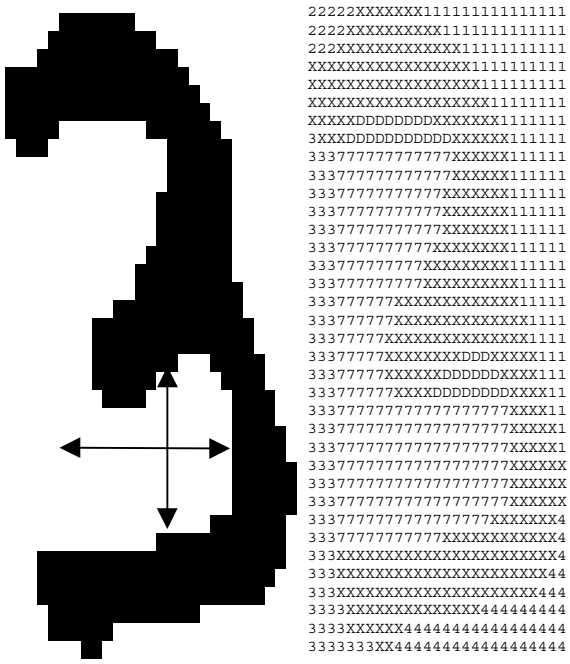
The structure of the numeral HMMs was experimentally defined. The best results were obtained using a Bakis topology, where the number of states for each numeral class was optimized taking into account the methodology defined in [11]. Different codebook lengths

were also evaluated. The best results were achieved using a codebook with 256 entries.

3. Verification Stage

The verification stage is composed of 20 numeral HMMs: 10 based on the image columns ($\lambda_{vc}^{0}, \lambda_{vc}^{1}, ..., \lambda_{vc}^{9}$) and 10 based on the image rows ($\lambda_{vr}^{0}, \lambda_{vr}^{1}, ..., \lambda_{vr}^{9}$) of the digit images. These complementary HMM models are used by the Verification module to re-rank the segmentation-recognition hypotheses provided by the SR module.

The same scheme used for optimizing the numeral HMMs of the SR module is applied to define the models of the Verification stage. However, these new models are trained without considering contextual information. The objective is to obtain a classifier more powerful in terms of isolated digit recognition performance than that used in the SR module. Even the database used to create these new models is different. The special isolated digit database used to keep string contextual information may be not representative of all digit recognition problems. For this reason, we use isolated digits from the original NIST SD19. The training and validation databases are composed of 106,646 and 20,000 samples, respectively.



Concavity features computed for the 1[a] column

0	3	29	0	0	0	0	0	0	0	0	0	0
1	2	3	4	5	6	7	8	9	A	B	C	D

Figure 5: Example of concavity features

The Foreground-Background Feature Extraction (FBFE) method completes the FFE method with background information. For this purpose, the background pixels of the digit image are labeled using concavity configurations. The label for each white pixel is chosen based on the Freeman code with 4 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 4 more configurations as defined in [12], in order to detect more precisely the presence of loops. Finally, each concavity feature representing a column or row of the image corresponds to the number of white pixels that belong to a specific concavity configuration (see Figure 5). The total length of the FBF vector is 67 (34 Foreground + 13 Background features).

4. Experimental Results

A rigorous experimental protocol has been used in order to construct and evaluate our string recognition system. The experiments are performed on isolated digits and numeral strings of different lengths extracted from NIST SD19 database. In this first version of our system, the string recognition is based on an informed strategy, i.e., the string length is known. The objective of using this strategy is to evaluate the system under different conditions, and at the same time adjusting some important aspects regarding string normalization, feature extraction and HMM parameters. Table 1 shows the isolated digit recognition results using the original NIST SD19.

	Validation	Testing
Column models	96.08	94.28
Row models	95.60	93.12
Combination (column x row models)	98.30	96.10

Table 1: Isolated digit recognition in percentage

The training, validation and testing database consist of 106,646, 20,000 and 20,000 digit samples, respectively. The experiments on numeral strings are carried out using 12,802 numeral strings extracted from the hsf_7 series of the NIST SD19 and distributed into 6 classes: 2_digit (2,370), 3_digit (2,385), 4_digit (2,345) and 5_digit (2,316), 6_digit (2,169) and 10_digit (1,217) strings. In addition, to evaluate the system in terms of touching digits we use a subset of data containing 2,069 touching digit pairs (TDPs) also extracted from NIST SD19.

During these experiments the SR module provided 10 segmentation-recognition paths for each numeral string. In the Verification stage, the FBFE module uses the segmentation points of each path as delimiters in the preprocessed string image to calculate new features based

on columns and rows for each digit candidate. The recognition result of the first stage is verified using the new set of features and numeral HMMs. We combine the recognition results of the SCB and Verification stages by summing the log of their probabilities. Table 2 shows the top 5 recognition results of the first stage of our system, while Table 3 presents the top 5 recognition results after the Verification stage.

Class	Top (1)	Top (2)	Top (3)	Top (4)	Top (5)
2_digit	90.29	95.35	96.91	97.25	97.46
3_digit	85.87	91.99	92.83	93.20	93.33
4_digit	81.66	89.38	91.17	91.81	91.98
5_digit	79.97	87.69	89.55	90.50	90.67
6_digit	76.76	85.85	87.32	88.47	88.84
10_digit	68.44	73.62	74.28	74.44	74.44
Global	81.65	88.57	90.00	90.62	90.81
TDPs	79.51	88.44	91.64	92.65	93.19

Table 2: SCB stage - numeral string recognition results

Class	Top (1)	Top (2)	Top (3)	Top (4)	Top (5)
2_digit	95.23	97.59	98.35	98.48	98.57
3_digit	92.62	95.60	96.18	96.27	96.28
4_digit	92.11	95.35	95.95	96.03	96.12
5_digit	90.00	93.96	94.52	94.69	94.73
6_digit	90.09	94.05	94.88	94.92	95.02
10_digit	86.94	90.30	90.38	90.46	90.46
Global	91.57	94.86	95.47	95.57	95.63
TDPs	89.61	94.39	95.36	95.70	95.84

Table 3: SCB + Verification stage - numeral string recognition results

We can see a significant improvement in the recognition performance by using the verification stage. The main reason is that the foreground features and the numeral HMMs based on contextual information may contemplate both segmentation and recognition tasks in an implicit segmentation approach, but they do not provide a strong enough recognition power. An error analysis in this stage showed that most of the time the system mistakes are related to mis-classification. This means that the first stage was able to find the right segmentation points for a given string, but sometimes it was not enough to distinguish between 5 and 3 or 4 and 9.

5. Conclusion and Future Works

The proposed handwritten numeral string recognition method is composed of two stages. The first stage uses an implicit segmentation strategy and string contextual information to provide multiple segmentation-recognition paths. The hypotheses generated by the first stage are re-ranked in a Verification stage, which is based on a digit classifier.

The Verification stage has shown to be a promising idea to deal with the loss in terms of recognition performance brought by the necessary tradeoff between segmentation and recognition tasks carried out in the first system stage. Our future work will evaluate the use of a pause model built in the numeral models of the SR module to represent the interactions between adjacent digits in strings.

References

[1] Shi Z., Srihari N. Shin Y.C. and Ramanaprasad V. A System for Segmentation and Recognition of Totally Unconstrained Handwritten Numeral Strings. ICDAR'97, Ulm, Germany, 18-20 August 1997, Vol 2, pp. 455-458.

[2] Ha T.M., Zimmermann M. and Bunke H. Off-Line Handwritten Numeral String Recognition by Combining Segmentation-Based and Segmentation-Free Methods. Pattern Recognition, Vol. 31, No. 3, 257-272, 1998.

[3] Elms A.J., Procter S. and Illingworth J. The Advantage of using an HMM-bases Approach for Faxed Word Recognition. International Journal on Document Analysis and Recognition. pp. 18-36, 1998.

[4] Elms A.J. The representation and recognition of text using hidden Markov models (PhD thesis). Department of Eletronic and Electrical Engineering, University of Surrey, April 1996.

[5] Oliveira L.S., Lethelier R., Bortolozzi F., Sabourin R. A new approach to segment handwritten digits. IWFRH'2000, Amsterdam, The Netherlands, 11-13 Septembre 2000, pp.577-582.

[6] Lethelier E., Leroux M., and Gilloux M. An Automatic Reading System for Handwritten Numeral Amounts on French Checks. ICDAR' 95, Montreal, Canada, 14-16 August 1995, Vol. 1, pp.92-97.

[7] Gader P.D., Keller J.M., Krishnapuram R., Chiang J. and Mohamed M.A. Neural and Fuzzy Methods in Handwriting Recognition. Computer Magazine, pp. 79-85, February, 1997.

[8] Britto A.S., Sabourin R., Lethelier E., Bortolozzi F., and Suen C.Y. Improvement in Handwritten Numeral String Recognition by Slant Normalization and Contextual Information. IWFRH'2000, Amsterdam, The Netherlands, 11-13 Septembre 2000, pp. 323-332.

[9] Suen C.Y., Nadal C., Legault R., Mai T.A., and Lam L. Computer recognition of uncostrained handwritten numerals, Proc IEEE, 80(7):1162-1180, 1992.

[10] Suen, C.Y., Liu K. and Strathy N.W. Sorting and Recognizing Cheques and Financial Documents. Proc. of Third IAPR Workshop on Document Analysis Systems, 16-20 August 1998, Brisbane, Australia, pp. 1-18.

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

[12] 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, pp. 629-641, 1998.