An Implicit Segmentation-based Method for Recognition of Handwritten Strings of Characters

Paulo Rodrigo Cavalin
Pontifícia Universidade Católica do Paraná
Curitiba, Brazil
cavalin@ppgia.pucpr.br

Alceu de Souza Britto Jr.
Pontifícia Universidade Católica do Paraná
Universidade Estadual de Ponta Grossa
Curitiba, Brazil
alceu@ppgia.pucpr.br

Flávio Bortolozzi
Pontifícia Universidade Católica do Paraná
Curitiba, Brazil
fborto@ppgia.pucpr.br

Robert Sabourin
École de Technologie Supériere
Montreal, Canada
robert.sabourin@etsmtl.ca

Luiz E. Soares Oliveira
Pontifícia Universidade Católica do Paraná
Curitiba, Brazil
soares@ppgia.pucpr.br

ABSTRACT

This paper describes an implicit segmentation-based method for recognition of strings of characters (words or numerals). In a two-stage HMM-based method, an implicit segmentation is applied to segment either words or numeral strings, and in the verification stage, foreground and background features are combined to compensate the loss in terms of recognition rate when segmentation and recognition are performed in the same process. A rigorous experimental protocol shows the performance of the proposed method for isolated characters, numeral strings, and words.

1. INTRODUCTION

The recognition of handwritten words and numeral strings were researched as two different problems in the past few years. A considerable number of methods were developed to recognize either words or numeral strings. That splitting of the problem has resulted in methods with great performance for one of those problems, but not suitable for both.

In spite of that splitting, both problems are treated in the same way. The recognition can be done in a holistic or in an analytic approach. The former is suitable to deal with both problems at the same time, but the applications are constrained to small lexicon size as in bank check processing systems [1, 2]. By constrast, the last one is suitable to deal with greater lexicon size, but the segmentation of string into segments that relate to characters is required. That is not a trivial task due to problems such as touching, overlapping, and broken characters in numeral strings, or even, the ambiguity encountered in case of handwritten cursive words (see Figure 1).

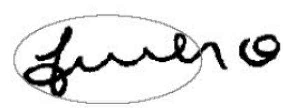

Figure 1: Ambiguity in handwritten words: "june","fune" or "fine"?.

To deal with the segmentation problem, analytical methods employ segmentation-based recognition strategies where the segmentation can be explicit [3, 4] or implicit [5, 6]. Explicit when the segmentation is based on cut rules, and implicit when each pixel column is a potential cut location. In the case of explicit segmentation several algorithms have been proposed during the last years. They normally take into consideration a set of heuristics and information of the foreground [7], background [8], or a combination of them [9] in order to generate potential segmentation cuts. Generally, the heuristics used to make the algorithm robust make it specific for the applied problem, and a good segmentation algorithm for numeral strings may not have the same performance for words, and vice-versa.

An alternative aimed at avoiding the prior segmentation of the string has been the use of implicit segmentation-based methods to integrate segmentation and recognition processes. A promising approach to achieve this has been based on Hidden Markov Models (HMMs). This approach was originally developed for the field of speech recognition [10], where it has been applied with much success. The benefits of applying such a technique to recognize printed words have been shown in [11]. In [13], the method proposed in [12] was adapted for handwritten numeral strings. Other works, like in [14], have also shown that approach as a good method for reconizing handwritten words. From these stud-

SAC'06 April 23-27, 2006, Dijon, France

ies, we may conclude that such an approach is a promising way of integrating segmentation and recognition to deal with the difficulties encountered in processing both handwritten numeral strings and handwritten words.

However, Britto et al. [15, 16] have observed some cost attached to that integration, which is a loss in recognition performance caused by combining segmentation with recognition. So, a segmentation/verification strategy shows to be suitable to compensate the loss in terms of recognition caused by the implicit-segmentation strategy.

In this paper, we propose a method for the recognition of both handwritten numeral strings and handwritten words, combining segmentation and recognition taking into account the tradeoff caused by an implicit segmentation method. The method is based on a two-stage recognition strategy that enables the use of two sets of features and character models: one taking into account both the segmentation and recognition aspects in an implicit segmentation-based process, and another considering just the recognition aspects in a further verification process (see Figure 2). The recognition performance of the feature set is evaluated on isolated characters, and the whole method is evaluated on numeral strings and words.

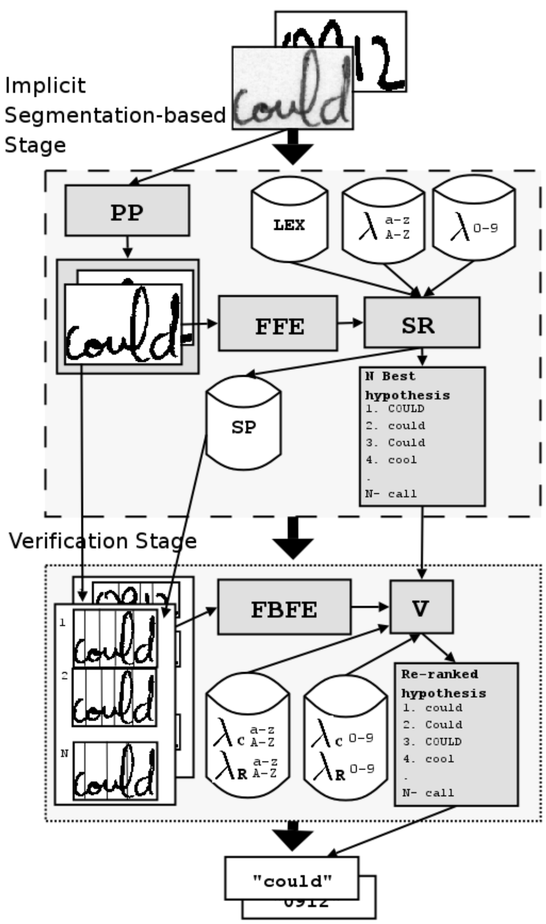

Figure 2: System architecture.

This paper is divided into 4 sections. Section 2 describes each module of the proposed method. Section 3 presents the experimental results, while Section 4 shows the conclusion and future works.

2. PROPOSED METHOD

The proposed method can be categorized as a segmentation-based approach that avoids a prior segmentation of the string into characters by using an implicit segmentation strategy. In this method the challenge consists in 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 character models: one taking into account both segmentation and recognition aspects, and another considering just the recognition aspects.

The general architecture is shown in Figure 2. First, the N best segmentation-recognition paths for a given string of characters are found. For this purpose, a dynamic programming is used to match character Hidden Markov Models (HMMs) against to a given string. The character HMMs used in this stage are trained on isolated characters based on features extracted from the foreground pixels of the string image columns. The objective of these features is to contemplate both segmentation and recognition processes. 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 character recognizer. This stage consists of an HMM-based character classifier trained on isolated characters. A new set of features based on foreground and background information is used in order to improve the recognition performance of the character HMMs. Moreover, additional character HMMs based on the rows of the string images are combined with the column-based models during the character recognition process. In the following sub-sections we describe all the modules of the architecture in details.

2.1 Implicit segmentation-based stage

The general objective is to provide the N best segmentation-recognition paths for a given string: recognition results and segmentation points (SP). To this end, it is composed of three modules: Preprocessing (PP), Foreground Feature Extraction (FFE) and Segmentation-Recognition (SR). The character HMMs are trained on isolated digits and a lexicon (LEX) is used in case of word recognition.

2.1.1 *Preprocessing module.*

The string slant is corrected in order to reduce the script variability. The method proposed in [17] has also shown to be really helpful in alleviating overlapping between adjacent digits which interferes the columns of pixels extracted from them. The smoothing method described in [18] is used, before and after the slant correction, in order to reduce possible artifacts on the string contour.

2.1.2 *Foreground feature extraction (FFE).*

The objective of the foreground features is to contemplate both segmentation and recognition in an implicit segmentation strategy. It consists of scanning the 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.

For each transition, the mean direction and corresponding variance are obtained by means of statistic estimators. These estimators are more suitable for directional observa-

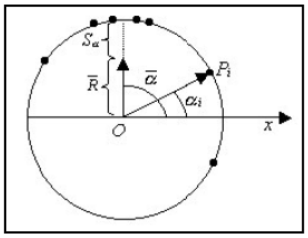

Figure 3: Circular mean direction $\overline{\alpha}$ and variance S_α for a distribution $F(\alpha_i)$.

tions, 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 3 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:

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

The circular mean direction $\overline{\alpha_i}$ 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 ($\overline{C}, \overline{S}$) is defined as:

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

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

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

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

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

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

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

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

To estimate $\overline{\alpha}$ and S_α for each transition of a numeral image, we have considered $\{0^o, 45^o, 90^o, 135^o, 180^o, 225^o, 270^o, 315^o\}$ 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 4, the transitions in a column of the word image "held" are enumerated from 1 to 6, and the possible directional observations from transitions 3 and 5 are shown.

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

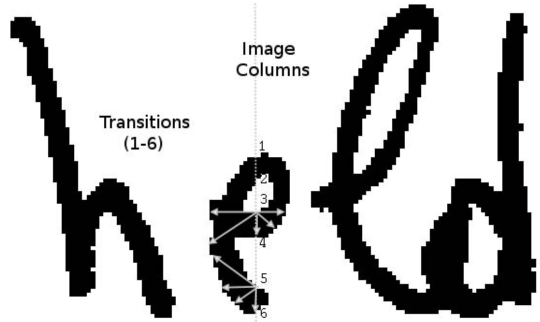

Figure 4: Foreground features: transitions in a column image of word "held", and the directional observations to estimate the mean direction for transitions 3 and 5.

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 extracted from each column image and normalized between 0 and 1.

2.1.3 Segmentation-recognition module (SR).

The SR module matches character HMMs against the preprocessed string using the Level Building Algorithm [10]. The character HMMs (λ_c^0, λ_c^1, ..., λ_c^9) and (λ_c^a, λ_c^b, ..., λ_c^z, λ_c^A, λ_c^B, ..., λ_c^Z) representing character classes are trained on isolated samples.

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 character class was optimized taking into account the mean length of the observation sequences on the training database. Different codebook lengths were also evaluated. The best results were achieved using a codebook with 256 entries.

2.2 Verification stage

The verification stage is composed of 62 character HMMs: 36 based on the image columns (λ_c^0, λ_c^1, ..., λ_c^9, λ_c^a, λ_c^b, ..., λ_c^z, λ_c^A, λ_c^B, ..., λ_c^Z) and 36 based on the image rows (λ_r^0, λ_r^1, ..., λ_r^9, λ_r^a, λ_r^b, ..., λ_r^z, λ_r^A, λ_r^B, ..., λ_r^Z) of the digit images. These complementary HMMs are used by the Verification module (V) to re-rank the segmentation-recognition hypotheses provided by the SR module.

The same scheme used for defining the character HMMs of the SR module is applied to define the models of the Verification stage. However, these new models are trained considering additional features extracted from the background of the string images. The objective is to obtain a classifier more powerful in terms of isolated character recognition performance than that used in the SR module.

The FBFE extraction method completes the FFE feature set with background information. For this purpose, the background pixels of the character image are labeled using concavity configurations. The label for each white pixel is chosen based on the directional code with 4 directions. Each direction is explored until the encounter of a black pixel or the limits imposed by the character 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 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. The total length of the FBFE vector is 47 (34 Foreground + 13 Background features).

3. 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 characters, numeral strings of different lengths extracted from NIST SD19 database [19], and unconstrained words available on the IAM database [20].

3.1 Experiments on isolated characters

The isolated characters available on the NIST SD19 were used in these experiments. Table 1 shows our experimental protocol and recognition rates for isolated digits, uppercase and lowercase characters.

Table 1: Experimental protocol and recognition rates on isolated characters.

Class	#Train	#Valid	#Test	RR%
digits(10)	195,000	28,000	60,000	98.0
uppercase(26)	37,440	12,092	11,941	90.0
lowercase(26)	37,440	11,578	12,000	84.0
upper/lower(52)	74,880	23,670	23,941	87.0

We have used the same experimental protocol than Koerich [21] for upper/lowercase characters. The author has achieved 92%, 84% and 85% for upper, lower and upper/lower, respectively. When both, upper and lowercase, are considered in the same experiment, our feature set has shown to be better. The reason is that, the features based on columns and rows have shown to be more suitable to discriminate classes where the difference is just the scale, such as C and c.

The recognition rate of 98% for isolated digits is very close to works reported in the literature, in which the objective is just to recognize isolated digits, such as Suen in [18].

3.2 Experiments on numeral strings

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), 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 (SP) 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 Implicit-Segmentation based and the 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.

Table 2: Implicit Segmentation stage - numeral string recognition results.

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 3: IImplicit Segmentation stage + Verification 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

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 misclassification. 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.

3.3 Experiments on words

The experiments on words are carried out using 18,624 unconstrained word images available in the IAM database, distributed as follows: 12,651 for training, 3,168 for validation and 2,805 for testing. During the experiments we have considered four lexicons sizes containing 10, 100, 1000 and 3.717 word classes.

Table 4 shows the recognition results for top 1 and 5. In Table 4, we can also observe some improvement in terms of recognition performance after the verification stage. However, the improvement is more evident for large lexicons, where the recognition error of the first stage is greater.

4. CONCLUSION AND FUTURE WORKS

The experimental results of this method for reconizing numeral strings and words have shown promising perfor-

Table 4: Performance on word recognition (%).

	Implicit -based stage		After Verification stage	
#Lex	Top 1	Top 5	Top 1	Top 5
10	96.7	98.9	97.4	99.1
100	93.0	96.5	93.9	96.0
1,000	83.6	92.5	86.0	93.0
3,717	74.7	87.8	78.0	88.2

mance to have a method to recognize any kind of handwritten string. This two-stage method has enabled the use of a implicit segmentation-based approach that is adjusted by the training samples and a verification strategy to compesate the loss in terms of recognition. The feature set used in the first stage has shown to be suitable to the task proposed for both words and numeral strings, and the feature set used in the verification stage has shown good performance to recognize both digits and letters.

We may improve the performance of the proposed method by further development in a number of areas. One way to do that is by further investigating feature sets, since this method enables the combination of different features at each stage. For instance, a new set of foreground features can be defined to improve the segmentation-recognition performance of the first stage, while new features with powerful recognition performance can be evaluated in the second stage. Another point to be investigated is the increasing of the training data, what can adjust better the segmentation and the recognition.

5. REFERENCES

[1] Dimauro, G., Impedovo, S., Pirlo, G., Salzo, A.: Automatic bankcheck processing: A new engineered system. In S.Impedovo et al, editor, International Journal of Pattern Recognition and Artificial Intelligence, World Scientific, pp.467-503, 1997.

[2] Freitas, C., Bortolozzi, F., Sabourin, R.: Handwritten isolated word recognition: An approach based on mutual information for feature set validation. In Proc. 6th International Conference on Document Analysis and Recognition, Seattle-USA, September, pp. 665-669, 2001.

[3] El-Yacoubi, A., Gilloux, M., Sabourin, R., Suen, C. Y.: An hmm-based approach for off-line unconstrained handwritten word modeling and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(8):752-760, 1999.

[4] Arica, N., Yarman-Vural, F. T.: Optical character recognition for cursive handwriting. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(6):801-813, 2002.

[5] Gillies, M.: Cursive word recognition using hidden markov models. In Proc. Fifth U.S. Postal Service Advanced Technology Conference, pages 557-562, 1992.

[6] Cho, W., Lee, S. W., Kim, J. H.: Modeling and recognition of cursive words with hidden markov models. Pattern Recognition, 28(12):1941-1953, 1995.

[7] Kim, K. K., Kim, J. H., Suen, C. Y.: Segmentation-based recognition of handwritten touching pairs of digits using structural features. Pattern Recognition, 23(1):13-21, 2002.

[8] Pal, U., Belaid, A., Choisy, C.: Touching numeral segmentation using water reservoir concept. Pattern Recognition Letters, 24:261-272, 2003.

[9] Oliveira, L. S., Sabourin, R., Bortolozzi, F., Suen, C. Y.: Automatic recognition of handwritten numerical strings: A recognition and verification strategy. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(11):1438-1454, 2002.

[10] Rabiner, L.: 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.

[11] Bose, C. B., Kuo, S.: Connected and degraded text recognition using hidden Markov model. Pattern Recognition, Vol. 27, No. 10, pp. 1345-1363, 1994.

[12] Elms, A. J., Procter, S., Illingworth, J.: The advantage of using an HMM-based approach for faxed word recognition. International Journal on Document Analysis and Recognition. pp. 18-36, 1998.

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

[14] Zimmermann, M., Bunke, H.: Hidden Markov model length optimization for handwriting recognition systems. International Workshop on Frontiers in Handwriting Recognition, Niagra-on-the-Lakes, pp. 369-374, 2002.

[15] Britto, A. S., Sabourin, R., Bortolozzi, F., Suen, C. Y.: An enhanced HMM topology in an LBA framework for the recognition of handwritten numeral strings, Proceedings of the International Conference on Advances in Pattern Recognition, Vol 1, pp. 105-114, Rio de Janeiro-Brazil, March 2001.

[16] Britto, A. S., Sabourin, R., Bortolozzi, F., Suen, C. Y.: A two-stage HMM-based systems for recognizing handwritten numeral strings. Proceedings of the International Conference on Document Analysis and Recognition, pp. 396-400, Seattle, USA, September 2001.

[17] Britto, A. S., Sabourin, R., Lethelier, E., Bortolozzi, F., Suen, C. Y.: Improvement in handwritten numeral string recognition by slant normalization and contextual information. Proceedings of the Seventh International Workshop on Frontiers on Handwriting Recognition, September 11-13, Amsterdam, The Netherlands, Vol. 1, pp. 323-332, 2000.

[18] Suen, C. Y., Nadal, C., Legault, R., Mai, T. A., Lam, L.: Computer recognition of unconstrained handwritten numerals. Proc. IEEE, Vol. 80. No. 7, pp. 1162-1180, 1992.

[19] Grother, P. J.: NIST Special Database 19 - handprinted forms and characters database. National Institute of Standards and Technology, March 1995.

[20] Bunke, H., Marti, U.: A full english sentence database for off-line handwriting recognition. Proc. of the 5th Int. Conf. on Document Analysis and Recognition, ICDAR 99. Bangalore, p. 705-708, 1999.

[21] Koerich, A. L.: Large vocabulary off-line handwritten word recognition. PhD thesis, École de Technologie Supérieure, Montreal-Canada, August, 2002.