

On-line Handwriting word recognition using a bi-character model

Sopheia PRUM

L3i Laboratory, University of La
Rochelle
La Rochelle, France
sopheia.prum@univ-lr.fr

Muriel Visani

L3i Laboratory, University of La
Rochelle
La Rochelle, France
muriel.visani@univ-lr.fr

Jean-Marc Ogier

L3i Laboratory, University of La
Rochelle
La Rochelle, France
jean-marc.ogier@univ-lr.fr

Abstract—This paper deals with on-line handwriting recognition. Analytic approaches have attracted an increasing interest during the last ten years. These approaches rely on a preliminary segmentation stage, which remains one of the most difficult problems and may affect strongly the quality of the global recognition process. In order to circumvent this problem, this paper introduces a bi-character model, where each character is recognized jointly with its neighboring characters. This model yields two main advantages. First, it reduces the number of confusions due to connections between characters during the character recognition step. Second, it avoids some possible confusion at the character recognition level during the word recognition stage. Our experimentation on significant databases shows some interesting improvements of the recognition rate, since the recognition rate is increased from 65% to 83% by using this bi-character strategy.

Keywords— *Handwriting recognition; Character and text recognition; Online documents.*

I. INTRODUCTION

The last 20 years have known an explosion of the number of mobile devices, offering many mobile services in a context where the technology has allowed the development of many different kinds of acquisition devices such as PDA, electronic tablets.... These devices store strokes of pen-tip movements, constituting an on-line signal (each element being represented by a time-code and the corresponding x-y coordinates). These technological improvements precipitated considerable activities on the on-line handwriting recognition problem. Indeed, this research activity started during the 1960s, knew a break in the 1970s, [1, 2], and has been re-activated in the 1980s with the development of new electronic tablets, the increase of the computational performances of mobile devices, and with the development of new recognition algorithms. Jointly with the on-line signal many methods consider additionally the shape of the characters (off-line signal), which may be reconstructed by using the on-line signal.

Two main approaches have been considered in this research area: global approaches and analytical approaches. Global approaches consider the word shape or on-line signal as a whole and try to recognize it. Systems relying on this approach therefore have to be trained with a large training set containing all the words in the lexicon. In analytic

approaches, input word shape or on-line signal will be segmented into individual characters (or sections of characters), each of which is recognized (generally independently) and combined to identify the whole word. Systems relying on an analytic approach need to be trained using a training set containing all the possible characters of each language. They may be adapted to different lexicons without any re-training. Due to these major advantages, the analytical approach has been strongly interested during the last few years [1, 3, 4]. However, the segmentation step remains a very difficult problem because of the possible connections between characters, the large variability of the handwriting due to different scriptwriters or different contexts, generating many segmentation confusions [5].

This paper is organized as follows. Our proposed method is described in section II, while section III provides an experimental assessment for this method and section IV concludes this paper.

II. OVERVIEW OF THE PROPOSED APPROACH

The proposed method relies on an analytic approach and an explicit segmentation method. The main idea is to recognize jointly pairs of neighboring characters taking into account the recognition results of the isolated character, before recognizing the whole words by considering the information at these two levels, our approach becomes more robust towards character segmentation errors. Our method is divided into 4 steps: pre-processing and segmentation, characters recognition, bi-characters recognition, and final words recognition process. The following subsection presents the details each of these steps.

A. Signal pre-processing and segmentation

After acquiring a stroke and normalizing its resolution, the stroke is segmented into graphemes using the segmentation method proposed by R.Amad [1]. In this method, each grapheme contains all the consecutive points in the stroke located between the corresponding lower and upper points (on the y coordinate), as illustrated in “Fig. 1”.



Figure 1. Example of segmentation result

Once the graphemes have been segmented, a N-levels graph is reconstructed, representing all the possible graphemes concatenation (for an example see Fig. 2). Each node was assumed as a character and is therefore introduced as the input of the isolated characters recognition system described in the next section.

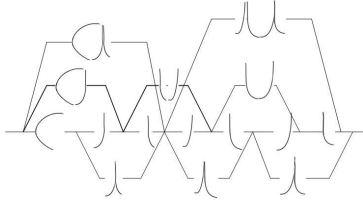


Figure 2. Graph reconstruction of the word "au" based on the segmentation illustrated in Fig. 1.

B. Isolated character recognition system

Character recognition system is obviously a crucial step for handwriting words recognition using analytic methods. The system we use relies on the two usual steps for pattern recognition: feature extraction and recognition.

1) Feature extraction

From the sequence of points constituting the on-line signal, an artificial image (off-line signal) of the word can be estimated. Our method relies on the combination of both on-line and off-line features (computed respectively from the on-line and off-line signals) in order to benefit from the advantages of both kinds of features and leads to a reduction of the dissimilarities between right-handed and left-handed scriptwriters' handwritings.

Seven families of off-line statistical and structural features described in [6] are used, including: 7 Hu invariant moments; the horizontal and vertical projections; the top, bottom, left and right profiles; the intersections with horizontal and vertical straight lines; the holes and concave arcs; the top, bottom, left and right extrema; the end points and junctions. Further we add more features including Radon invariants [7] and Zernike moments [8].

The on-line features we consider are the starting point and ending points of each stroke, the number of strokes, the starting direction of each stroke (gradient at the starting point), the ending direction of each stroke, etc...

Off-line and on-line feature vectors are further concatenated to obtain a feature vector of large size 254. In order to circumvent all the problems related to this large size and for efficiency matters, we use the SFFS feature selection method described in [9]. Using this method, the size of the combined vector is reduced to 45.

2) Isolated character recognition system

Our isolated character recognition system relies on the use of Support Vector Machine (SVM). The SVM is trained to discriminate between the 26 characters in the roman alphabet ($\{a, b, \dots, z\}$). The training set is composed of characters segmented from cursive words taken from the IRONOFF database [11] and written by different writers. The SVM outputs have been converted into probabilities by applying a softmax function. This database is cut into a

training set (used to train the SVM), a validation set (used to fit the parameters of the SVM), and test set used for performance evaluation. Table I shows the performances of our isolated character recognizer, highlighting its relatively good performances (85% recognition rate on the test set) in this context where segmentation is quite often inaccurate.

TABLE I. PERFORMANCE EVALUATION OF THE ISOLATED CHARACTER RECOGNITION SYSTEM (BOLD PERCENTAGES ARE RECOGNITION RATES)

Segmented characters	Training set	Test set	Validation set
Size	15000	5200	5200
		85%	84%

C. Bi-character recognition system

1) Objective

As mentioned in the introduction, one of the main disadvantages of the analytic approaches is that it tries to recognize each character independently, which may lead to confusions between a piece of character and a whole character. An example of such confusion is presented in "Fig. 3", where the analytic approach may recognize the bi-character "au" as the sequence of characters "ouui". Indeed, the letter "o" and "u" are respectively associated to high probabilities for the first and second graphemes at the level 2 of the graph, explaining this confusion. In order to avoid these confusions, we introduced a bi-character based model in recognizing each character taking into account its neighbor to ensure that the character sequences are correctly recognized at both the character and bi-character recognition levels. Indeed, if we consider the example in "Fig. 3" the bi-character "au" is associated to a higher probability than the bi-character "ou", which enables the bi-character based method to provide better recognition results than the classical analytic approach.

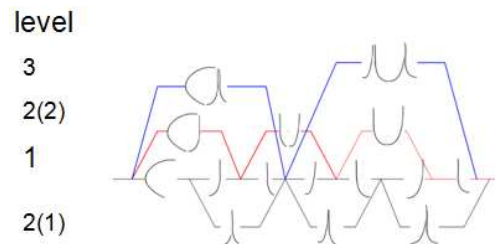


Figure 3. Example of a conflicting sequence of characters

2) Bi-character recognition method

As the isolated character recognizer, the bi-character recognition method relies on the use of SVMs (which outputs have also been transformed into probabilities by applying a softmax function). In order to assess the effectiveness of this approach we performed a series of preliminary experiments using a set of 30 French words from bank checks containing 68 different bi-characters. We have created a training set containing 25600 examples of these 68 bi-characters written by different scriptwriters. Experimental results are presented in Table II. Due to the higher number of classes to discriminate compared to the isolated character

recognizer, the recognition rates are lower (79% vs. 85% for the isolated character recognizer). However we aim at combining both recognizers in a more global scheme of word recognition using a Hidden Markov Model, and therefore we hope that the combination of our two recognizers will provide better robustness towards segmentation errors.

TABLE II. PERFORMANCES OF THE BI-CHARACTER RECOGNITION SYSTEM

Bi-characters	Training set	Test set	Validation set
Size	25600	6800	6800
		79%	79%

D. Word recognition system

For every cursive word, once the graph corresponding to all the possible grapheme concatenations has been computed (see section II.A. and “Fig. 2”) and enriched with the corresponding isolated characters and bi-characters probabilities (see sections II.B and II.C), this graph will be represented by using a Hidden Markov Model (HMM). The Viterbi algorithm is further used to decode the corresponding HMM graph and thus recognize the input cursive word. In order to provide a preliminary evaluation of our system, we consider a closed-world but this work may be extended to an open-world environment.

1) Representation by using a HMM

A Hidden Markov Model (HMM) is a model representing the transitions between states. It is usually denoted using $Y = \{A, B, II\}$ for a set of states $S = \{s_1, s_2, \dots\}$ and observations $O = \{o_1, o_2, \dots\}$.

The graph of all the possible graphemes concatenation (for an example see “Fig. 2”) will be represented by using a HMM model as described below:

- A: is the estimation of the recognition probabilities of each pair of nodes in the graph by using our bi-character recognition system.
- B: is the estimation of the recognition probabilities of each node in the graph by using our isolated character recognition system.
- II: contains neutral values (equal to 1), as we consider that every starting node may have the same probability.

2) Viterbi algorithm for word recognition

The Viterbi algorithm is a dynamic programming algorithm which aims at finding the most likely sequence of hidden states $S = \{S_1, S_2, \dots, S_t\}$ on the model Y with observations $O = \{O_1, O_2, \dots, O_t\}$ [10].

When applying the Viterbi algorithm to our problem, we consider that the observations are the sequences of characters taken from a lexicon. Then the Viterbi algorithm is used to find the sequence of states that yields the maximum probability. And finally the N words associated to the highest probabilities (ranked by descending probability order) are provided as the output of the system.

III. EXPERIMENTAL RESULTS

In order to evaluate the overall procedure of word recognition, we performed two series of experiments using

500 scripts of 30 words from French bank checks. These words are written by different writers and selected from the IRONOFF database [11]. The two series of experiments use different lexicons. The first lexicon contains 30 words (used in checks) and the second lexicon is composed of 100 words selected randomly among a lexicon of 500 words which contains only 68 bi-characters. We reconstruct a graph (see section II.A.) into 7 levels of graphemes basing on different experimentations.

The experimental results given in Table 3 show the recognition rates at the n first ranks, for $n=1,2,3$ and 10. We can consider that a word is correctly recognized at rank n if the correct word is among the n first words returned by Viterbi method (see section II.D.2). The results using the proposed bi-character model are compared to the results using only isolated characters (which is very similar to the approach proposed in [1]).

In the first experiment, when adding the bi-character model, the recognition rate at rank 1 is increased from 65.4% to 83.8%. This improvement of 18.4% of the recognition rate shows the effectiveness of the bi-character model.

In the second experiment we can observe the behavior of the proposed approach when increasing the size of the lexicon. The recognition rates are still very superior to the recognition rates without the bi-character recognizer (+22,8%); however they are decreased compared to the first experiment but do not drop drastically. They are even satisfying at ranks $n>1$.

TABLE III. PERFORMANCE EVALUATION

Lexicon	Bi-char	Top1	Top2	Top3	Top10
30	No	65.40	73.20	79.60	94.40
	Yes	83.8	90.6	92.6	98.00
100/500	No	54.00	61.00	63.00	77.00
	Yes	76.80	83.78	87.68	93.80

IV. CONCLUSION

In this paper, we have introduced a new method based on a bi-character recognizer, which is able to solve conflicts occurring in the segmentation process. Preliminary experiments show a significant improvement of the recognition rates. This model may be easily applied to many different alphabets, provided convenient features. We have been working for a few weeks on the extension of our model to open-world environments where the size of the lexicon may be huge.

REFERENCES

- [1] Ahmad, A.R, Viard-Gaudin, C., Khalid, M. Lexicon-based. Word Recognition Using Support Vector Marching and Hidden Markov Model (ICDAR2009), pp 161-165
- [2] [2] C.C. Tappert, C.Y. Suen, T. Wakahara, State of the Art in On-Line Handwriting Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, August 1990.
- [3] [3] Y. Kessentini, T. Paquet, A. Benhamadou, A Multi-stream Approach to Off-Line Handwritten Word Recognition. Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), vol. 1, pp.317-321, 2007.

- [4] [4] Y. Kessentini, T. Paquet, A. Benhamadou Multi-Stream HMM-based Approach for off-line Multi-Script Handwriting Word Recognition. In International Conference on Frontiers in Handwriting Recognition 2008.
- [5] [5] R. Saabni, J. El-San., Hierarchical On-line Arabic Handwriting Recognition, 10th International Conference on Document Analysis and Recognition (ICDAR 2009), 2009.
- [6] [6] L. Heutte, T. Paquet, J.V. Moreau, Y. Lecourtier, C. Olivier, A structural/ statistical feature based vector for handwritten character recognition, Pattern Recognition Letters, Vol 19, pp. 629–641, 1998.
- [7] [7] D.V. Jadhao, R.S. Holambe, Feature Extraction and Dimensionality Reduction Using Radon and Fourier Transforms with Application to Face Recognition, Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007) - Volume 02, Pages 254-260 , 2007.
- [8] [8] M. .Zhenjiang, Zernike moment-based image shape analysis and its application, Pattern Recognition Letters 21 (2000), pp. 169 – 177.
- [9] [9] P. Pudil,J. Novovicova, J. Kittler. Floating search methods in feature selection, Pattern Recognition Letters 1994, Vol. 15, pp 1119-1125.
- [10] [10] A.J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory. Vol. 13, issue 2, pp. 260-269.
- [11] [11] C. Viard-Gaudin, The user IRONOFF manual, IRESTE, University of Nantes. 1999.