Journal of Science and Technology Volume 48, Issue 4, 2010 pp. 1-12

BI-CHARACTER MODEL FOR ON-LINE CURSIVE HANDWRITING RECOGNITION

DE CAO TRAN

ABSTRACT

This paper deals with on-line cursive handwriting recognition. Analytic approach has got more attraction during the last ten years. It relies on a preliminary segmentation stage, which remains one of the challenges and might have a strong effect to the correct recognition rate. The segmentation aims to cut the ink strokes into a set of small pieces, called graphemes. The recognition process tries to combine them to build different segments of cursive pattern, which correspond to individual characters in the strokes. This is not a trivial process because there is no effective algorithm to decide which grapheme belongs to which character. Traditionally, the recognition process makes different assumptions about word segments which corresponding to the characters presenting in the cursive handwriting pattern. Then, the recognition process chooses the best possibility based on the probabilities of the recognition results. However, there is very little information to validate or re-evaluate that "the best possibility" is appropriate in the real world. In order to overcome this problem, this paper introduces a bi-character model, where each character is recognized jointly with its neighbor. It offers a possibility to validate a segment of word (with its neighbor) to see if it is a correct segmentation (respecting to a character). The experimental investigation on a standard dataset illustrates that the proposed model has a significant contribution to improve the recognition rate. In fact, the recognition rate is move from 65% to 83% by using the bi-character model.

Keywords. On-line cursive handwriting, Hidden Markov Model, Handwriting recognition model, Bi-character model.

1. INTRODUCTION

In the last 20 years, there has been an explosion of the number of mobile devices. The technology has allowed the development of many different kinds of acquisition devices such as PDA, electronic tablets…. These devices capture pen-tip movements as strokes that are sequences of ink points stored as a sequence of (x, y)-coordinates. Such a device pushes considerably activities on on-line handwriting recognition research. Indeed, the research on online handwriting started during the 1960s, knew a break in the 1970s [1, 2], and 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, a lot of recognizing methods use additionally the shape of the characters (off-line data), which may be reconstructed from the on-line signal, to capture a high variation of handwriting.

For cursive handwriting research, two main approaches can be figured out: global approach and analytical approach. The former processes the word shape or on-line signal as a whole pattern and tries to recognize it as a whole word. Systems relying on this approach, therefore, need to be trained with a large training set that contains all words in the lexicon and in a great amount of pattern variations. In other words, by this approach, we need a big dataset for a whole set of lexicons (the vocabulary) and its variations. In analytic approach, the word shape or on-line signal in input will be segmented into individual characters. These characters are recognized independently and are concatenated to build the whole word. Systems relying on the analytic approach need to be trained only the alphabet of the language. Moreover, such a system might be adapted to different lexicons (on the same alphabet) without re-training. Due to these major advantages, the analytical approach has got a lot of attention during the last few years [1, 3, 4]. However, the segmentation step remains a very difficult problem because of the free possible connections between characters, the large variability of the handwriting due to different scriptwriters or different contexts. No effective algorithm is well known to generate a correct segmentation for a cursive handwriting pattern. Actually, many segmentation strategies are applied with a significant error rates or confusions [5].

In this paper, the analytical approach is used. The segmentation aims to cut the ink strokes into a set of small pieces, called graphemes. The recognition process tries to combine them to rebuild different segments of cursive pattern, which correspond to characters in the word. This is not a trivial process because there is no effective algorithm to decide which grapheme belongs to which character. Traditionally, the recognition process makes different assumptions about word segments which corresponding to characters presenting in the cursive handwriting pattern. Then, the recognition process chooses the best possibility based on the probabilities of the recognition results. However, there is very little feature to validate or re-evaluate that "the best possibility" is appropriate in the real world. In order to overcome this problem, the actual work introduces a bi-character model, in which each character is recognized jointly with its neighbor. It offers a possibility to validate segment of word (with it neighbor) to see if it is a correct segment of the word (respecting to a character). The main idea of bi- character model is that when concatenating graphemes into word segments respecting to characters, these segments are processed two times:

- Each segment is supposed as a character and is passed into a handwriting character recognizer (HCR) for isolated character recognition.
- Two consecutive segments are considered as a pattern and referenced as a bi-character pattern. It is passed into a recognizer for bi-character recognition.

The first step aims to recognize the character presenting in the cursive pattern. The second step aims to re-evaluation the pattern jointly with it neighbor to make sure that it is separated correctly with it neighbor.

The remaining of paper is organized as follows: general architecture of analytic model is described in section 2; section 3 represents briefly Hidden Markov Model (HMM) for cursive online handwriting; section 4 and 5 provides detail implementation of two HCRs corresponding to two processing steps mentioned above; section 6 explains the method for cursive pattern recognition and the experimental assessment for this method are followed in section 7. Finally, some conclusion and remarks are made in the last section.

2. ANALYTIC MODEL FOR CURSIVE HANDWRITING RECOGNITION USING HMM

This section reserves for representing the model for online cursive handwriting processing and recognizing. In this paper, the analytic approach is addressed and the online data is in considered. Figure 1 illustrates an overview on the recognition system.

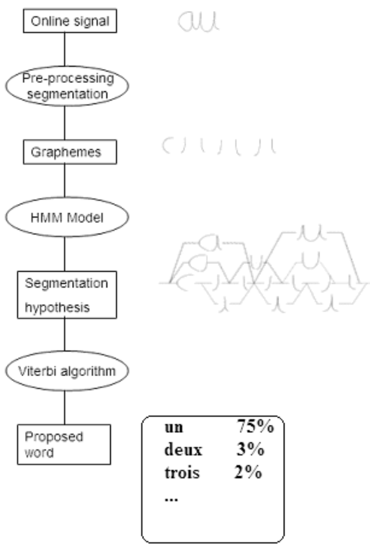


Figure 1. Overview on recognition system with analytic approach using HMM

First of all, online signal is preprocessed and segmented into graphemes. There is a lot of work that proposes different ways to realize this task, for example, using a slide window; cutting by using maximum and minimum point in y-coordinate [1]. The latter is used in this paper for segmentation. Based on y-coordinate, the sequence of points that compose the word is split into graphemes. Each grapheme is a sequence of point variable from maximum (of y-coordinate) to minimum or from minimum to maximum. Figure 2 represents the graphemes (in the second line) obtained by segmenting the word "au" (in the first line) using maximum and minimum point in y-coordinate.

Figure 2. Segmentation of word "au" using maximum and minimum point in y-coordinate

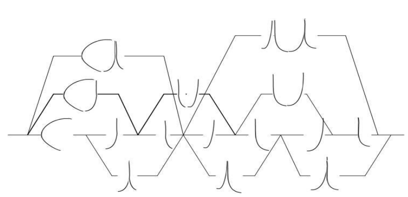


Figure 3. Some possibilities of concatenation of graphemes from "au"

Once handwriting pattern has been segmented into graphemes, an N-levels graph is constructed. It represents possible concatenation of graphemes (see Figure 3). Each node represents a possibility of concatenation of graphemes and it is assumed as a character in the handwriting pattern (i.e. cursive handwritten word). Each node is, therefore, introduced as an input to the isolated character recognizer. A proposed character along with a probability for each node is considered as an observation associated with this node in the HMM model.

The problem is how to determine a sequence of nodes to build the word corresponding to the online signal (i.e. handwriting pattern). The decision can be based on Viterbi algorithm. The algorithm determines the probability where a lexicon matches with the handwriting pattern. As a result, finding a label (a word) for a handwriting pattern can be seen as selecting the word with the highest probability proposed by Viterbi algorithm.

The whole recognition process can be described as following: the online signal (handwriting pattern) is segmented into graphemes. Then, a HMM model that represents all possibilities of concatenate them is build. Based on this model, the probability of matching between the handwritten pattern and each lexicon is calculated by using Viterbi algorithm. Lexicons are sorted by the matching probability (ranking). Finally, selected word is chosen from top n of ranking, i.e from n first elements on the top of ranking.

3. HIDDEN MARKOV MODEL FOR REPRESENTING ONLINE CURSIVE HANDWRITING

Mathematically, a Hidden Markov Model (HMM) is a state model that represents the states of system and the transitions between them. It is usually denoted a HMM model as $\gamma = \{A, B, \pi\}$ for a set of states $S = \{s_1, s_2, \ldots, s_n\}$ and a set of observations $O = \{o_1, o_2, \ldots, o_m\}$ respectively (see Figure 4):

- $A = (a_{ij})$ denotes a matrix that represents the probably of transition from state i to state j.
- $B = (b_{ij})$ denotes a matrix that represents the probability at which the observation j is appeared at the state i.
- $\pi = (\pi_i)$ denotes the initial probability, i.e. the probability of stating model at state i.

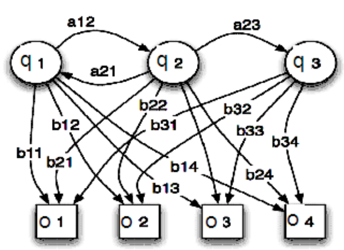


Figure 4. HMM model (image adapted from Wiki)

The N-levels graph mentioned in the section 2 can be seen as an HMM model for the cursive handwriting pattern in consideration:

- S: Set of possibilities of concatenation of graphemes. Each node in the graph is a state.

- O: Set of observations. Each observation is a character in the alphabet that supports the lexicon.
- A = (a_{ij}): The matrix of the probability of transition from state i to state j. In the context of the graph presentation in section 2, it is the probability for a character that follows a character. It can be seen as the frequency of a couple of characters in the lexicon. In our work, it is the probability in the result of recognizing of each couple of nodes in the graph, called bi-characters model. Therefore, the probability of the transition is the probability of bi-characters recognition.
- B = (b_{ij}): the probability for appearing character j associating with state i. In our system, it is the probability of the character that is recognized from the combination of graphemes at a node by using the isolated character recognition system.
- π: contains a neutral value (equal to 1), as we consider that every starting node may have the same probability.

The proposed method relies on an analytic approach and an explicit segmentation method (from maximum to minimum and from minimum to maximum). The bi-characters model helps to recognize a character jointly with its neighbor character. This approach offers a possibility to eliminate segmentation errors. The method is divided into 4 steps: pre-processing and segmentation, character recognition, bi-characters recognition, and pos-processing.

4. ISOLATED CHARACTER RECOGNITION SYSTEM

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

4.1. Feature extraction

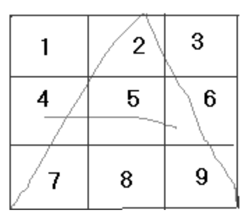


Figure 5. Basic 3 × 3 grid for extracting statistical and structural variables

In this paper, the combination of on-line and off-line data is used. The data captured by the tablet are naturally on-line. The off-line features are extracted from the artificial image which is obtained by converting on-line data into an image. First of all, seven statistical and structural feature families proposed by Heutte [6] are used, including: 7 Hu invariant moments; horizontal and vertical projections; top, bottom, left and right profiles; intersections with horizontal and vertical straight lines; holes and concave arcs; the top, bottom, left and right extremities; the end points and junctions. Some features are computed basically on each cell in a 3 × 3 grid (see Figure 5 and in the appendix)

Then we add some more features extracted from off-line data that include:

- Radon invariants [12];
- Zernike moments [13, 14].

Finally, the features extracted from on-line data such as starting points and ending points (relative position on global bounding box of on-line signals), number of strokes, direction of stroke at starting, direction of stroke at ending are added to the features set. Each character is represented by a large vector of 254 dimensions, which is a concatenation of the off-line and the on-line features mentioned above. Logically, a process for selecting relevant features is needed to eliminate redundancies.

For the selection matter, first of all, the best-first algorithm [14] is used. It does not give an optimal solution, i.e. the best set of features. However, it helps to eliminate many features that are not relevant. Then, the selected features are re-evaluated individually by Weka tools (attribute evaluator = CfsSubsetEval and search method=BestFirst) [15]. Finally, 45 features considered as the most pertinent are retained. The list of 45 features selected from the 254 variables mentioned above can be found in the appendix.

4.2. Isolated character recognition system

Our isolated character recognition system relies on the use of Support Vector Machine (SVM). The SVM classifier used in our work is the one implemented in the LibSVM software package [16]. SVM is identified as a good classifier for handwritten character recognition. A comparison of SVM to other classifiers, including Neural Networks and KNN (K-Nearest Neighbors) can be found in [1,18]. In our work, Radial Basis Function kernel given by $K(x, x') = \exp(-\gamma \|x - x'\|)^2)$ has been used. An exhaustive test with cross validation on a training set has been performed to find a good cost parameter C and the radius of the RBF kernel γ. The range of values tested were $C = 2^0,..., 2^{10}$ and $\gamma = 2^{-1},.. 2^{-10}$. We also try to find the "optimal parameters" by refining with smaller steps around the good values found from the exhaustive test above. It is observed that the best value for C is approximately 4 and the best value for γ is approximately 2^{-5}. Finally, for simplification purposes, the values chosen in our experimentation are $C = 4$ and $\gamma = 2^{-5}$.

Table 1. Results of testing on non-accent characters in comparison with those reported in [1]

HCR for	Samples for training per class	Samples for testing per class	Correct recognition rate	Recognition rate reported in [1]
Digit (0..9)	1600	400	98.7 %	98.6%
Upper case 'A'..'Z'	1600	400	95.6 %	95.1%
Lower case 'a'..'z'	1600	400	93.3 %	93.7%

First of all, the experimentation focused on testing non-accented characters. This test aims to verify the feature selection to build our HCRs. The combination of UNIPEN and IRONOFF is used to obtain a high variability of handwriting characteristics. In fact, we used all the data of IRONOFF and randomly selected a number of samples from UNIPEN to get 2000 samples for each non-accented class of characters. 1600 samples are randomly selected from these 2000 for training and the 400 remaining are used for testing. Table 1 illustrates the performance of the HCR for non-accented characters. The results show that the combination of on-line and off-line data may result in a good recognition rate. In addition, the number of features for handwritten Latin alphabet is not very high. The HCRs in this work are built on only 45 features. However, it can be compared to other HCRs built in [1] with 210 features (7 features/point × 30 points, computed on on-line signal). The last column in Table 1 refers to the results reported in [1].

5. BI-CHARACTER RECOGNITION SYSTEM

5.1 Bi-character recognition model

A great disadvantage of the analytic approach is how to concatenate graphemes and how to recognize such a combination as a character. It may lead to confusions between a piece of character and a whole character. An uncompleted part of a character in the handwriting pattern can be recognized as a character. An example of such confusion is presented in Figure 6: the graphemes analyzed from the word "au" can be reformulated and recognized as "ouui" (sequence 2 in Figure 6) or "ciii" (sequence 3 in Figure 6). In order to avoid these confusions, we introduced a bi-character model in which a character is not only recognized as an isolated character but it is also recognized in combination with the character following it. In order word, a character is always checked in associating with its neighbor to ensure that two sequences of graphemes are correctly recognized at both character and bi-character levels. In fact, if we check the bi-character corresponding to "au" in Figure 6 and the one respecting to two first characters in the sequence 2 (i.e. "ou") the combination of these graphemes is as illustrated in Figure 7.

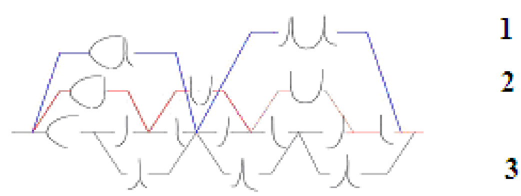


*Figure 6.*Example of misleading in isolated character recognition

a) b)

Figure 7. a) bi-character corresponding to "au" (sequence 1);
b) bi-character corresponding to two first characters in the sequence 2 ("ou")

Indeed, if we consider the sequence of two characters and check it as a bi-character we hope that the bi-character in Figure 7a would be recognized as "au" with a probability higher than the probability associated with the recognition of the bi-character in Figure 7b as "ou". It provides an alternative to validate the recognition model in the real world.

5.2 Bi-character recognition

Table 2. Performance of the recognition on 68 bi-character classes.

HCR for	Samples for training per class	Samples for testing per class	Correct recognition rate
68 Bi-characters	400	100	85.6 %

The classification of bi-character samples or bi-character recognition relies on the use of SVM. In the practice, we use the SVM that is represented in section 4 and re-training and testing on bi-character samples. 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 27200 examples (400 samples per class) of these 68 bi-characters written by different scriptwriters. Experimental results are presented in Table 2. Due to the higher number of classes to discriminate compared to the isolated character recognizer, the recognition rate on 68 classes of bi-character are lower than the one obtained on 52 isolated character classes (alphabet). The recognition rate is 85.6% versus of 93.3% obtained on 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.

6. WORD RECOGNITION SYSTEM

The general schema for recognizing a cursive word has been done in Figure 1. Once the N-level graph corresponding to all the possible grapheme concatenations has been computed (section 2) and enriched with the corresponding isolated characters and bi-characters probabilities (sections 4 and 5), it represents a Hidden Markov Model (HMM). The Viterbi algorithm is used further to decode the corresponding HMM model 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. The Viterbi algorithm is a dynamic programming algorithm which aims at finding the most likely sequence of hidden states $S = \{S_1, S_2, \ldots, S_t\}$ on the model Y with observations $O = \{O_1, O_2, \ldots, O_t\}$ [10].

When applying the Viterbi algorithm to a closed world, the observations are the lexicons. Then the Viterbi algorithm is used to find the sequence of states that yields the maximum probability. Finally, N words associated to the highest probabilities (ranked by descending probability order) are provided as the output of the system.

7. EXPERIMENTAL RESULTS ON CURSIVE WORD RECOGNITION

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 bi-characters among 68 bi-characters mentioned above.

Table 3. Performance evaluation

Lexicon size	Using bi-character	Top 1	Top 2	Top 3	Top 10
30	No	65.4	73.2	79.6	94.4
	Yes	83.8	90.6	92.6	98.0
100	No	54.0	61.0	63.0	77.0
	Yes	76.8	83.8	87.7	93.8

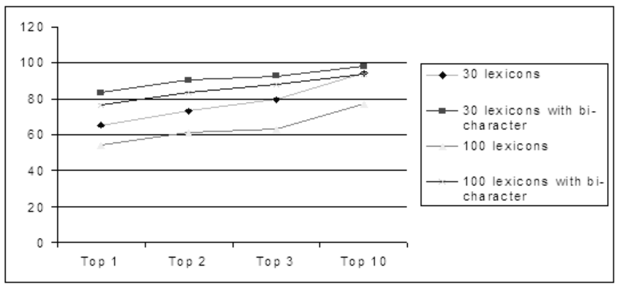


Figure 8. performance of cursive handwriting recognition test with and without bi-character model

The experimental results given in Table 3 show the recognition rates at the n first ranks, for n = 1, 2, 3 and 10 (referenced as top 1, top 2,…, top 10). A word is correctly recognized at rank n if the correct word is among the n first words on the top of ranking on probability returned from Viterbi algorithm. The results obtained using our approach (bi-character) will also be compared with the results based on isolated character. The test on isolated character has been reported 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, the proposed approach is tested with a larger number of lexicons. The recognition rates decrease a little because of the size of lexicon, but they are still higher than the recognition rates without the support of bi-character model.

8. CONCLUSION

In this paper, we have applied HMM model to the online cursive handwriting problem. The analytic approach is used: handwriting pattern is segmented into graphemes. Then, the system tries to concatenate these graphemes into word segments that correspond to characters in the handwriting pattern. It is not trivial to decide which grapheme belongs into which segment. We have introduced a new method based on a bi-character recognition, which offers a possibility to check a character along with it neighbor to make sure that the combination of graphemes is correct. It helps to reduce the confusion of recognition. Preliminary experiments show a significant improvement of the recognition rates. This method may be applied to other alphabets. The extension of the proposed model to open-world is totally logic and feasible.

REFERENCES

1. Ahmad A. R., Viard-Gaudin, C., Khalid, M. Lexicon-based - Word Recognition Using Support Vector Marching and Hidden Markov Model, ICDAR 2009, pp. 161-165.
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. 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, 2007, pp.317-321.
4. Y. Kessentini, T. Paquet - A. Benhamadou Multi-Stream HMM-based Approch for off-line Multi-Script Handwriting Word Recognition, In International Conference on Frontiers in Handwriting Recognition, 2008.
5. R. Saabni, J. El-San - Hierarchical On-line Arabic Handwriting Recognition, 10th International Conference on Document Analysis and Recognition, ICDAR 2009, 2009.
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 **19** (1998) 629-641.
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, Vol. 2, 2007, pp. 254-260.
8. M. Zhenjiang - Zernike moment-based image shape analysis and its application, Pattern Recognition Letters 21, 2000, pp. 169-177.
9. P. Pudil,J. Novovicova, J. Kittler - Floating search methods in feature selection, Pattern Recognition Letters 1994, Vol. 15, pp. 1119-1125.
10. A. J. Viterbi - Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, IEEE Transactions on Information Theory **13** (2) 260-269.
11. Christian Viard-Gaudin, Pierre Michel Lallican, Philippe Binter, Stefan Knerr - The IRESTE On/Off (IRONOFF) Dual Handwriting Database, Proceedings of the Fifth International Conference on Document Analysis and Recognition (ICDAR), 1999, p. 455.
12. Dattatray V. Jadhao, Raghunath 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, Vol. 2, 2007, pp. 254-260.

13. Miao Zhenjiang - Zernike moment-based image shape analysis and its application, Pattern Recognition Letters 21, 2000, pp. 169-177.

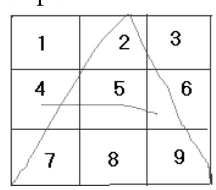
14. Pudil P., Novovicova J., Kittler J. - Floating search methods in feature selection, Pattern, Recognition Letters 1994, 15: 1119-1125.

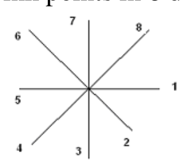
15. Weka home page http://www.cs.waikato.ac.nz/~ml/weka/

16. http://www.csie.ntu.edu.tw/~cjlin/libsvm/

17. Christopher Kermorvant, Anne-Laure Bianne, Patrick Marty, Farès Menasri - From Isolated Handwritten Characters to Fields Recognition: There's Many a Slip Twixt Cup and Lip, Proceedings of the 2009 10th International Conference on Document Analysis and Recognition, 2009, pp. 1031-1035.

APPENDIX: LIST OF 45 FEATURES FOR CHARACTER REGCONITION

Off-line features	
Off-line feature	Meaning
ANGLE_A4 ANGLE_A5 ANGLE_A7	Number of angles <180° in cell 4,5,7
F_RATE_1 F_RATE_2 F_RATE_3 F_RATE_4 F_RATE_5 F_RATE_6 F_RATE_8 F_RATE_9	% of points in each cell. 
G_X_RATE G_Y_RATE OCCLUSION	Relative coordinates of gravity centre in bounding box and the number of occlusions
POLL_L1H10 POLL_L1V5 POLL_L9H10	Number of intersections of the image (character) with the horizontal lines at the 1/10 and 9/10 of width and with the vertical line at 1/5 of the height.
PROFILE_L	Left profile
PROJECTION_H1 PROJECTION_H3 PROJECTION_H8 PROJECTION_H9 PROJECTION_V3 PROJECTION_V9	% of horizontal projection in cell 1,3,8,9 and % vertical project in cell 3 and 9 above.

RAD2 RAD3 RAD4 RAD7	Radon moments: 2,3,4 and 7
ZER0 ZER14 ZER6	Zernike moments: 0, 6 and 14.
On-line features	
On-line feature	Meaning
DS_H	% of the height of the longest up-down trace on the height of character bounding box.
END_X END_Y	Coordinates (relative position (%) in bounding box) of the ending point (the last pen-up)
LG_H	% of total length of all traces on the height of character bounding box.
LOCDIR_2 LOCDIR_3 LOCDIR_7 LOCDIR_8	Local histograms of ink points in 8 directions 
NB_STR	Number of traces (pen-down pen-up couple)
PROFILE_DIR_L	The average of cosines of peak points with cosines >0
PROFILE_DIR_R	The average of cosines of peak points with cosines <0
REB_NB	Number of peak points
START_X START_Y	Coordinates (relative position (%) in bounding box) of the starting point (the first pen-down)

Address:
Received June 16, 2010
College of Information and Communication Technology

Can Tho University, Vietnam