Combination of Contextual Information for Handwritten Word Recognition

Guillaume Koch, Thierry Paquet, Laurent Heutte
Laboratoire PSI – FRE CNRS 2645, Université de Rouen, France
{Guillaume.Koch, Thierry.Paquet, Laurent.Heutte}@univ-rouen.fr

Abstract

In this paper, we present a method for the recognition of handwritten words extracted from real incoming mail documents. The word recognition process is based on three different sources of information: outputs of a character classifier, contextual information extracted from word shapes and some a priori knowledge. Reported results demonstrate the benefit of those additional information on the word recognition rates. This approach is evaluated on a database of 5000 words examples.

1. Introduction

Today, firms are faced with the problem of processing incoming mail documents: mail reception, envelope opening, document type recognition (form, invoice, letter, ...), mail object identification (address change, complaint, termination, ...), dispatching towards the competent service and finally mail processing. Whereas part of the overall process can be fully automated (envelope opening with specific equipment, mail scanning for easy dispatching, printed form automatic reading), a large amount of handwritten documents cannot yet be automatically processed. Indeed, no system is currently able to read automatically a whole page of cursive handwriting without any a priori knowledge. This is due to the extreme complexity of the task when dealing with free layout documents, unconstrained cursive handwriting, unknown textual content of the documen. Nevertheless, it is now possible to consider restricted applications of handwritten text processing which may correspond to a real industrial need and for which the state of the art approaches can bring interesting responses. The mail topic classification from handwritten incoming mail documents is such a realistic problem.

The aim of the mail object identification task is to search for the topic of the incoming mail document. This needs therefore a robust word recognizer in order to extract a set of keywords which are representative of the mail topic. However, as we have to deal with complete mail images (examples are given in figure 1), this complex process can be divided into the two following stages:

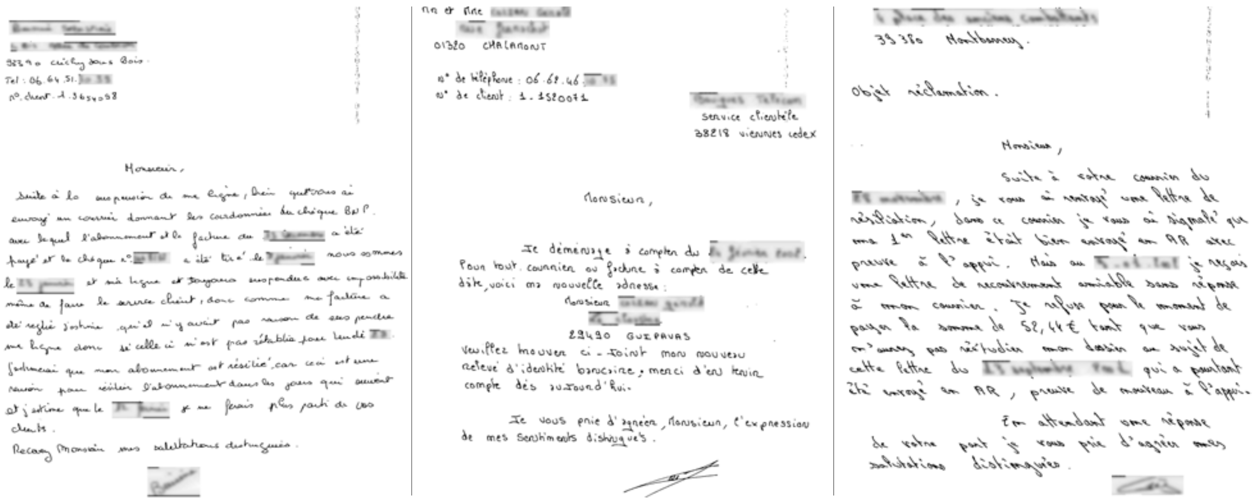
- layout analysis: as the writing style is not constrained, there is a large variability in the document layout. This implies that we have to extract word images prior to their recognition. Segmentation of documents into words is done in two steps. First, documents are segmented into lines using a method inspired from [Lik 95]. Then, each line is segmented into words using a distance based criterion [Sen 94].
- word recognition: as each mail is written by a different writer, it seems clear that the word recognition module must be omni-writer. We have therefore to reduce variability in writing style. This is done by a preprocessing step which corrects skew and slant in word images. Then, we use an explicit segmentation approach to recognize word images. This module is described in details in sections 2 and 3.

In this paper, we propose a robust handwritten cursive word recognizer which has to deal with real images obtained from the mail service of a french firm. As a consequence, we have to process poor quality images due to the main following factors:

- industrial scanner resolution: to decrease the digitization time as well as storage space, images are scanned at 240 dpi.
- online binarization: to reduce image file size, scanners give binarized images as outputs. As we have no control on this process, images or parts of images may be definitively damaged.
- free writing style: mails have been written without any *a priori* knowledge of automatic processing (no guide line, no constraint layout...). Writing styles are then heterogeneous and can vary from pure cursive to pure script (see figure 1). Furthermore, layout of the document can vary the same way for one writer to another.

The paper is organized as follows. Section 2 is devoted to the description of a first word recognizer. Some improvements are proposed in section 3 based on contextual information. Experimental results on real word images extracted from incoming mail documents are presented in section 4. Finally, conclusions and future works are drawn in section 5.

Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 **IEEE**



a. pure cursive b. pure script c. mixed cursive and script

Figure 1: Complete mail images

2. Word recognition

Recognition of cursive handwritten documents is a complex task. However, there exists nowadays industrial applications such as mail address reading, bank check processing, form reading, ... All these applications exploit the context of the analysed document and redundancy of information: for mail address reading, word recognition is directed using the zip code; bank check reading systems implement a verification stage in which the results of the legal and courtesy amounts are compared. Besides, in handwritten incoming mail documents, we have no such redundancy of information to exploit. Our application can be however constrained in terms of vocabulary size. Indeed, the entire content of a mail document is not useful for topic classification: we assume that a combination of keywords can be sufficient for the task. As a consequence, the lexicon size can be drastically decreased from about ten thousands to about a hundred words. As it is known that the complexity of recognition task increases with the lexicon size, a small lexicon ensure a reasonable and probably realistic industrial task (however not yet demonstrated).

Our word recognition system is divided into four steps (which are detailed in the following section):

• preprocessing: the aim is to reduce writing style variability such as skew of the baseline and slant of characters.

• explicit segmentation of word into letters or parts of letters (graphemes): to avoid under-segmentation which is fatal for recognition, over-segmentation is privileged. As a consequence, characters such as 'm' and 'n' may be cut into several graphemes. We then have to deal with a segmentation trellis.

• letter recognition: a feature vector is extracted and then given as input to a character classifier.

• word recognition: the segmentation trellis is explored using dynamic programming to align each word of the lexicon on the trellis. This results in a ranked list of propositions associated to a confidence.

2.1 Preprocessing

As our word recognition system must be omniwriter, we have to reduce the writing style variability which is induced by each writer. To do so, we correct two kinds of variability: skew of word image (angle of the imaginary baseline of the word image); slant of letters (average angle of the letters in the word image).

Skew correction: Skew correction is based on the estimation of an imaginary baseline used during the writing process. Estimation of this baseline is usely made by a linear regression on local *minima* (figure 2.b) of the word contour. However, these points contain *minima* of descending letters and the baseline estimation is then biased. As a consequence, those spurious points have to be filtered prior to the baseline estimation.

Our filtering method is inspired from [Ari 02]. It is based on the following assumption: an extremum coming from a descender is not aligned with those belonging to the baseline. As a consequence, the angles from horizontal it form with all other minima (angle distribution) show larger variations than the angle distribution of an extrema near the baseline. In [Ari 02], this angle distribution is clustered by C-Means algorithm into two classes: points belonging to descenders and those belonging to the baseline. However, when the word contains no descender, the algorithm still tries to find two clusters which will result in a wrong labeling of some points. To avoid this problem, spurious points are filtered as follows:

- for each point, compute the orientation of all segments containing it (figure 2.c), and sort this list.
- for each list, compute the difference between two consecutive angles. The list of differences can be evaluated using the median.
- get the highest median. Remove the corresponding point from the list if the median is greater than a threshold (experimentally determined).
- branch to first step until no points are removed.

A linear regression is performed on the remaining points to estimate the baseline skew of the word image. The skew is then corrected using a rotation (figure 2.d).

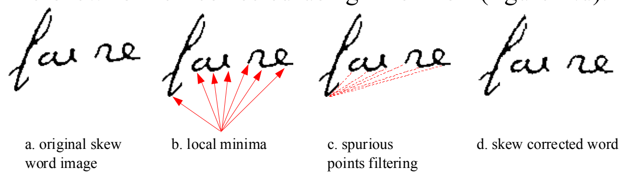


Figure 2: skew correction process

Slant correction: Word slant correction is done using a commonly used method described in detail in [Kim 94]. The chaincode using Freeman encoding is first extracted from the word image. Second step is the computation of the average slant angle α (from vertical) using the following formula: $\alpha = \tan^{-1}\left(\frac{n_1 - n_3}{n_1 + n_2 + n_3}\right)$

where n_i is the number of freeman direction "i" contained in the chaincode.

The word image is then corrected using a shearing transform in order to preserve the corrected baseline.

2.2 Segmentation into graphemes

Now that variability in the writing style is reduced, the next main stage in our word recognition system is the segmentation of the word image into graphemes. The aim of this segmentation is to disjoint all letters in the word in order to use a letter classifier. As a consequence, if a segmentation point is missed (under segmentation) (i.e. two letters are still connected after the segmentation process), it will be nearly impossible to correctly recognize the word. In order to avoid cases of under segmentation, over segmentation must be privileged, and then letters may be divided into many graphemes. So many graphemes must be grouped to recognize letters. Such graphemes are denoted "graphemes of level N", where N is the number of graphemes that have been grouped. This imposed the use of a segmentation trellis. Its exploration will be discussed later.

Segmentation of the word image into graphemes is performed using local extrema of the word contour. First the chaincode of the word is extracted and splited into upper and lower contour. Then, local mimina (respectively maxima) are extracted from the upper contour (respectively lower contour). All these points are potential segmentation points. For each point, a segmentation path is built. This path is vertical and must reach another external contour point. If an occlusion is encountered, the segmentation path is invalid and the corresponding point is removed from the list. This filtering ensures that each segmentation point splits the image in two parts at most.

Figure 3: segmentation of the french "resiliation" into graphemes

2.3 Letter recognition

In order to recognize the word image, we first need to recognize characters. The previous segmentation stage has built a segmentation trellis which contains graphemes of different levels. Each grapheme has to be submitted to a character classifier which provides character hypothesis coupled with a confidence value. The character recognition is subdivided into the two following stages:

Feature extraction: the feature set we use to describe the shape of a grapheme is inspired from [Kim 94]. The image is splited according to a 4x4 grid. For each cell, the chaincode is extracted and one builds the histogram of the 8 directions. This histogram is normalized by the chaincode length. This results in a 128-feature vector.

Letter recognition: The recognition process is performed by a multilayer perceptron (MLP) which has the following topology: 128 inputs, 200 neurons on the hidden layer, 26 outputs corresponding to the a-z letters.

At the end of this stage, the segmentation trellis is thus made up of a list of letter hypothesis associated with confidence value, for each grapheme in the trellis.

2.4 Word recognition

As letters in the word image may (or not) be segmented into several graphemes, word recognition is performed using dynamic programming. Each word contained in a specified lexicon is aligned on the trellis. The best path in this trellis is extracted and the confidence for the word is the product of the letter confidences along this path. While repeating this process for each entry in the lexicon, one obtains a ranked list of solutions.

2.5 Training

To train our recognition system, one just has to train the MLP for the letter recognition. However, its training requires a labelled letter database. To build such a realistic database, letters must be extracted from real word images, mainly because the ligature between letters may seriously transform letter shapes. If a human operator manually segments word into letters and extract them, letter samples may be far from the result of a segmentation procedure. To avoid this problem, a labelling system may segment word images and let the operator group graphems together to build letters and label them. However this is a boring and time consuming task. As it is simpler to label word images, we have developed an automatic process for letter labelling from word images (see Figure 4).

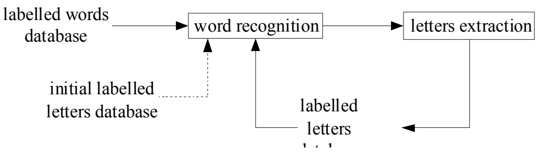


Figure 4: automatic labelled letters database creation

The word recognition process used here differs in two points with the one described earlier: (i) the lexicon used here contains only one entry: the word label; (ii) the letter recognizer has been replaced by a KNN classifier (which does not need a training stage).

Once the recognition process is done, letters are extracted along the best path in the trellis. While repeating this procedure on the entire word database, a new labelled letter database is created. This new database can be given as input to the KNN classifier and the process can be itered again.

As we are mainly interested in maximazing the word recognition rate, the letter database of each iteration is evaluated using a word test set. The one which gives the best performance is used to train the MLP, which is classically trained using back propagation [Bis 95].

3. Combination of contextual information

3.1 Using word context

Letter recognition is a complex task: there are 26 classes to discriminate (lower case letters only). Even more, some couples of letters are impossible to distinguish without context (e.g. “e” and “l”) because of the feature vector normalization ... (figure 5.a). However, these ambiguities can be solved using the word context, especially base lines (figure 5.b and 5.c). Thus, we will try to exploit this context to increase the performance of letter recognition.

We focus on one kind of contextual information: writing zones. On word images, three different writing zones can be distinguished [Gra 03] (figure 5.b and 5.c): ascender, descender and median zones. One useful feature set to characterize letters can be the ratio of the letter in the three different zones.

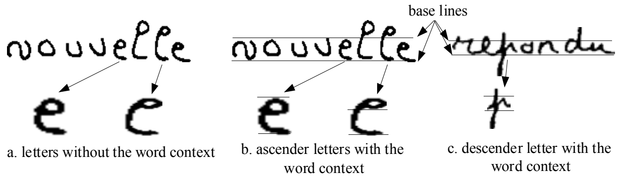


Figure 5: use of word context in letter recognition

First, writing zones must be located. As our preprocessing step normalizes the skew of word images, one can made the assumption that words are written horizontally, which simplifies the writing zone detection: orientation of border lines is known (horizontal). As a consequence, it is possible to locate these zones using the horizontal projection of black pixels: the median zone contains the largest number of pixels contained in the word image. On the contrary, ascenders and descenders contain a small proportion of black pixels. Writing zones are then detected using an experimentally estimated threshold. Figure 5 shows example of writing zones.

Four classes of letters can thus be distinguished:

- median letters: those who are usually written mainly in the median part of the word (“a c e ...”)
- ascender letters: letters containing an ascender part
- descender letters: letters containing a descender part
- ascender and descender letters: letters which are written along the three writing zones (“f”)

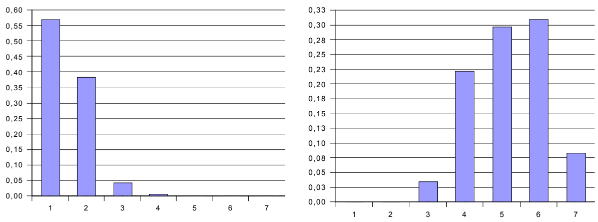
Recognition is made by a MLP classifier which has as inputs the proportion of the letter in each writing zone. This neural network has the following topology: 3 input neurons, 10 neurons on the hidden layer and 4 neurons on the output one.

The last stage aims at integrating the results of the previous classification in our word recognition process. This is described in section 3.3.

3.2 *a priori* knowledge

The grapheme segmentation method described in section 2.2 is designed to over-segment letters. For example, an “m” is usually divided into five parts. On the contrary, an “i” is rarely segmented. One has highlighted here the fact that letters are not equiprobably distributed over all the segmentation levels (see figure 6). This a priori knowledge can be used to improve the recognition.

Statistics about these distributions are collected on the letter database and then reinjected in the recognition process, as described in the next section.



a. Segmentation of letter 'c'

b. Segmentation of letter 'm'

Figure 6: Number of graphemes produce

3.3 Combination

Now that different contextual information are extracted from letters, they must be combined [Oud 03].

We introduce now some notations. Let G be the grapheme image, C a character, S the shape class of C ("m" for median, "b" for ascender, "j" for descender and "f" for ascender and descender shapes) and Sl a segmentation level.

The information we want to combine are:

- the output of the letter classifier: $P(C/G)$
- the output of the contextual classifier: $P(S/G)$
- the *a priori* knowledge about the segmentation level: $P(C/Sl)$

For each grapheme image G, the score of the letter hypothesis L $P(L/G,Sl)$ can be expressed as follows:

$$P(L/G,Sl) = \sum_{C='a'}^{'z'} \sum_{S=m,b,j,f} P(L/C,S)P(C,S/G,Sl) \quad (1)$$

In equation (1), P(L/C,S) is the confusion probability between a letter hypothesis L and letter C with the shape S. This factor can be rewritten as follows:

$$P(L/C,S) = \frac{P(L)P(C,S/L)}{P(C,S)}$$

assuming C and S are independant conditional to L

$$= \frac{P(L)P(C/L)P(S/L)}{P(C)P(S)} \quad = \frac{P(L/C)P(L/S)}{P(L)} \quad (2)$$

Using this result in equation (1), we obtain:

$$P(L/G,Sl) = \frac{1}{P(L)} \sum_{C='a'}^{'z'} \sum_{S=m,b,j,f} P(L/C)P(L/S)P(C,S/G,Sl)$$

$$= \frac{1}{P(L)} \sum_{C='a'}^{'z'} P(L/C)P(C/G,Sl) * \sum_{S=m,b,j,f} P(L/S)P(S/G,Sl)$$

Considering G and Sl independent conditional to C and to S, we obtain:

$$P(L/G,Sl) = \frac{1}{P(L)} \sum_{C='a'}^{'z'} \frac{P(L/C)P(C/G)P(C/Sl)}{P(C)}$$

$$* \sum_{S=m,b,j,f} \frac{P(L/S)P(S/G)P(S/Sl)}{P(S)}$$

Assuming that there is no confusion between L and C $(P(C/L) = 0 \text{ if } L \neq C)$, that each letter has only one shape class (see section 3.1) and that C and S are equiprobable, we obtain:

$$P(L/G,Sl) \approx P(L/G)P(L/Sl)P(S/G)P(S/Sl)$$

As one can see, the combination of our different informations can be computed by multiplying them.

Figure 7 summarizes the new recognition process.

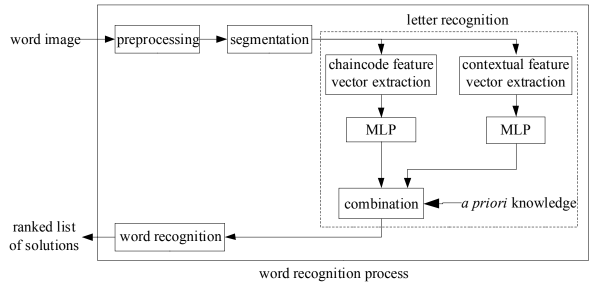


Figure 7: word recognition process using contextual information

4. Experimental results

A labelled word image database has been extracted from incoming mail documents. It contains about 5000 word images written by 500 writers. The database has been split into two parts: training set (about 4500 images from 450 writers) used for the MLPs training and test set (about 500 images from 50 unknown writers)

The lexicon generated by our labelled word database contains about 1 400 words. In order to evaluate the performance of our recognition process, different lexicon sizes must be used. To simulate a lexicon of size N from the original one, each word is recognized using a randomly generated lexicon containing the label of the word to recognize, completed by N-1 words randomly chosen from the original lexicon.

Figure 8 gives letter recognition rates with and without information combination. As expected, the word context improves letter recognition performances (about 2%). However, information about segmentation has no consequence on letter recognition rates. We will see below that these two informations are useful for the word recognition.

Table 1 presents the results obtained on training and testing sets of the word recognition process with and without contextual information using various lexicon sizes. As one can see, the use of contextual information improves significantly word recognition rates. Moreover, better improvements are observed with larger lexicon. Indeed, large lexicon disambiguation requires more information to discriminate word hypothesis since distance between two words among thousand tend to become smaller than distance between 2 words among 10.

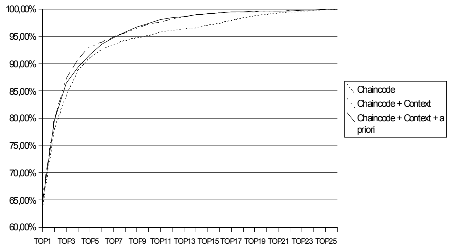


Figure 8: Letter recognition rate

The same conclusions can be drawn regarding the use of a priori knowledge in spite of the lack of improvement at the letter level. This indicates that information about segmentation is useful to reject non-letters generated by the grapheme merging.

	original classifier			with context		
lexicon size	10	100	1000	10	100	100
original classifier	92,3%	72,8%	43,6%	95,6%	82,6%	55,0
	91,8%	75,0%	43,8%	95,8%	83,0%	55,0
a priori knowledge	95,1%	83,7%	61,3%	96,6%	88,2%	68,6
	96,2%	84,4%	62,2%	96,8%	90,0%	67,8

Table 1: TOP1 recognition rates on several lexicon sizes

Another interesting result is the combination of the three sources of information: results are greatly improved. This indicates that information extracted in the three different parts are complementary.

As shown on table 2, recognition rates can be significantly improved if the first five hypotheses (TOP5) are taken into account.

lexicon size	10	100	1000
TOP1	96,80%	90,00%	67,80%
TOP2	99,40%	94,60%	81,20%
TOP5	100,00%	97,20%	90,20%

Table 2: Recognition rates of the word recognition process considering TOP1, TOP2 and TOP5

Finally, testing set contains about 500 words written by 50 different writers. As these writers are unknown from the learning set, the results demonstrate that our recognition system has done the potential of an omni-writer system. Some examples of correctly recognized words are shown figure 9.

5. Conclusions and future works

In this paper, we have described a cursive word recognition system for unconstrained handwritten incoming mail document recognition. Experimental results confirm that the use of additional information such as context and *a priori* knowledge can drastically improve word recognition rates, especially on large lexicon for which the discrimination between words is hard.

The segmentation process as it is described in section 2.2 generates a large number of graphemes in order to avoid under segmentation. However, the more the word is over-segmented, the more difficult the recognition is. In order to reduce this over-segmentation, some heuristics can be used, but their justification and their estimation is often difficult. We propose the use of a classifier in order to filter the potential segmentation points. Some contextual features based on the graphemes near the segmentation point under investigation could be used.

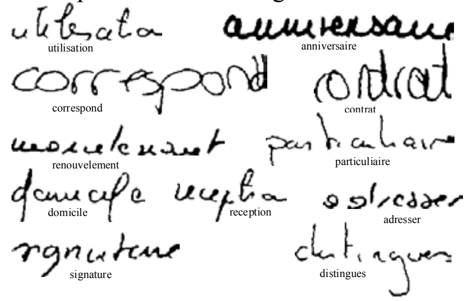


Figure 9: Examples of french words well recognized with an 100-word lexicon

Finally, segmentation of lines into words is a hard task. As for segmentation of words into graphemes, it is nearly impossible to correctly segment lines into words without a word recognition process. We will thus study a multi-hypothesis approach guided by the word recognizer.

6. References

[Ari 02] N. Arica, F.T. Yarman-Vural, “Optical character recognition for cursive handwriting”, IEEE Trans. PAMI 2002, Vol. 24, No. 6, pp. 801-813

[Bis 95] C.M Bishop, “Neural networks for pattern recognition”, Oxford, New York: Clarendon Press; Oxford University Press. xvii, 482 (1995)

[Gra 03] F. Grandidier, R. Sabourin, C. Suen, “Integration of Contextual Information in Handwriting Recognition Systems”, ICDAR (2003), pp 1252-1256

[Kim 94] F. Kimura, S. Tsuruoka, Y. Miayke, M. Shridhar, “A lexicon directed algorithm for recognition of unconstrained handwritten words”, IEICE Trans. Inf. & Syst. 1994, Vol. E77-D, No. 7, pp 785-793

[Kim 97] G. Kim, V. Govindaraju, “A lexicon driven approach to handwritten word recognition for real-time appliatons, IEEE Trans. PAMI 1997, vol. 19, no. 4, pp 366-378

[Lik 95] L. Likforman-Sulem, C. Faure, “ Extracting text lines in handwritten documents by perceptual grouping”, Advances in handwriting and drawing : a multidisciplinary approach, 1994, pp 117-135, Europia, Paris

[Oud 03] L. Oudot, “Fusion d'information et adaptation pour la reconnaissance de textes manuscrits dynamiques”, PhD thesis, Université Pierre et Marie Curie, Paris, France, 2003.

[Sen 94] G. Seni, E. Cohen, “External Word Segmentation of off-line Handwritten Text Lines”, Pattern Reognition, vol. 27, no. 1, pp 41-52, 1994