ROBUST TEXT-LINE AND WORD SEGMENTATION FOR HANDWRITTEN DOCUMENTS IMAGES[†]

Themos Stafylakis, Vassilis Papavassiliou, Vassilis Katsouros, George Carayannis
Institute for Language and Speech Processing of Athena - Research and Innovation Center in Information, Communication and Knowledge Technologies, Athens, Greece
{themosst, vpapa, vsk, gcara}@ilsp.gr

ABSTRACT

This paper addresses the problem of automatic text-line and word segmentation in handwritten document images. Two novel approaches are presented, one for each task. In text-line segmentation a Viterbi algorithm is proposed while an SVM-based metric is adopted to locate words in each text-line. The overall algorithm was tested in the ICDAR2007 handwriting segmentation contest and showed highly promising results.

Index Terms— Document image processing, handwriting recognition, image segmentation, Viterbi estimation, support vector machines

1. INTRODUCTION

Document image segmentation of text-lines and words is a critical stage towards unconstrained handwritten document recognition. The difference in the skew angle between text lines or along the same text line, the existence of touching lines and overlapping words are the most usual problems that the algorithms have to deal with.

Several methods have been proposed for these purposes. Many approaches are based on the extreme points of global projection profiles or the similarities of piece-wise projection profiles [1], [2]. Clustering algorithms based on heuristics are also used [3]. Most of these use the distances between adjacent connected components (CCs) and estimate an optimal threshold for classification [4]. Other techniques use statistics (i.e. Gaussian densities) to model lines [5]. The majority of the proposed systems were originally designed for specific tasks such as handwritten postal envelopes, historical documents, bank checks, etc. Thus, they do not generalize well to variations encountered in other handwritten documents. For example, some of these fail to detect short lines and lines that do not begin from the left of the document, or split CCs running into more than one line or word.

An algorithm for text-line extraction from handwritten document images is proposed in this paper (Section 2). The proposed method is able to handle skewed documents as well as documents with lines running into each other. Furthermore, a novel technique for locating words in text-lines is also presented in Section 3. The proposed method uses the assumption that every component belongs to one word and is flexible to capture the variation of slant.

2. TEXT-LINE SEGMENTATION

The main idea behind the proposed algorithm is the utilization of two properties that govern the main body of the document: a) the vertical quasi-periodicity of the text-lines and b) the relatively low variability of the ink width, character size and within/between word gaps that the writer uses as well as the rather uniform illumination in a given document. To incorporate this prior knowledge in our modeling, we form a Viterbi algorithm, where the two states correspond to text and gap areas. The emission probabilities of the two states are modeled by the foreground pixel densities, while the transition probabilities are given by the average height each state has. All the above measures are estimated from each document, i.e., no training is used to form the statistics.

The majority of real-world handwritten documents, though, have skew with high (inter- and intra- document) variability as well as non-strict left and right margins. In order to deal with these issues, the document image is segmented into a sequence of N consecutive non-overlapping vertical zones. The number of zones is a trade-off issue: narrow-enough so that the influence of skew in the zone can be neglected, wide-enough so that the sufficient statistics can be estimated robustly. For our experiments, we use a global N equal to 20, which produced the best results on the training set.

For each zone, we calculate the foreground pixel density. The zones having density above a certain threshold are classified as text zones while the remaining as margins. The margin zones will be split to text lines by simple extension of the separators estimated from the near most text zones. The threshold is set to half of the median value of all densities.

[†] This work is partly funded by the Greek Secretariat for Research and Technology under the program PENED-03/251.

In order to locate the upper and lower bounds of the text lines we work as follows. Considering only the text zones, we calculate the horizontal projection profiles and smooth them with the neighboring ones (using second-order non-causal FIR filter). Each smoothed profile is differentiated and the upper and lower bounds of the text lines are considered as the maxima and minima of the differential profiles and force them to be interleaved. The first extreme point is also forced to be a maximum. This step should be considered as an initial over-segmentation procedure. What we obtain is the localization of the boundaries of the candidate text lines. Due to this over-segmentation step, an estimation of the statistics needed to feed the Viterbi algorithm can be reached. We form the two states, i.e., the gap (denoted by c_0) and text-line (denoted by c_1) states as the areas between consecutive minima-maxima and maxima-minima respectively. To obtain robust statistics, we do not take into account the 30% of the areas of each state having lower heights, thereby avoiding the effect of over-segmentation. The results showed that the accuracy is rather insensitive to this parameter, as long as the over-segmentation degree is low. Thus, the statistics are estimated from the remaining 70% of each state as follows.

The prior probabilities are 1 for c_1, given that the first point is maximum. The emission probability for each state is modeled with a log-normal pdf, i.e., we transform the density values into the log-domain and calculate the class-conditional means and variances $\left\{\left(\mu_j, \sigma_j^2\right)\right\}, j = \{0,1\}$. Therefore, the emission probabilities are defined by

$$p\left(x_i \middle| s_{[h,h+H_i]} = c_j\right) \sim N\left(\mu_j, \sigma_j^2\right), \tag{1}$$

where the random variable x_i denotes foreground pixel log-density of the ith area.

The transition probabilities are modeled using an exponential distribution, with mean value $m_j, j \in \{0,1\}$, estimated from the median height of each class as follows:

$$p\left(s_{[h,h+H_i]} = c_1 \middle| s_{[h-H_{i-1},h]} = c_1\right) = \exp\left(-\frac{H_i}{m_1}\right), \tag{2}$$

$$p\left(s_{[h,h+H_i]} = c_0 \middle| s_{[h-H_{i-1},h]} = c_0\right) = \exp\left(-\frac{H_i}{m_0}\right), \tag{3}$$

$$p\left(s_{[h,h+H_i]} = c_1 \middle| s_{[h-H_{i-1},h]} = c_0\right) = 1 - \exp\left(-\frac{H_i}{m_0}\right), \tag{4}$$

$$p\left(s_{[h,h+H_i]} = c_0 \middle| s_{[h-H_{i-1},h]} = c_1\right) = 1 - \exp\left(-\frac{H_i}{m_1}\right). \tag{5}$$

The variable H_i denotes the height of the ith area, and $s_{[h,h+H]}$ the state at range between h and $h+H$. Note that the use of an exponential distribution for the transition probabilities, apart from being a well-established framework for modeling such processes, has the benefit of the so-called memoryless property: $p\left(x > h_1 + h_2 \middle| x > h_1\right) = p\left(x > h_2\right)$. Therefore, the Markov property is satisfied. The optimal state path can be reached by applying the Viterbi algorithm in each zone.

In order to merge the zone-based segmentation results and obtain the overall document's segmentation we work as follows.

Starting from left to right, each line separator of a zone is merged with the closest line separator of the following zone, while the remaining separators either define new lines or extend existing ones (see Fig. 1).

Finally, CCs that belong to two consecutive lines are split at the pixel of the skeleton having 4-neighbors and lies nearest to the separator (see Fig. 2).

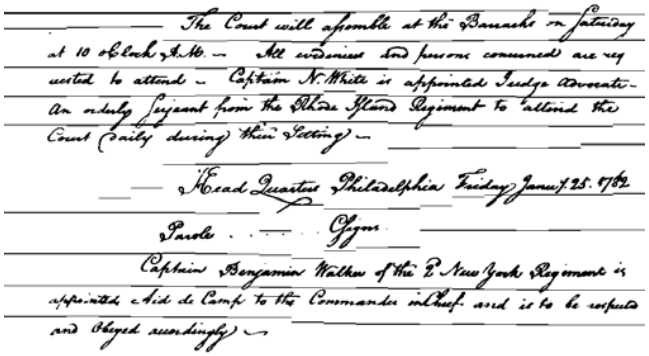

Figure 1: The resulting text-line segmentation for a typical handwritten document.

Figure 2: An example of splitting a connected component

3. WORD SEGMENTATION

Since in our approach, text-line and word segmentation are decoupled procedures, we consider here a document already segmented in text lines. We further make use of a common assumption, that given a text line, each CC belongs to only one word (i.e. successive words are not connected to each other).

The word segmentation task can be seen as a clustering problem, where the two classes are defined as within and between words gaps, denoted by C_w and C_b, respectively. Let us use the notation

$$h = \begin{cases} -1, & \text{if } \boldsymbol{z} \in C_w \\ +1, & \text{if } \boldsymbol{z} \in C_b \end{cases}, \tag{6}$$

where the element $\boldsymbol{z} = (\boldsymbol{X}, \boldsymbol{Y})$ is defined as a pair of two foreground pixel groups $\{\boldsymbol{x}_i \in \boldsymbol{X} \mid y_i = -1\}$ and

$\{\boldsymbol{x}_i \in \boldsymbol{X} \mid y_i = +1\}$. The variable $\boldsymbol{x} \in \mathbb{R}^2$ stands for the foreground pixel coordinate vector (2-dimensional) while the variable $y \in \{\pm 1\}$ defines the group that the pixel belongs to. As we will show shortly, y_i is completely determined by the (relative to the candidate gap) position of the CC the *i*th pixel belongs to. Making use of the aforementioned assumption, the group of pixels $\{\boldsymbol{x}_i \in \boldsymbol{X} \mid y_i = \pm 1\}$ is defined as a group of CCs. What we need is a function $f_t(\boldsymbol{z}) \to \{\pm 1\}$ that maps the element $\boldsymbol{z} = (\boldsymbol{X}, \boldsymbol{Y})$ to the desired class.

We first sort (in ascending order) the CCs per text line, by using the horizontal coordinate of their centroid and create the index $k = 1, \ldots, K_l$, where K_l is the total number of CCs in the *l*th text line.

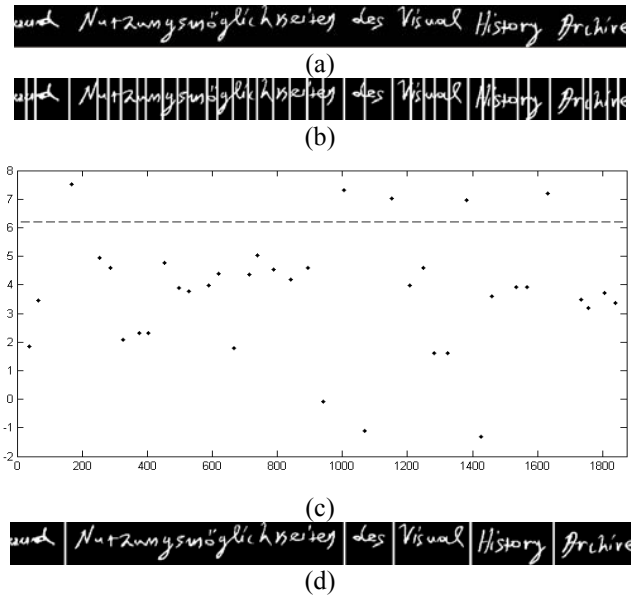

Figure 3: Word segmentation results for a given text line. (a) The original text-line image (b) Over-segmentation (c) SVM-metric for each candidate gap and the estimated threshold (d) Final segmentation.

The distance metric we use is derived directly from Support Vector Machines (SVM) theory. Given two linearly separable classes and a labeled set of *N*-dimensional patterns, among all hyperplanes (**W**,*b*) separating the data, there exists a unique one yielding the maximum margin of separation between the classes. The introduction of the optimal hyperplane (a line in our 2-dimensional problem) eliminates the need of slant correction that most of the horizontal-projection based methods require.

The optimization problem is solved using the Lagrangian dual, i.e. $\max_{\boldsymbol{a} \geq 0} \min_{\mathbf{W}, b} L(\boldsymbol{w}, b, \boldsymbol{a})$, where:

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_{i=1}^{m} a_i \left\{ y_i \cdot \left[(\mathbf{x}_i \cdot \boldsymbol{w}) + b \right] - 1 \right\}. \quad (7)$$

The variables $\mathbf{x}_i$, $i = 1, \ldots, m$ (feature space variables) are derived indirectly from the corresponding $\boldsymbol{x}_i$ (primary space variables) by the choice of the kernel function, $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$.

The above implementation makes use of hard-margin SVM, lying on the linear separability of the set. This assumption should be removed since two groups of consecutive CCs may not be linearly separable. Thus, we adopt a soft-margin SVM implementation by introducing the slack variables $\xi_i \geq 0$, $i = 1, \ldots m$. The objective function for the soft-margin case is given by

$$L^s(\boldsymbol{w}, b, \boldsymbol{a}, \boldsymbol{\xi}) = \frac{1}{2}\|w\|_2^2 + \sum_{i=1}^{m} C\xi_i - \sum_{i=1}^{m} a_i \left\{ y_i \cdot \left[(\mathbf{x}_i \cdot \boldsymbol{w}) + b \right] - 1 + \xi_i \right\}. \quad (8)$$

The non-negative constant *C* controls the amount of regularization incorporated into the model by the use of slack variables. For a complete tutorial on SVM theory we refer to [6].

Returning to our problem, what we actually need from the above analysis is not the optimal hyperplane[1] (***w***,*b*) but the corresponding margin (i.e. the negative objective function). We define the distance function $g(\boldsymbol{z})$ as follows:

$$g(\boldsymbol{z}) = -\log\left[\max_{\boldsymbol{a} \geq 0} \min_{\mathbf{W}, b} L^s(\boldsymbol{w}, b, \boldsymbol{a}, \boldsymbol{\xi}) \Big|_{\boldsymbol{z}} \right]. \quad (9)$$

In order to define completely the classification function $f_t(\boldsymbol{z}) = \text{sign}\left[g(\boldsymbol{z}) - t \right]$, we need to calculate the threshold *t*. Due to the high variability of writing styles, sizes, etc. across documents, a global threshold would be an inadequate solution. Thus, we estimate *t* from the values of $g(\boldsymbol{z})$ for the candidate gaps found in the whole document. To do so, we first evaluate all the candidate gap distances (margins) and then estimate the underlying pdf using non-parametric kernel smoothing methods. The threshold is set to be equal to the higher (rightmost) significant minimum found on the estimated pdf.

Thus, the overall word segmentation algorithm can be described in the following steps:

1. For each line $l = 1, \ldots, L$, sort the CCs by their centroid and form the set $\boldsymbol{X}^l$ as the matrix of the coordinates of all foreground pixels in the text line.
2. For each CC $k = 1, \ldots, K_l - 1$, set the corresponding labels $\boldsymbol{Y}_k^l$ to +1 for all the pixels belonging to CC having index greater than *k* and –1 for the others. Calculate $d_k^l = g(\boldsymbol{z}_k^l)$ and store them.
3. Estimate the pdf of *d* to determine the optimal threshold

[1] The optimal hyperplanes of the overall gaps may be used as a feature for measuring the skew of text lines.

t from the whole set of distances in the document, i.e., $d_k^l,\ l=1,\ldots,L,\ k=1,\ldots,K_l$.

4. The candidate gaps are labeled by $h_k^l = f_t\left(\boldsymbol{z}_k^l\right)$.

The main steps of the algorithm are shown in Fig. 3. In order to decrease the computational cost, we add two rules. We first reduce the pool of candidate gaps K_l per line to only the significant minima of the horizontal projections of the text line. Finally, we compress the set $\boldsymbol{z}_k^l = \left(\boldsymbol{X}^l, \boldsymbol{Y}_k^l\right)$ per l and k, by keeping only the critical foreground pixels that affect the distance metric. We achieve this by splitting each text line into 12 equally spaced non-overlapping horizontal zones and keeping only the 20 pixels for each zone and group that are closest to the candidate gap k. This procedure trims down the set $\boldsymbol{z}_k^l$ to only 240 points per group at maximum, increasing the tractability of the SVM optimization task without altering the results.

We finally note that in our implementation we used a simple linear kernel function, which slightly outperformed Radial Basis Functions (RBF) kernel [6]. That is the dot-product acts in the primary space.

4. EVALUATION

In order to test the two algorithms (text-line and word segmentation) we participated in the ICDAR2007 Handwriting Segmentation Contest with the shortcut ILSP-LWSeg. A training dataset consisting of 20 document images and associated ground truth, along with the corresponding evaluation software were given to us by the organizers. Then we submitted two executables (one for each method) in the form of a Win32 console application for evaluation on the test dataset, which consists of 80 document images.

The performance evaluation is presented in [7] and is based on [8]. If N is the count of ground-truth elements, M is the count of results elements and w_1, w_2, w_3, w_4, w_5, w_6 are the predetermined weights (set to 1, 0.25, 0.25, 1, 0.25, 0.25 respectively), the detection rate (DR) and the recognition accuracy (RA) are estimated as follows:

$$DR = w_1 \frac{\text{o2o}}{N} + w_2 \frac{g\_\text{o2m}}{N} + w_3 \frac{g\_\text{m2o}}{N} \tag{10}$$

$$RA = w_4 \frac{\text{o2o}}{M} + w_5 \frac{d\_\text{o2m}}{M} + w_6 \frac{d\_\text{m2o}}{M} \tag{11}$$

where o2o (one to one match), g_o2m (one ground truth to many detected), g_m2o (many ground truth to one detected), d_o2m (one detected to many ground truth), and d_m2o (many detected to one ground truth).

As the organizers of the contest reported, the documents used in order to build both datasets came from (i) several writers that were asked to copy a given text, (ii) historical handwritten archives, and (iii) scanned handwritten document samples selected from the web. None of the documents included any non-text elements and were all written in several languages including English, French, German, and Greek.

The evaluation results for line and word detection are shown in "Table 1." On 80 documents containing a total of 1771 lines and words, 1713 lines and 11732 words were segmented correctly.

	TEXT LINES	**WORDS**
N	**1771**	**13312**
M	**1773**	**13027**
O2O	**1713**	**11732**
***G*_O2M**	**5**	**303**
***G*_M2O**	**34**	**834**
***D*_O2M**	**17**	**378**
***D*_M2O**	**10**	**819**
DR	**97.3%**	**90.3%**
RA	**97.0%**	**92.4%**
FM	**97.1%**	**91.3%**
SM	**94.2%**	

Table 1.: Evaluation results

The proposed algorithm outperformed the other participants' algorithms, as well as state-of-the-art techniques as RLSA (SM=60.1%) and Projection Profiles (SM=61.6%) in both tasks [7].

5. REFERENCES

[1] C. Weliwitage, A. L. Harvey, and A. B. Jennings, "Handwritten document offline text line segmentation," in *Proc. Digital Image Computing: Techniques and Applications*, *DICTA*, pp. 184-187, December 2005.

[2] B Yanikoglu, and P.A. Sandon, "Segmentation of off-line cursive handwriting using linear programming," *Pattern Recognition, Vol. 31*, pp. 1825-1833, 1998.

[3] S. N. Srihari, B. Zhang, and C. Tomai, S. Lee, Y.C. Shin, "A system for handwriting matching and recognition," in *Proc. Symp. Document Image Understanding Technology*, Greenbelt, MD, pp. 67-75, April 2003.

[4] G. Seni and E. Cohen, "External word segmentation of off-line handwritten text lines," *Pattern Recognition*, *Vol. 27*, pp. 41-52, 1994.

[5] M. Arivazhagan, H. Srinivasan, and S. Srihari, "A Statistical approach to line segmentation in handwritten documents," in *Proc. SPIE,* pp. 6500T-1-11, 2007.

[6] V. Vapnik. "The Nature of Statistical Learning Theory," New York, *Springer Verlag*, 1995.

[7] B. Gatos, A. Antonacopoulos, and N. Stamatopoulos, "ICDAR2007 handwriting segmentation contest," in *Proc. International Conference on Document Analysis and Recognition*, pp. 1284-1288, September 2007.

[8] B.A. Yanikoglu and L. Vincent, "Pink Panther: a complete environment for ground-truthing and benchmarking document page segmentation," *Pattern Recognition, Vol. 31*, *n. 9*, pp. 1191-1204, 1994.