Pattern Recognition 43 (2010) 369–377

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Handwritten document image segmentation into text lines and words

Vassilis Papavassiliou[a,b,*], Themos Stafylakis[a,b], Vassilis Katsouros[a], George Carayannis[a,b]

[a] Institute for Language and Speech Processing of R.C. "Athena" Artemidos 6 & Epidavrou, GR-151 25 Maroussi, Greece
[b] National Technical University of Athens, School of Electrical and Computer Engineers, 9, Iroon Polytechniou str, GR 157 80 Athens, Greece

ARTICLE INFO

Article history:
Received 22 July 2008
Received in revised form 23 February 2009
Accepted 14 May 2009

Keywords:
Handwritten text line segmentation
Handwritten word segmentation
Document image processing
Viterbi estimation
Support vector machines

ABSTRACT

Two novel approaches to extract text lines and words from handwritten document are presented. The line segmentation algorithm is based on locating the optimal succession of text and gap areas within vertical zones by applying Viterbi algorithm. Then, a text-line separator drawing technique is applied and finally the connected components are assigned to text lines. Word segmentation is based on a gap metric that exploits the objective function of a soft-margin linear SVM that separates successive connected components. The algorithms tested on the benchmarking datasets of ICDAR07 handwriting segmentation contest and outperformed the participating algorithms.

1. Introduction

Document image segmentation to text lines and words is a critical stage towards unconstrained handwritten document recognition. Variation of the skew angle between text lines or along the same text line, existence of overlapping or touching lines, variable character size and non-Manhattan layout are the challenges of text line extraction. Due to high variability of writing styles, scripts, etc., methods that do not use any prior knowledge and adapt to the properties of the document image, as the proposed, would be more robust. Line extraction techniques may be categorized as projection based, grouping, smearing and Hough-based [1].

Global projections based approaches are very effective for machine printed documents but cannot handle text lines with different skew angles. However, they can be applied for skew correction in documents with constant skew angle [2]. Hough-based methods handle documents with variation in the skew angle between text lines, but are not very effective when the skew of a text line varies along its width [3]. Thus, we adopt piece-wise projections which can deal with both types of skew angle variation [4,5].

On the other hand, piece-wise projections are sensitive to characters' size variation within text lines and significant gaps between successive words. These occurrences influence the effectiveness of smearing methods too [6]. In such cases, the results of two adjacent zones may be ambiguous, affecting the drawing of text-line separators along the document width. To deal with these problems we introduce a smooth version of the projection profiles to oversegment each zone into candidate text and gap regions. Then, we reclassify these regions by applying an HMM formulation that enhances statistics from the whole document page. Starting from left and moving to the right we combine separators of consecutive zones considering their proximity and the local foreground density.

Grouping approaches can handle complex layouts, but they fail to distinguish touching text lines [7]. In our approach, we deal with such a case by splitting the respective connected component (CC) and assign the individual parts to the corresponding text lines.

In word segmentation, most of the proposed techniques consider a spatial measure of the gap between successive CCs and define a threshold to classify "within" and "between" word gaps [8]. These measures are sensitive to CCs' shape, e.g. a simple extension of the horizontal part of character "*t*". We introduce a novel gap measure which is more tolerant to such cases. The proposed measure results from the optimal value of the objective function of a soft-margin linear SVM that separates consecutive CCs.

Preliminary versions of the text-line and word segmentation algorithms were submitted to the Handwriting Segmentation Contest in ICDAR07, under the name ILSP-LWSeg, and performed the best results [9]. A short description of the participating algorithms was published in our conference paper [10]. The major steps of the proposed algorithms are illustrated in Fig. 1.

The organization of the rest of the paper is as follows: In Section 2, we refer to recent related work. In Section 3, we describe in detail the algorithm for text-line extraction from handwritten document

* Corresponding author at: Institute for Language and Speech Processing of R.C. "Athena" Artemidos 6 & Epidavrou, GR-151 25 Maroussi, Greece.
Tel.: +30 210 6875332; fax: +30 210 6854270.
E-mail addresses: vpapa@ilsp.gr (V. Papavassiliou), themosst@ilsp.gr (T. Stafylakis), vsk@ilsp.gr (V. Katsouros), gcara@ilsp.gr (G. Carayannis).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.05.007

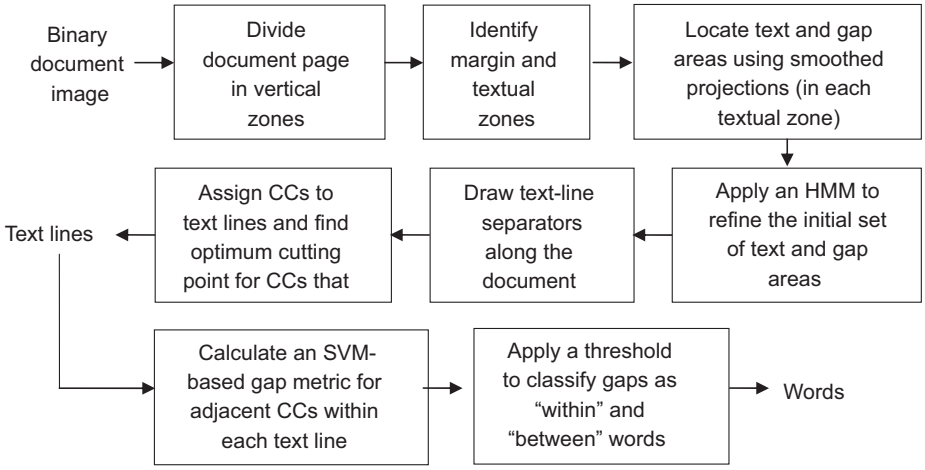

Fig. 1. Block diagram of proposed algorithms.

images. The proposed method of segmenting text lines into words is presented in Section 4. Experimental results and conclusions are discussed in Sections 5 and 6, respectively.

2. Related work

In this section, we give a brief review of recent work on text line and word segmentation in handwritten document images. As far as we know, the following techniques either achieved the best results in the corresponding test datasets, or are elements of integrated systems for specific tasks.

One of the most accurate methods uses piece-wise projection profiles to obtain an initial set of candidate lines and bivariate Gaussian densities to assign overlapping CCs into text lines [5]. Experimental results on a collection of 720 documents (English, Arabic and children's handwriting) show that 97.31% of text lines were segmented correctly. The writers mention that "a more intelligent approach to cut an overlapping component is the goal of future work".

Li et al. [11] discuss the text-line detection task as an image segmentation problem. They use a Gaussian window to convert a binary image into a smooth gray-scale. Then they adopt the level set method to evolve text-line boundaries and finally, geometrical constrains are imposed to group CCs or segments as text lines. They report pixel-level hit rates varying from 92% to 98% on different scripts and mention that "the major failures happen because two neighbouring text lines touch each other significantly".

A recent approach [3] uses block-based Hough transform to detect lines and merging methods to correct false alarms. Although the algorithm achieves a 93.1% detection rate and a 96% recognition rate, it is not flexible to follow variation of skew angle along the same text line and not very precise in the assignment of accents to text lines.

There are also systems designed for specific tasks such as handwritten postal envelopes which use clustering algorithms based on heuristics [12]. These methods do not generalize well to variations encountered in handwritten documents.

A much more challenging task is line segmentation in historical documents due to a great deal of noise. Feldbach and Tonnies [13] have proposed a bottom up method for historical church documents that requires parameters to be set according to the type of handwriting. They report a 90% correct segmentation rate for constant parameter values which rises to 97% for adjusted ones. Another integrated system for such documents [6] creates a foreground/background transition count map to find probable locations of text lines and applies min-cut/max-flow algorithm to separate initially connected text lines .The method performs high accuracy (over 98%) in 20 images of George Washington's manuscript.

Most word segmentation approaches consider text-line images to extract words. The main assumptions are that each CC belongs to only one word and gaps between words are greater than gaps between characters. A typical gap metric algorithm [14] first labels the CCs and computes their convex hulls. Then, the Euclidean distances between the convex hulls are calculated and sorted. At last, a threshold for each text line is computed and is used for the classification of gaps to "inter" or "intra" words.

A similar method [8] evaluates eight different spatial measures between pairs of CCs to locate words in handwritten postal addresses. The best metric proved to be the one which combines the result of the minimum run-length method and the vertical overlapping of two successive CCs. Additionally, this metric is adjusted by utilizing the results of a punctuation detection algorithm (periods and commas). Then, a suitable threshold is computed by an iterative procedure. The algorithm tested on 1000 address images and performed an error rate of about 10%.

Manmatha and Rothfeder [15] propose an effective for noisy historical documents scale space approach. The line image is filtered with an anisotropic Laplacian at several scales in order to produce blobs which correspond to portions of characters at small scales and to words at larger scales. The optimum scale is estimated by three different techniques (line height, page averaging and free search) from which the line height showed best results.

3. Text-line segmentation

3.1. Initial set of text and gap areas in vertical zones

In our approach, we segment the document image into non-overlapping equi-width vertical zones, as in [4,5]. The width of the zones has to be narrow enough so that the influence of skew to be neglected, and wide enough to include adequate amount of text. A zone width equal to 5% of the document image width seems to satisfy these requirements. Further discussion on the appropriate value of the zone width is carried out in Section 5.

Some vertical zones, mainly those that are close to the left and right edges of the document page, the so called "margin" zones, will not contain sufficient amount of text. We therefore disregard them and consider only the zones with a proportion of foreground pixels above a threshold (th), say, half of the median value of the foreground

pixel density of all vertical zones. By using the following equation

$$\delta_i = [1 + \mathrm{sgn}(d_i - th)]/2, \quad i \in \{1, 2, \ldots, N\}, \tag{1}$$

where d_i denotes the foreground pixel density of i-th zone and N is the number of zones, we label each zone as margin ($\delta_i = 0$) or textual ($\delta_i = 1$).

In the case where the writing style results in large gaps between successive words, a vertical zone may not contain enough foreground pixels for every text line (see third zone in Fig. 3a). In order to abate the influence of these occurrences on the projection profile PR_i, we introduce the smoothed projection profile SPR_i as a normalized weighted sum of the profiles of the M on either side neighbouring zones, by

$$SPR_i = \sum_{j=-M}^{M} \delta_{i+j} \cdot w_j \cdot PR_{i+j}. \tag{2}$$

The weights

$$w_i = \exp(-3|i|/(M+1)) \Big/ \sum_{k=-M}^{M} \exp(-3|k|/(M+1)), \quad i \in \{-M, \ldots, M\} \tag{3}$$

are defined to decay exponentially with respect to the distance from the current zone. One may observe the gain of this transformation by comparing the red with the blue profile at the regions of the capital "I" in Fig. 3b.

Subsequently, we produce a smooth estimate of the first derivate of each projection (Fig. 3c) using the symmetric difference equation

$$\Delta SPR_i(j) = \frac{1}{h(h+1)} \sum_{k=1}^{h} k \cdot (SPR_i(j+k) - SPR_i(j-k)), \tag{4}$$

where h is set to the integer value of the half of the closest odd number to the mean height of all CCs. We assign local maxima of ΔSPR_i as upper bounds of text regions and their following minima as lower bounds (Fig. 3d in red). Similarly, gap regions are identified as the areas between consecutive minima and maxima of ΔSPR_i (Fig. 3d in green).

3.2. Refinement of initial text-line separators

Having specified the succession of text and gap regions for each zone text-line separators are drawn in the middle of each gap region (Fig. 3d). However, the resulting line separators in each zone include two types of errors; superfluous separators and separators that cut across descenders or ascenders. In other words, some areas have been misclassified as text or gap due to local extrema introduced in the smoothed derivative. In order to locate more accurately the main body of each text line in each zone, we formulate an HMM for the text and gap stripes with parameters drawn from statistics of the initial set of text and gap areas within the document image. We apply a Viterbi decoding scheme on each zone to obtain a better succession scheme (Fig. 3e).

Let us define an HMM with two states; c_0 denotes a text and c_1 a gap region. The initial probabilities are set to be equal, i.e. $\pi_0 = \pi_1 = 0.5$. The transition probabilities are modeled by an exponential distribution with parameter m_j, $j \in \{0, 1\}$, the mean height of each region for the whole document image, as follows

$$a_{ij}(i) = P(s_{[h,h+H_i]} = c_j | s_{[h-H_{i-1},h]} = c_j) = \exp(-H_i/m_j), \quad j \in \{0, 1\}, \tag{5}$$

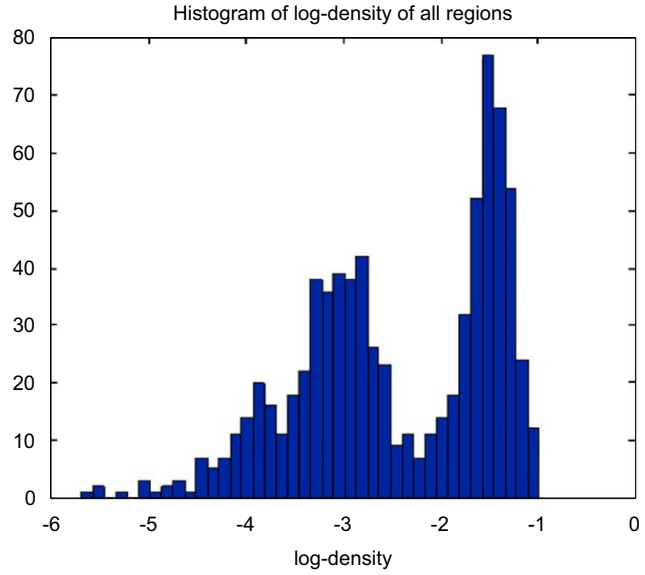

Fig. 2. Histogram of foreground pixel log–density of all regions.

where H_i denotes the height of the i-th area and $s_{[h,h+H_i]}$ the state of the region that starts at height h and extends up to $h + H_i$. The transition probabilities $a_{01}(i)$ and $a_{10}(i)$ result as the difference of $a_{00}(i)$ and $a_{11}(i)$ from 1, respectively. The use of an exponential distribution for the transition probabilities has the benefit of the so-called memory less property, i.e. depends only on the height of the current region. In addition, state transition occurs with high probability for regions with height comparable to the mean height of the previous state.

For the emission probabilities, we transform the density values of all regions into the log-domain. The histogram in Fig. 2 shows the log–normal distribution of the foreground pixels' density for all regions. One can observe that the two states may be distinguished by considering two log–normal conditional distributions. Thus, the emission probabilities are modeled with a log–normal probability density function as follows

$$p(x_i | s_{[h,h+H_i]} = c_j) \sim N(\mu_j, \sigma_j^2), \quad j \in \{0, 1\}, \tag{6}$$

where μ_j and σ_j^2 denote the mean and the variance of state j, the random variable x_i denotes the foreground pixel log–density of the i-th region bounded by h and $h + H_i$ and c_j denotes the respective state.

Regions with non zero densities and reasonable heights, e.g. over than a fifth of the mean height of all CCs, are used for estimating the parameters of the HMM. Having specified the parameters of the HMM, a vertical zone can be seen as a sequence of observations $O = o_1, o_2, \ldots, o_M$, where o_j, $j = 1, \ldots, M$ are the log–densities of regions. The corresponding state sequence $Q = q_1, q_2, \ldots, q_M$ that characterizes the respective regions as text or gap, results with the application of the Viterbi algorithm for the maximization of the probability to obtain the observation sequence O, given the model λ [16] (Fig. 3d). As mentioned above, text-line separators are drawn in the middle of gap regions (Fig. 3f).

3.3. Text-line separators drawing algorithm

So far we have estimated the text-line separators in each vertical zone. Let $\{\psi_i^j, j = 1, \ldots, S_i\}$ denote an increasing sequence of height locations for the separators of the i-th zone, with S_i be the total number of separators in this zone. The drawing algorithm of text line

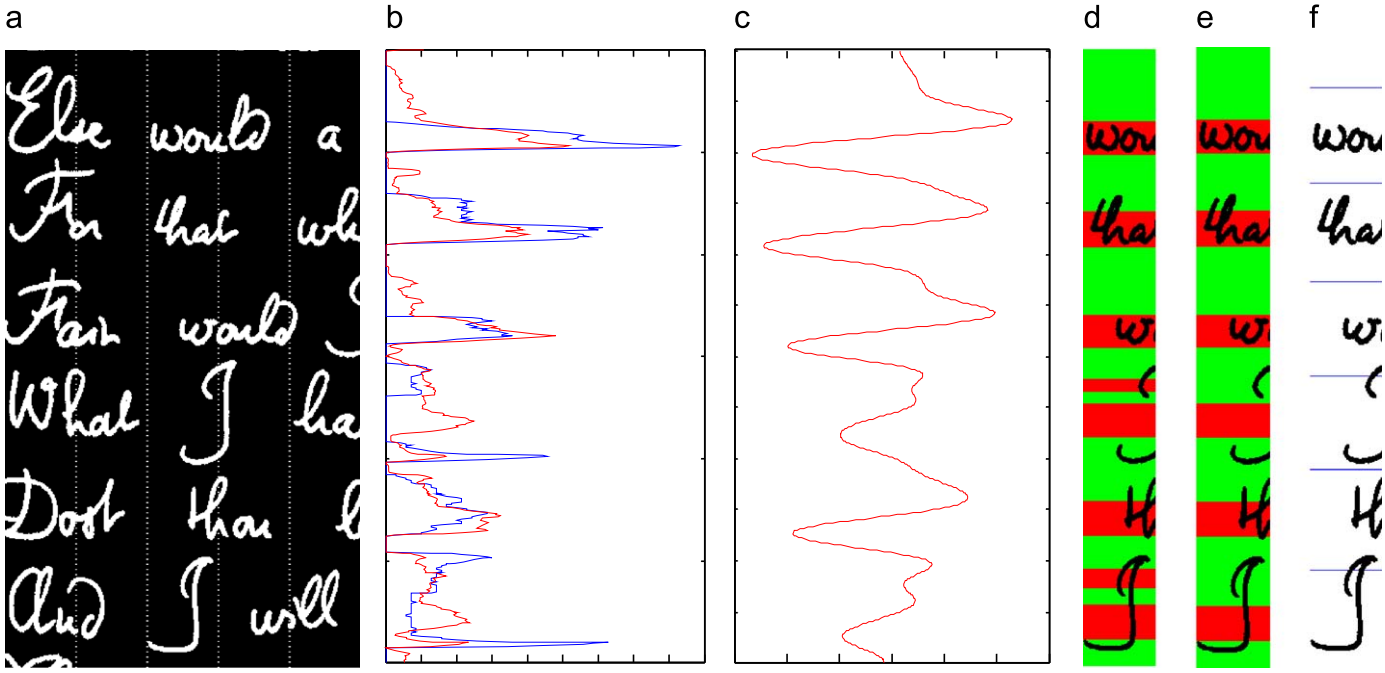

Fig. 3. Steps for locating the text-line separators in a fragment of zone 3 of image013.tif from ICDAR-07: (a) fragments of zones 1–5; (b) the projection profile (blue) of zone 3 and the corresponding smoothed one (red); (c) the first derivative; (d) initial text (red) and gap (green) regions; (e) refined text (red) and gap (green) regions; (f) final text-line separators for zone 3.

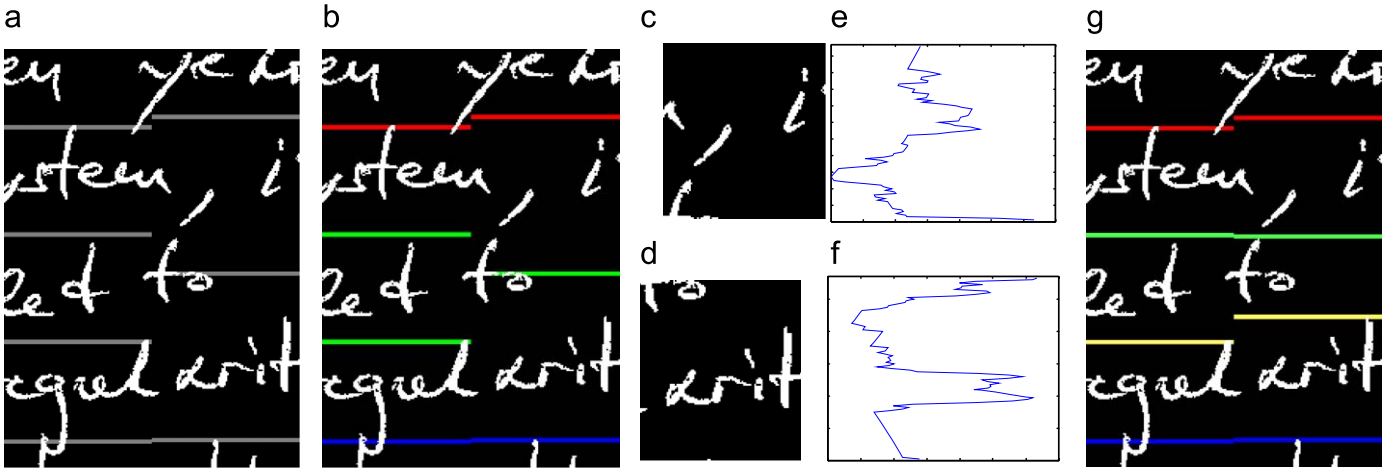

Fig. 4. Example of text-line separators drawing algorithm for 15th zone of image 025.tif from ICDAR-07: (a) the text-line separators for 14th and 15th zones; (b) the separators with same colors are associated; (c) and (d) the areas in which we search for new separators; (e) and (f) the graphs of the metric function for the areas in (c) and (d); (e) paired separators of 14th and 15th zones.

separators along the width of the document image combines text-line separators of adjacent zones starting from the leftmost textual zone.

Let us focus on the association of the separators in the i-th zone with the ones in the (i+1)-th zone denoted by $\{\psi_{i+1}^k,\ k=1,\ldots,S_{i+1}\}$ (Fig. 4a). First, each ψ_i^j is associated with the nearest ψ_{i+1}^k. Three possible cases would occur for the (i+1)-th zone's separators:

(a) ψ_{i+1}^k is associated with only one separator in the i-th zone, e.g. red and blue separators in Fig. 4b. Then we have the case of one-to-one correspondence and no further process is required.

(b) ψ_{i+1}^k is associated with r separators of the i-th zone denoted by ψ_i^ℓ, $\ell=j,\ldots,j+r-1$ (green separators in Fig. 4b). This happens due to text lines endings or large gaps between words. Then, for each ψ_i^ℓ we define the stripe L in (i+1)-th zone which is bounded by the closest to ψ_i^ℓ separators of the (i+1)-th zone (Fig. 4c, d). In the L stripe we would place new separators which will be associated with the respective separators in the i-th zone. The new separators should lie near their matching pair in the i-th zone and should avoid crossing foreground pixels, as much as possible. These requirements can be expressed as a minimization problem of the following function with respect to the row m of the L stripe

$$Q_m=(d_m+c_1)\cdot(P_m+c_2), \tag{7}$$

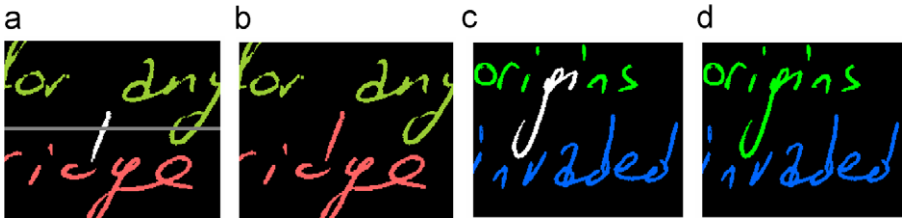

Fig. 5. Examples of CCs assignment to text lines in image 014.tif from ICDAR-07: (a–b) extension of the zone on either side. The ratios for the CC with id 257 (in white) are $r_{green}=0.2072$ and $r_{red}=0.1248$. The red text line contains 66% of the CC, and the CC is assigned to this text line. (c–d) extension of the zone on either side. The ratios for the CC with id 29 (in white) are $r_{green}=0.8552$ and $r_{blue}=0.3175$. The CC is assigned to the green text line.

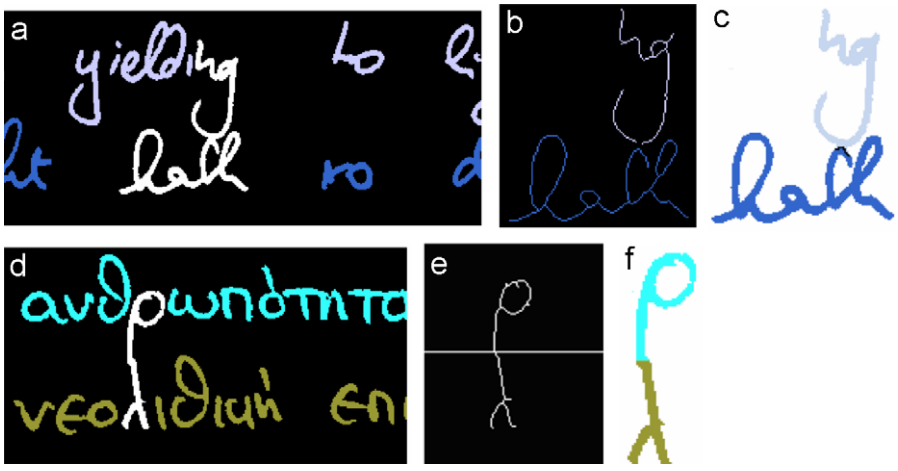

Fig. 6. Segmentation of CCs running along two text lines: (a) extension of the zone for CC with id 210 in image 013.tif from ICDAR-07; (b) the skeleton is segmented in three parts; (c) assignment of segments of the CC to text lines; (d) extension of the zone for CC with id 161 in image 003.tif from ICDAR-07; (e) CC's skeleton and the corresponding text-line separator. No junction points can be found; (f) split of the CC at the intersection point with the separator.

where d_m denotes the normalized distance of the m-th row from ψ_i^ℓ, P_m is the normalized value of projection profile of the L stripe at the m-th row, and c_1 and c_2 are constants both set to be equal to one.

The first product factor in Eq. (7) models the distance between the new separator in the (i+1)-th zone and the separator ψ_i^ℓ in the i-th zone. The second term corresponds to the crossing foreground pixels of the new separator in the (i+1)-th zone. Figs. 4e and f illustrate the graph of the function defined in Eq. (7) for the regions of Figs. 4c and d, respectively. Two pairs of associated separators (light green and yellow) are produced (Fig. 4(g)). From Fig. 4b one may observe that if c_1 was equal to zero, the new separator in 15th zone would result by extending the third separator of 14th zone and would hit the stress mark and "t" of the word "regularity".

(c) ψ_{i+1}^k is not associated with any separator in the i-th zone. This happens when a text line of the document begins from this zone onwards. In this case we have to find the respective separators in previous zones. For this purpose, we apply the procedure described above starting from the current zone and moving to the left.

3.4. CCs' assignment to text lines

The ultimate step of a text-line separation algorithm is to assign foreground pixels of CCs to text lines. This procedure is carried out with the application of a simple geometrical constraint. A CC is assigned to a text line if its intersection with the area defined by the boundaries of the text line exceeds a certain threshold R. A reasonable value for the threshold R would be 75% of the CC height (see Section 5 for more details). CCs that do not satisfy this constraint have to be decided whether they should be split, as in the case of a CC running into successive text lines, or not, as in the case of CCs resulting from ascenders or descenders.

Suppose that a CC in the m-th vertical zone runs into text lines j and j+1. We extend the m-th zone horizontally on either side up to the point where the new area includes a reasonable number of foreground pixels that unquestionably have been assigned either to line j or to j+1 (Figs. 5a, c, 6a, d). In the extended area, let N_j^b and N_{j+1}^b be the number of foreground pixels assigned to lines j and (j+1), respectively. Similarly, let N_j^a and N_{j+1}^a be the number of foreground pixels assigned to lines j and (j+1) that lie at the same horizontal line (x-coordinate) with the pixels of the CC under consideration. The ratios $r_k=N_k^a/N_k^b$, for $k=j$, (j+1) can be considered as a measurement of attraction of the CC to either text line. If both ratios are below a threshold (experimentally set to 0.4), the CC is assigned to the text line with the greatest overlap with the CC (Figs. 5a, b). This is the case of a CC like a stress mark or a segment of a broken character. If only one ratio is over the threshold, the CC is assigned to the text line with the highest ratio (Fig. 5c, d). This is the case of CCs that involve ascenders and/or descenders.

If both ratios are over the threshold, the CC under consideration runs along two text lines (Fig. 6a) and need to be split. We determine the skeleton [17] of the CC and find the junction points (pixels with more than two neighbours) with the application of a 3×3 mask. We focus on the junction points that lie near the separator between the j-th and (j+1)-th text lines by excluding the junction points whose distance from the separator is greater than a threshold, e.g. half of

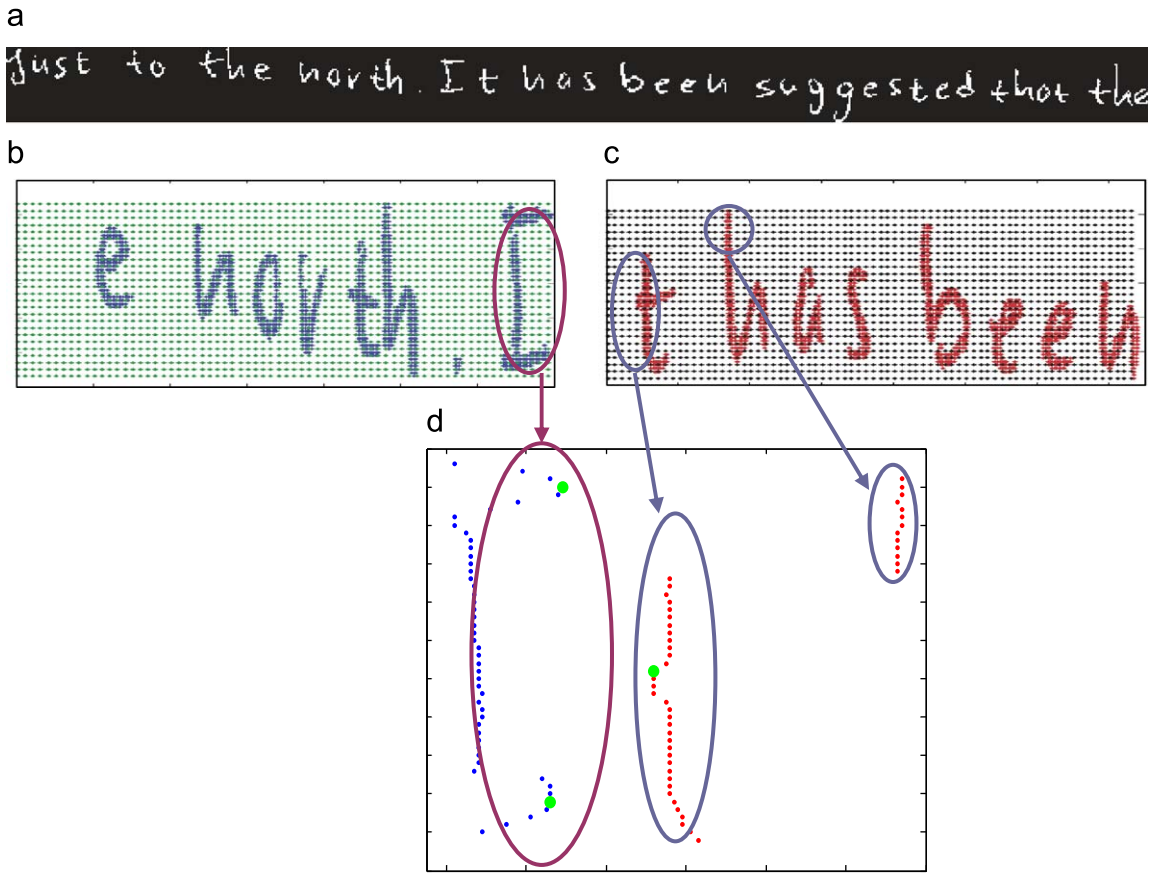

Fig. 7. Estimation of the gap metric for the candidate word separator between "*l*" and "*t*" in text line 10 of image 022.tif from ICDAR-07: (a) the text line; (b) the zones of the "left" group; (c) the zones of the "right" group; (d) the pixels of Z_k^c for each group and the resulting support vectors (in green).

the mean height of all CCs in the document page. Starting from the junction points nearest to the separator we apply the following procedure. Firstly, we convert this point and its neighbours into background pixels; hence the skeleton is segmented into two or more parts. Then, for each part we estimate the ratios r_k as mentioned above. If all parts can be assigned to text lines by applying the aforementioned constraint (Fig. 6b), we select this point as a separator and each pixel of the CC is assigned to the text line of the nearest skeleton point (Fig. 6c). Otherwise, we repeat this process with the next junction point. If no suitable junction point can be found, the CC is split at the point of intersection with the separator (Figs. 6d–f).

4. Word segmentation

In our approach, word segmentation requires that the document is already segmented in text lines. We also assume that given a text line, each CC belongs to only one word, i.e. successive words are not attached to each other. As a result, candidate word separators would lie at the gap between two successive CCs. Therefore, word segmentation can be seen as a problem which requires the formulation of a gap metric and the clustering of the gaps in "within" or "between" words classes.

4.1. Gap metric

We introduce a novel metric for measuring the separability between successive CCs. Considering the pixels of two successive CCs as elements of two distinct classes, a reasonable choice to measure their separability would be the margin of an SVM classifier that separates these components. We would employ a linear kernel since more complex kernels, such as RBF or polynomial, would result in larger margins for CCs with significant vertical overlapping that most probably belong to the same word. Moreover, a linear kernel guarantees lower complexity. In addition, we formulate the problem as a soft-margin SVM to allow penetration of pixels from either class into the margin zone.

Let g_k^ℓ be the gap metric between the k-th and the $(k+1)$-th CCs of the ℓ-th text line. The foreground pixels are divided into two groups, namely, the "left" group consists of the pixels of all CCs up to the k-th, and the "right" group involves the pixels from the $(k+1)$-th up to the last CC. Fig. 7b illustrates the group of pixels on the left of "*l*" and Fig. 7c on the right of "*t*".

We introduce the variables $x_m \in X_k \subseteq \mathbb{R}^2$ that correspond to the 2-d coordinates of the m-th foreground pixel and $y_m \in Y_k = \{-1, 1\}$ to denote the group that the m-th pixel belongs to. More compactly, the dataset for the k-th gap is denoted by $Z_k = (X_k, Y_k)$. Since the gap metric will be derived from SVM theory [18], we consider a subset Z_k^c of the Z_k by keeping only the foreground pixels that affect significantly the support vectors of the word separator. We obtain Z_k^c by splitting each group into non-overlapping horizontal zones of height equal to 2-pixels. For each zone we keep the q right-most pixels for the "left" group and the q left-most pixels for the "right" group. In our approach we set q equal to 4. In Fig. 7d, the selected pixels for the "left" group are shown in blue color and the pixels for the "right" group are shown in red. The aim for reducing the number of points is to increase the tractability of the SVM without altering the resulting gap metric.

The primary objective function for the soft margin SVM for the dataset Z_k^c is given by

$$L(\mathbf{w},b,a,\xi)=\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{|Z_k^c|}\xi_i - \sum_{i=1}^{|Z_k^c|}\alpha_i\{y_i[(x_i\cdot\mathbf{w})+b]-1+\xi_i\}-\sum_{i=1}^{|Z_k^c|}\mu_i\xi_i, \quad (8)$$

where $(\mathbf{w},b)$ define the hyperplane, ξ_i are the slack variables, a_i and μ_i are the Lagrange multipliers, C is a non-negative constant used to penalize classification errors, x_i are the feature space data points—in our case are the 2-d coordinates of the foreground pixels—and $|Z_k^c|$ is the cardinality of the dataset. The value C is chosen to be inversely proportional to $|Z_k^c|$. The optimal classifier for the two groups results from the minimization of L, i.e. the lowest value of L corresponds to the smallest $\|\mathbf{w}\|$ and consequently to the largest margin. We, therefore, define the gap metric between the k-th and the (k+1)-th CCs of the ℓ-th text line as

$$g_k^\ell=-\log\left\{\min_{0<\alpha_i\le C}(L)\right\} \quad (9)$$

Note that the transform in the log domain is introduced to enhance small size differences in L and the minus sign so that the gap metric increases with respect to the margin.

4.2. Threshold estimation

Using the above mentioned procedure we calculate the gap metrics for every selected pair of successive CCs in the whole document page. In order to classify the gaps into "within" or "between" words, we need to calculate a threshold. Due to the high variability of writing styles, sizes of letters, etc., a global threshold across all documents would be an inadequate solution. On the other hand, a variable threshold for each text line would not encompass the information of the writer's style. We, therefore, calculate the threshold taking into account all gap metric values within a given document page.

We denote by $G=\{g_m\}_{m=1}^M$ the set of all gap metrics in a document page. In principle, one should expect that low values correspond to small gaps, i.e. "within" word class, and vice versa. We employ a non-parametric approach for estimating the probability density function [19] of the gap metrics using the following formulae

$$p(x)=\frac{1}{Mh}\sum_{t=1}^{M}K\left(\frac{x-x^t}{h}\right) \quad (10)$$

where $K(\cdot)$ denotes the normal kernel.

Fig. 8 illustrates the normalized histogram of gap metric values in a document page and the estimated probability density function. One can identify two main lobes, with the right most corresponding to the "between" words gaps. The threshold is chosen to be equal to the minimum between the two main lobes of the estimated probability density function of the gap metrics.

5. Experimental results

In order to test the two algorithms (text-line and word segmentation) we used the datasets from the ICDAR07 Handwriting Segmentation Contest [20]. The training and the test datasets consist of 20 and 80 document images, respectively. The associated ground truth and the corresponding evaluation software were provided by the organizers. We developed two executables (one for each algorithm) in the form of a Win32 console application for evaluation on the test dataset. Both the ground truth and the result information

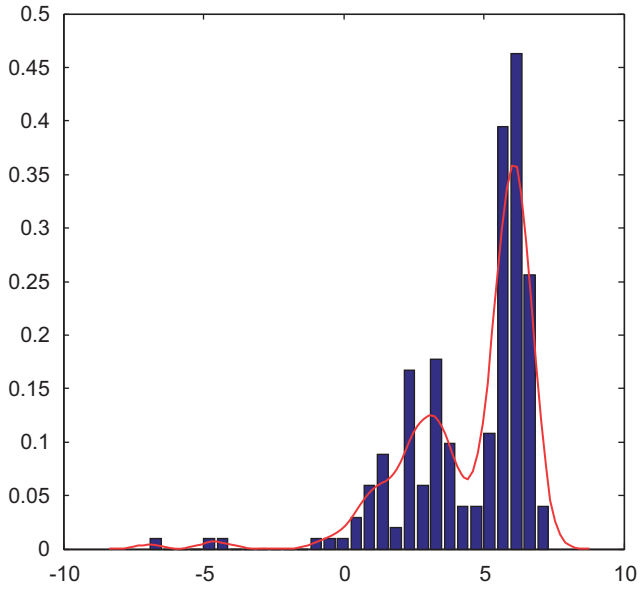

Fig. 8. The normalized histogram of the gap metric values and the estimated pdf of gap metrics in document image 003.tif from ICDAR-07.

were raw data image files with zeros corresponding to background and all the other values defining different segmentation regions.

The document images in the datasets cover a wide range of cases which occur in handwriting. "The documents used in order to build the training and test 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 (lines, drawings, etc.) and were all written in several languages including English, French, German and Greek", as the organizers report [9]. The document images are binary and theirs dimensions vary from 650×825 to 2500×3500 pixels.

5.1. Performance evaluation

The performance evaluation method is based on counting the number of matches between the entities detected by the algorithm and the entities in the ground truth [21]. A *MatchScore*(*i*,*j*) table was employed, representing the matching results of the j ground truth region and the i result region [22]. A match is only considered if the matching score is equal to or above a specified acceptance threshold T_a. It must be noted that T_a was set to 95% and 90% for line and word detection, respectively. If G and F are the numbers of ground-truth and result elements, respectively, and w_1, w_2, w_3, w_4, w_5, w_6 are 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 calculated as follows:

$$DR=w_1\frac{o_g2o_d}{G}+w_2\frac{o_g2m_d}{G}+w_3\frac{m_g2o_d}{G} \quad (11)$$

$$RA=w_4\frac{o_d2o_g}{F}+w_5\frac{o_d2m_g}{F}+w_6\frac{m_d2o_g}{F} \quad (12)$$

where o_g2o_d and o_d2o_g denote the one to one matches, o_g2m_d denotes one ground truth to many detected, m_g2o_d denotes many ground truth to one detected, o_d2m_g denotes one detected to many ground truth, and m_d2o_g denotes many detected to one ground truth.

The evaluation results for line and word detection on both datasets are shown in Table 1. Concisely, on 80 documents (test set) containing a total of 1771 lines and 13 311 words, 1738 lines and

Table 1
Detailed evaluation results.

	Training set		Test set	
	Lines	Words	Lines	Words
G	476	3832	1771	13 311
F	475	3905	1774	13 426
o_g2o_d	473	3570	1738	12 181
o_g2m_d	1	11	3	269
m_g2o_d	0	16	20	829
o_d2m_g	0	8	10	373
m_d2o_g	2	22	6	557
DR (%)	**99.42**	**93.34**	**98.46**	**93.57**
RA (%)	**99.68**	**91.61**	**98.20**	**92.46**

Table 2
Comparison of the algorithms.

Algorithm		*DR* (%)	*RA* (%)	*FM* (%)	*SM* (%)
BESUS	Lines	86.6	79.7	83.0	73.1
	Words	80.7	52.0	63.3	
DUTH-ARLSA	Lines	73.9	70.2	72.0	70.7
	Words	80.2	61.3	69.5	
ILSP-LWSeg	**Lines**	**97.3**	**93.0**	**97.1**	**94.2**
	Words	**90.3**	**92.4**	**91.3**	
PARC	Lines	92.2	93.0	92.6	85.4
	Words	84.3	72.8	78.1	
UoA-HT	Lines	95.5	95.4	95.4	92.5
	Words	91.7	87.6	89.6	
RLSA	Lines	44.3	45.4	44.8	60.1
	Words	76.9	74.0	75.4	
Projections	Lines	68.8	63.2	65.9	61.6
	Words	69.2	48.9	57.3	
Proposed	**Lines**	**98.46**	**98.20**	**98.33**	**95.67**
	Words	**93.57**	**92.46**	**93.01**	

12 181 words were segmented correctly. In the results presented in the following tables, the value of N and R equal to 20 and 0.75, respectively. These values exhibit the best performance on the training dataset.

The proposed algorithms outperformed the participants' algorithms, as well as state-of-the-art techniques as RLSA and Projection Profiles in both tasks [9] as shown in Table 2, where *FM* is the *F*-measure of *DR* and *RA* and the combined performance metric *SM* is the mean value of *FM*s.

By observing carefully the results for each document, we conclude that the text-line segmentation algorithm can deal with the variation of skew, character size, irregular patterns for page margins, as well as occurrences of touching lines. It must be mentioned that the proposed line segmentation algorithm is not effective when the majority of characters are broken into many fragments. This is due to the CC assignment method mentioned in Section 3.4. The integration of a morphological process, e.g. dilation with anisotropic filters, as a preprocessing module, would be adequate to overcome this problem. The word segmentation algorithm fails in the cases where the writing style varies significantly across the document image. This problem may be rectified by varying the threshold according to the writing style, i.e. text lines that include more dense words.

Table 3
Performance evaluation of text-line segmentation for various values of N.

N	*DR* (%)	*RA* (%)	*FM* (%)	N	*DR* (%)	*RA* (%)	*FM* (%)
6	93.04	91.06	92.04	26	98.38	98.15	98.26
8	94.15	93.88	94.01	28	98.29	98.00	98.15
10	96.47	95.62	96.04	30	98.38	98.04	98.21
12	97.43	96.81	97.12	40	97.63	96.50	97.06
14	97.80	97.30	97.55	50	97.37	95.55	96.45
16	98.31	98.01	98.16	60	96.80	94.07	95.42
18	98.21	97.83	98.02	70	95.23	90.90	93.01
20	98.46	98.20	98.33	80	94.00	88.80	91.33
22	98.48	98.21	98.34	90	92.80	85.81	89.17
24	98.07	97.69	97.88	100	91.98	84.12	87.87

Table 4
Performance evaluation of text-line segmentation for various values of R.

R	#CCs	*FM* (%)	R	#CCs	*FM* (%)
0.55	157	95.46	0.80	1554	98.57
0.60	318	97.84	0.85	1821	98.57
0.65	462	98.54	0.90	1872	98.57
0.70	684	98.61	0.95	1886	98.57
0.75	984	98.57	1.00	1892	98.57

5.2. Robustness test for text line

In this section we show the performance of the text line segmentation algorithm for different values of N, the number of vertical zones, and R, the parameter used for the assignment of CCs in text lines. In Table 3, one can see the performance of the text-line segmentation algorithm for different values of N. One may observe that for values of N between 16 and 30, the corresponding FMs vary from 97.88% to 98.33%. For lower values of N, the results are getting worse because the algorithm can not handle the skew angle variation within each zone. In addition, as N rises, say over 30, the zones are getting too narrow to locate text and gap regions.

We also tested the text-line segmentation algorithm for different values of R (see Section 3.4). In Table 4, one can observe that although the number of not initially allocated CCs increases with R, as expected, the FM remains almost unaffected for values of R above 0.65.

6. Conclusions

We have presented two effective techniques for segmenting handwritten documents into text lines and words. In line segmentation, the document image is divided in vertical zones and the extreme points of the piece-wise projection profiles are used to over-segment each zone in "gap" and "text" regions. Then, statistics of an HMM are estimated to feed Viterbi algorithm in order to find the optimal succession of text and gap areas in each zone. Line separators of adjacent zones are combined with respect to their proximity and the local foreground density. Text line separators are drawn in staircase function fashion across the whole document. Finally, CCs are assigned to text lines or are cut in a suitable junction point of their skeleton by applying a simple geometrical constrain.

In word segmentation, a novel metric is used to measure the separability of adjacent CCs. The gap metric results from the value of the objective function of a soft-margin linear SVM used to separate consecutive CCs. Then, the underlying pdf of the gap metric values is estimated for the whole document page. At last, the candidate word separators are classified as "within" or "between" words by using the rightmost minimum of the pdf as a threshold for the document page.

The algorithms tested on the two benchmarking datasets of ICDAR07 handwriting segmentation contest. It was shown that the

proposed approaches achieved better results than the other participating techniques and the two state of the art algorithms (Projections and RLSA). Since no prior knowledge is used, we believe that these approaches are appropriate for handwritten document retrieval systems.

Acknowledgment

The authors would like to thank for the support by the Greek Secretariat for Research and Technology under the program PENED-03/251.

References

[1] Z. Razak, K. Zulkiflee, et al., Off-line handwriting text line segmentation: a review, International Journal of Computer Science and Network Security 8 (7) (2008) 12–20.
[2] B. Yanikoglu, P.A. Sandon, Segmentation of off-line cursive handwriting using linear programming, Pattern Recognition 31 (12) (1998) 1825–1833.
[3] G. Louloudis, B. Gatos, C. Halatsis, Text line detection in unconstrained handwritten documents using a block-based Hough transform approach, in: Proceedings of International Conference on Document Analysis and Recognition, 2007, pp. 599–603.
[4] C.-H. Chou, S.-Y. Chu, F. Chang, Estimation of skew angles for scanned documents based on piecewise covering by parallelograms, Pattern Recognition 40 (2) (2007) 443–455.
[5] M. Arivazhagan, H. Srinivasan, S. Srihari, A statistical approach to line segmentation in handwritten documents, in: Proceedings of SPIE 2007, vol. 6500T.
[6] D.J. Kennard, W.A. Barrett, Separating lines of text in free-form handwritten historical documents, in: Proceedings of International Workshop on Document Image Analysis for Libraries, 2006, pp. 12–23.
[7] S. Nicolas, T. Paquet, L. Heutte, Text line segmentation in handwritten document using a production system, in: Proceedings of International Workshop on Frontiers in Handwriting Recognition (IWFHR'04), October 26–29, 2004, pp. 245–250.
[8] G. Seni, E. Cohen, External word segmentation of off-line handwritten text lines, Pattern Recognition 27 (1994) 41–52.
[9] B. Gatos, A. Antonacopoulos, N. Stamatopoulos, ICDAR2007 handwriting segmentation contest, in: Proceedings of International Conference on Document Analysis and Recognition, 2007, pp. 1284–1288.
[10] T. Stafylakis, V. Papavassiliou, V. Katsouros, G. Carayannis, Robust text-line and word segmentation for handwritten documents images, in: Proceedings of International Conference on Acoustics, Speech and Signal Processing, 2008, pp. 3393–3396.
[11] Y. Li, Y. Zheng, D. Doermann, S. Jaeger, Script-independent text line segmentation in freestyle handwritten documents, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (8) (2008) 1313–1329.
[12] S.N. Srihari, B. Zhang, C. Tomai, S. Lee, Y.C. Shin, A system for handwriting matching and recognition, in: Proceedings of Symposium Document Image Understanding Technology, 2003, pp. 67–75.
[13] M. Feldbach, K.D. Tonnies, Line detection and segmentation in historical church registers, in: Proceedings of International Conference on Document Analysis and Recognition, 2001, pp. 743–747.
[14] U.V. Marti, H. Bunke, Text line segmentation and word recognition in a system for general writer independent handwriting recognition, in: Proceedings of International Conference on Document Analysis and Recognition, 2001, pp. 159–163.
[15] R. Manmatha, J.L. Rothfeder, A scale space approach for automatically segmenting words from historical handwritten documents, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8) (2005) 1212–1225.
[16] R. Dugad, U.B. Desai, A tutorial on hidden Markov models, Technical Report no. SPANN-96.1, Signal Processing and Artificial Neural Networks Laboratory, Department of Electrical Engineering, Indian Institute of Technology, Bombay, India, 1996.
[17] T. Pavlidis, Algorithms for Graphics and Image Processing, Computer Science Press, Rockville, USA, 1982 pp. 195–208.
[18] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery 2 (2) (1998) 121–167.
[19] E. Alpaydin, Introduction to Machine Learning, The MIT Press, Cambridge, USA, 2004.
[20] ⟨http://www.icdar2007.org/competition.html⟩ Handwriting Segmentation ⟨www.iit.demokritos.gr/~bgat/HandSegmCont2007⟩.
[21] B.A. Yanikoglu, L. Vincent, Pink Panther: a complete environment for ground-truthing and benchmarking document page segmentation, Pattern Recognition 31 (9) (1994) 1191–1204.
[22] I. Phillips, A. Chhabra, Empirical performance evaluation of graphics recognition systems, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (9) (1999) 849–870.

About the Author—VASSILIS PAPAVASSILIOU received the Diploma degree in the Electrical and Computer Engineering and the M.Sc. degree in language technologies from the National Technical University of Athens (NTUA), Greece, in 1998 and 2000, respectively. Currently he is pursuing his Ph.D. degree at NTUA on handwritten character recognition and works for the Institute for Language and Speech Processing as a research assistant.

About the Author—THEMOS STAFYLAKIS received the Diploma degree in the Electrical and Computer Engineering from the National Technical University of Athens (NTUA), Greece and the M.Sc. degree in communication and signal processing from Imperial College London, UK in 2004 and 2005, respectively. Currently he is pursuing his Ph.D. degree at NTUA on speaker recognition and indexing and works for the Institute for Language and Speech Processing as a Research Assistant.

About the Author—VASSILIS KATSOUROS received the M.Sc. and Ph.D. degrees in Electrical and Electronics Engineering from Imperial College, London, UK in 1993 and 1997, respectively. Since 1998 he has been with the Institute for Language and Speech Processing first as researcher and since 2007 as senior researcher. His research interests involve signal processing, probability and statistics, game theory, image processing and pattern recognition.

About the Author—GEORGE CARAYANNIS received the Ph.D. degree in engineering from the University of Paris 7, France, in 1973 and the Ph.D. Doctorat d' Etat degree from the University of Paris-Sud/Orsay, France in 1978. Since 1984, he has been working as a Professor at the National Technical University of Athens, Greece. He is currently director of the Institute for Language and Speech Processing and president of "Athena"—Research and Innovation Center in Information, Communication and Knowledge Technologies. His research interests include speech processing and recognition, pattern recognition, image processing and computer vision, and natural language processing.