

ICDAR2009 handwriting segmentation contest

B. Gatos · N. Stamatopoulos · G. Louloudis

Received: 23 November 2009 / Revised: 26 May 2010 / Accepted: 27 May 2010 / Published online: 25 June 2010
© Springer-Verlag 2010

Abstract ICDAR 2009 Handwriting Segmentation Contest was organized in the context of ICDAR2009 conference in order to record recent advances in off-line handwriting segmentation. The contest includes handwritten document images produced by many writers in several languages (English, French, German and Greek). These images are manually annotated in order to produce the ground truth which corresponds to the correct text line and word segmentation result. For the evaluation, a well-established approach is used based on counting the number of matches between the entities detected by the segmentation algorithm and the entities in the ground truth. This paper describes the contest details including the dataset, the ground truth and the evaluation criteria and presents the results of the 12 participating methods as well as of two state-of-the-art algorithms. A description of the winning algorithms is also given.

Keywords Handwritten text line segmentation · Handwritten word segmentation · Document image processing evaluation

1 Introduction

One of the most important and challenging tasks in a handwritten recognition pipeline is the segmentation of handwritten document images into text lines and words. Several problems inherent in handwritten documents such as the difference in the skew angle between text lines or along the same text line, the existence of adjacent text lines or words touching, the existence of characters with different sizes and variable intra-word gaps seriously affect the segmentation and, consequently, the recognition accuracy. Therefore, it is imperative to have a benchmarking dataset along with an objective evaluation methodology in order to capture the efficiency of current and new practices in handwritten document segmentation.

Following the successful organization of the ICDAR 2007 Handwriting Segmentation Contest [1], we organized the ICDAR 2009 Handwriting Segmentation Contest in order to record recent advances in off-line handwriting segmentation. Two new benchmarking datasets, one for text line and one for word segmentation, were created in order to test and compare recent algorithms for handwritten document segmentation in realistic circumstances. For the evaluation, a well-established approach that is also employed by other document segmentation contests [1–3] is used.

The remainder of the paper is organized as follows. In Sect. 2, the contest details and an overview of the datasets are described, while, in Sect. 3, the performance evaluation method and metrics are described. Each of the participating methods is summarized in Sect. 4 and the results of the competition are presented in Sect. 5. Finally, the winning methods for line and word segmentation are detailed in Sect. 6, and the conclusions are drawn in Sect. 7.

B. Gatos (✉) · N. Stamatopoulos · G. Louloudis
Computational Intelligence Laboratory,
Institute of Informatics and Telecommunications,
National Center for Scientific Research “Demokritos”,
GR-153 10 Agia Paraskevi, Athens, Greece
e-mail: bgat@iit.demokritos.gr

N. Stamatopoulos
e-mail: nstam@iit.demokritos.gr

G. Louloudis
e-mail: louloud@iit.demokritos.gr

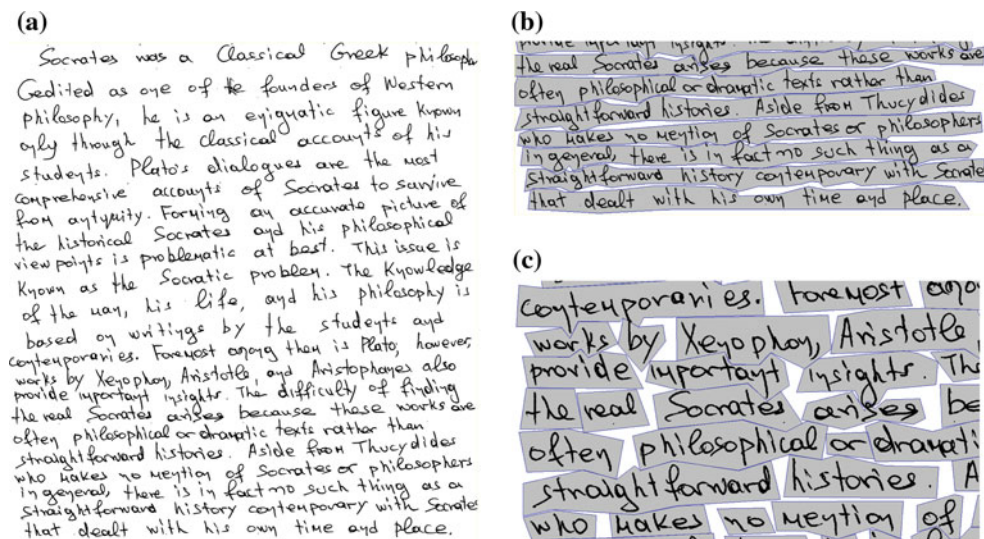


Fig. 1 a A sample of a handwritten document image part of the test dataset; samples of (b) text line and (c) word segmentation ground truth annotations

2 The contest

In this contest, we focused on the evaluation of text line and word segmentation methods using a variety of scanned handwritten documents. Based on these documents, we manually annotated the ground truth for text line and word segmentation and created the benchmarking datasets. The authors of candidate methods registered their interest in the competition and downloaded the training dataset (100 document images and associated ground truth from the ICDAR 2007 Handwriting Segmentation Contest [1]) as well as the corresponding evaluation software. At a next step, all registered participants were required to submit two executables (one for text line segmentation and one for word segmentation). Both the ground truth and the result information were raw data image files with zeros corresponding to the background and all other values defining different segmentation regions. After the evaluation of all candidate methods, the testing dataset (200 images and associated ground truth) along with the evaluation software became publicly available [4].

The documents used in order to build the training and test datasets came from several writers that were asked to copy a given text. All documents did not include any non-text elements (lines, drawings, etc.) and were written in several languages (English, French, German and Greek). The test set was created with the help of 50 writers that were asked to copy 4 pages each one corresponding to a different language. As a result, several produced documents are not written in the native language of the writer. A sample of a handwritten document image which is part of the test set and samples of text line and word segmentation ground truth annotations can be seen in Fig. 1. As it can be observed from Fig. 1, since there were no guidelines given to the writers, the produced images

contain all challenging problems for handwritten document segmentation, e.g. difference in the skew angle between text lines or along the same text line, existence of adjacent text lines or words touching, existence of characters with different sizes and variable intra-word gaps.

3 Performance evaluation

The performance evaluation method used was based on counting the number of matches between the entities detected by the algorithm and the entities in the ground truth [5]. We used a MatchScore table whose values are calculated according to the intersection of the ON pixel sets of the result and the ground truth.

Let I be the set of all image points, G_j the set of all points inside the j ground truth region, R_i the set of all points inside the i result region, $T(s)$ a function that counts the elements of set s . Table $MatchScore(i, j)$ represents the matching results of the j ground truth region and the i result region:

$$MatchScore(i, j) = \frac{T(G_j \cap R_i \cap I)}{T((G_j \cup R_i) \cap I)} \quad (1)$$

We consider a region pair as a one-to-one match only if the matching score is equal to or above the evaluator's acceptance threshold T_a . If N is the count of ground-truth elements, M is the count of result elements, and $o2o$ is the number of one-to-one matches, we calculate the detection rate (DR) and recognition accuracy (RA) as follows:

$$DR = \frac{o2o}{N}, \quad RA = \frac{o2o}{M} \quad (2)$$

A performance metric FM can be extracted if we combine the values of detection rate and recognition accuracy:

$$FM = \frac{2 DR RA}{DR + RA} \quad (3)$$

A global performance metric SM for handwriting segmentation is extracted by calculating the average values for FM metric for text line and word segmentation.

The performance evaluation method is robust and well established since it is also employed by other document segmentation contests [1–3]. It depends only on the selection of the acceptance threshold T_a . For this reason, in Sect. 5 we also report on experiments using various acceptance thresholds values (T_a).

4 Methods and participants

We had 12 submissions to the competition, while 4 of them included only a text line segmentation methodology. Brief descriptions of the methods are given in this section.

4.1 CASIA-MSTSeg method

Submitted by F. Yin, X.D. Zhou, Q.F. Wang and C.L. Liu of the Institute of Automation of Chinese Academy of Sciences (CASIA) in Beijing, China, and based on [6]. Connected components are first split based on several geometric constraints and then grouped into a tree structure by the minimal spanning tree (MST) algorithm with the distance metric designed by supervised learning. Text lines are extracted from the tree by dynamically cutting selected edges. Concerning word segmentation, for each gap between adjacent connected components in a line, 11 geometric features are extracted and fed to an SVM classifier for classifying gaps into between-word and within-word ones.

4.2 CMM method

Submitted by A. Hassaïne and B. Marcotegui of the Center of Mathematical Morphology in Paris School of Mines, France. A first labeling of the image is applied using the minimum number of horizontal intersections with the text. Cases of components with several labels or with labels that have to be merged are then handled based on several rules. For word detection, the average distance between the bounding boxes of the connected components of each line is computed. A distance is considered to be an inter-word distance if it is larger than a threshold.

4.3 CUBS method

Submitted by Z. Shi, S. Setlur and V. Govindaraju of the Center for Unified Biometrics and Sensors (CUBS), University at Buffalo, SUNY, in New York, USA. The line separation algorithm is based on an improved directional run-length analysis [7,8]. Concerning word segmentation, at each background pixel location, a horizontal background run including the location is traced and the run-length is saved in a new image buffer for each pixel location. A simple thresholding of the new buffer reveals word primitives. Then, the distances between the consecutive word primitives are computed using convex hull distance. A threshold for grouping of the word primitives is calculated based on the mean and variance of the distances.

4.4 ETS method

Submitted by D. Rivest-Henault and M. Cheriet of the Ecole de technologie superieure (ETS) of the University of Quebec in Montreal, Canada. Both text line and word segmentation methods are based on text smearing and morphological operations. Most of the involved operations take into account the local text line orientation. This has the benefit of greatly reducing the frequency of accidental line merging. The text is smeared using a modified version of Weickert's coherence-enhancing diffusion filter, while the smeared image is binarized using Otsu's algorithm.

4.5 ILSP-LWSeg-09 method

Submitted by V. Papavassiliou, T. Stafylakis, V. Katsouros and G. Carayannis of the Institute for Language and Speech Processing (ILSP) in Athens, Greece, and based on [9]. Text line detection makes use of the Viterbi algorithm. Candidate line separators are obtained and combined by minimizing a function which exploits the distance between the separators and the local foreground density. For word segmentation, as a metric of separability between the two sets (inter- and intra-word gap), the negative logarithm of the objective function of a soft-margin linear SVM is used.

4.6 JadavpurUniv method

Submitted by R. Sarkar, A. Khandelwal, P. Choudhury, N. Das, S. Basu, M. Kundu, M. Nasipury, D. K. Basu and A. F. Mollah of the CSE Dept., Jadavpur University in Kolkata, India. Text line detection is based on Connected Component Labeling and on comparison of components in a neighborhood. Then, the dimensional features of the components are analyzed to determine the style of handwriting, and threshold values are set for inter-word spacing in case of both isolated and cursive handwriting. Words are then

identified on the basis of difference in intra-word and inter-word spacing.

4.7 LRDE method

Submitted by T. Geraud of the EPITA Research and Development Laboratory (LRDE) in Le Kremlin-Bicetre, France. The input image is sub-sampled in both dimensions while turning it into a gray-level image. Then, an anisotropic Gaussian filtering is applied (mainly horizontal). The morphological watershed transform is computed, leading into a partition of the image into regions. To obtain the segmentation into lines, a simple merging procedure is run on the region adjacency graph. Word segmentation is based on attribute morphological closing as well as on morphological watershed transform. The source code can be downloaded from [10].

4.8 PAIS method

Submitted by S. Lu, S. Fan, Y. Wen and Y. Lu of the ECNU-SRI Joint Lab for Pattern Analysis and Intelligence System, Shanghai, China. The image is vertically divided into several strips. Potential text lines are detected based on the horizontal projection values of each strip in order to estimate the average distance between adjacent text lines. The text lines are then finalized by applying the knowledge of estimated line-distance and reasonable black-to-white traversal numbers. For word segmentation of each text line, the number of possible words is estimated by the black-to-white traversal numbers. A gap is considered as inter-word gap if it is larger than a threshold calculated by estimated number of possible words.

4.9 AegeanUniv method (text line segmentation only)

Submitted by E. Kavallieratou of the University of Aegean in Samos, Greece, and based on [11, 12]. The page is vertically separated into three areas and for each area, a horizontal projection profile is employed. The valleys with minima less than a certain threshold are considered to be likely beginnings of line segments. Sequentially, the area is examined pixel by pixel until an entire white path is outlined.

4.10 PortoUniv method (text line segmentation only)

Submitted by J. Cardoso of Faculdade de Engenharia, University of Porto in Porto, Portugal and based on [13]. The image is handled as a graph and the text lines as connected paths between the two lateral margins of the image. The paths to look for are the shortest paths between the two lateral margins while paths through black pixels are favoured. An efficient dynamic programming approach is used to find the minimum paths.

4.11 PPSL method (text line segmentation only)

Submitted by A. Alaei, P. Nagabhushan and U. Pal of the University of Mysore in Mysore, India. The text page is vertically crumbled into few strip-like structures. In order to get Potential Piece-wise Separation Line (PPSL) between two consecutive lines, the white/black spaces in each strip are analyzed. Next, such PPSLs are concatenated or extended in both directions to produce the complete segmentation lines based on distance analysis of each PPSL with left and right neighboring PPSLs.

4.12 REGIM method (text line segmentation only)

Submitted by M. Mezghani, W. Boussellaa and A. Alimi of the Research Group on Intelligent Machines (REGIM) of the University of Sfax in Tunisia and A. Zahour of the Equipe Gestion Electronique de Document (GED), University of Le Havre, France. The methodology is based on (a) document decomposition into columns and blocks covering all textual elements, (b) classification of the generated blocks using several statistical parameters and (c) text line detection based on a fuzzy base line determination using a fuzzy C-means algorithm.

5 Evaluation results

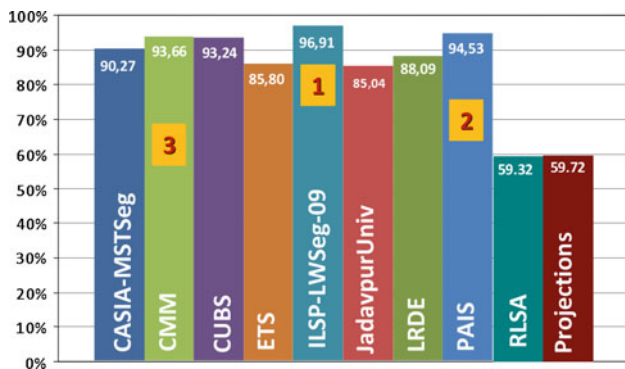
We evaluated the performance of all participating algorithms for text line and word segmentation using Eqs. (1)–(3), the test dataset (200 images) and the corresponding ground truth.

The acceptance threshold in order to define a one-to-one matching was set to $T_a = 95\%$ for text line segmentation and to $T_a = 90\%$ for word segmentation. The number of text lines and words for all 200 document images was 4034 and 29717, respectively. In order to compare the participant techniques with current state-of-the-art approaches, we also implemented an RLSA [14] and a projection profiles-based technique [15]. All evaluation results are shown in Table 1, while a graphical representation of the evaluation results is given in Figs. 2, 3, 4. In order to get an overall ranking for both text line and word segmentation, we used the global performance metric SM (see Sect. 3) in order to compare the 8 algorithms that provide both text line and word segmentation results (CASIA-MSTSeg, CMM, CUBS, ETS, ILSP-LWSeg-09, JadavpurUniv, LRDE and PAIS). As it can be observed (Fig. 2), the ILSP-LWSeg-09 method outperforms all other methodologies in the overall ranking, achieving $SM = 96.91\%$. The ranking list for the first five methodologies is:

1. ILSP-LWSeg-09 ($SM = 96.91\%$)
2. PAIS ($SM = 94.53\%$)

Table 1 Detailed evaluation results

		<i>M</i>	<i>o2o</i>	DR(%)	RA(%)	FM(%)	SM(%)
CASIA-MSTSeg	Lines	4049	3867	95.86	95.51	95.68	90.27
	Words	31421	25938	87.28	82.55	84.85	
CMM	Lines	4044	3975	98.54	98.29	98.42	93.66
	Words	31197	27078	91.12	86.80	88.91	
CUBS	Lines	4036	4016	99.55	99.50	99.53	93.24
	Words	31533	26631	89.62	84.45	86.96	
ETS	Lines	4033	3496	86.66	86.68	86.67	85.80
	Words	30848	25720	86.55	83.38	84.93	
ILSP-LWSeg-09	Lines	4043	4000	99.16	98.94	99.05	96.91
	Words	29962	28279	95.16	94.38	94.77	
Jadavpur Univ	Lines	4075	3541	87.78	86.90	87.34	85.04
	Words	27596	23710	79.79	85.92	82.74	
LRDE	Lines	4423	3901	96.70	88.20	92.25	88.09
	Words	33006	26318	88.56	79.74	83.92	
PAIS	Lines	4031	3973	98.49	98.56	98.52	94.53
	Words	30560	27288	91.83	89.29	90.54	
RLSA	Lines	4156	1615	40.03	38.86	39.44	59.32
	Words	30905	24006	80.78	77.68	79.20	
Projections	Lines	4391	2538	62.92	57.80	60.25	59.72
	Words	38340	20143	67.78	52.54	59.19	
AegeanUniv	Lines	4054	3130	77.59	77.21	77.40	–
PortoUniv	Lines	4028	3811	94.47	94.61	94.54	–
PPSL	Lines	4084	3792	94.00	92.85	93.42	–
REGIM	Lines	4563	1629	40.38	35.70	37.90	–

**Fig. 2** Overall evaluation performance for both text line and word segmentation

3. CMM (SM = 93.66%)
4. CUBS (SM = 93.24%)
5. CASIA-MSTSeg (SM = 90.27%)

Concerning text line segmentation, the CUBS method achieved the highest results with FM = 99.53% (Fig. 3).

The ranking list for the first five methodologies for text line segmentation is:

1. CUBS (FM = 99.53%)
2. ILSP-LWSeg-09 (FM = 99.05%)
3. PAIS (FM = 98.52%)
4. CMM (FM = 98.42%)
5. CASIA-MSTSeg (FM = 95.68%)

For the word segmentation stage, the ILSP-LWSeg-09 method obtained the highest results with FM = 94.77% (Fig. 4). The ranking list for the first five methodologies for word segmentation is:

1. ILSP-LWSeg-09 (FM = 94.77%)
2. PAIS (FM = 90.54%)
3. CMM (FM = 88.91%)
4. CUBS (FM = 86.96%)
5. ETS (FM = 84.93%)

For the sake of clarity, we also compared the first three methodologies concerning text line segmentation, word

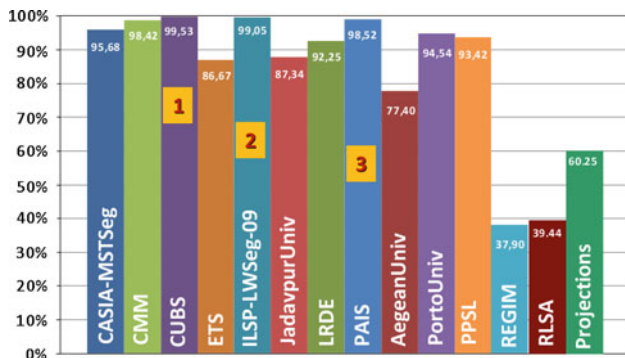


Fig. 3 Evaluation performance for text line segmentation

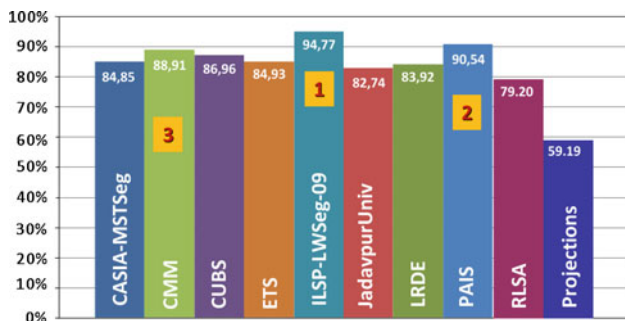


Fig. 4 Evaluation performance for word segmentation

segmentation as well as the global performance metric SM using various acceptance thresholds values (T_a) which range from 80% to 98% and interval value equal to 3. The overall results are represented in the graphs of Figs. 5, 6, 7. In all cases, the winning algorithms outperform the other techniques.

As it was mentioned in Sect. 2, the test set was created with the help of 50 writers that were asked to copy 4 pages each one corresponding to a different language (English, French, German and Greek). Among all documents, only the Greek documents were written in the native language of the writer. In order to compare the performance of the participating algorithms on documents written in the writers' native language with that obtained on documents in other languages, we also present the word segmentation results of the best 3 methodologies for each language independently in Fig. 8. As it is observed, the ranking for the Greek documents (written in the writers' native language) as well as for documents of other languages (English, French and German) remains the same compared to the total word segmentation ranking (see Fig. 4). It is noticed that the fact that several documents were not written in the native language of the writer does not seriously affect the results of the contest.

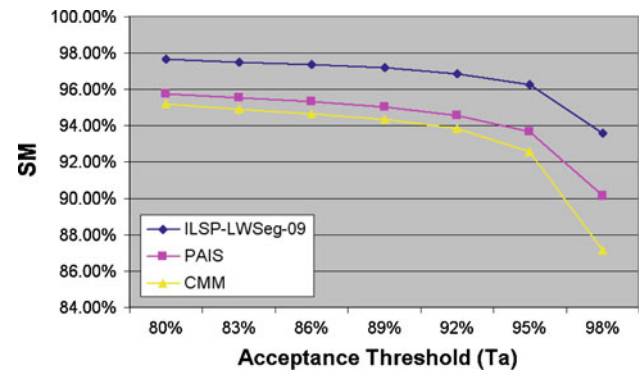


Fig. 5 Graphical detection of the comparison results for both text line and word segmentation using various acceptance threshold values

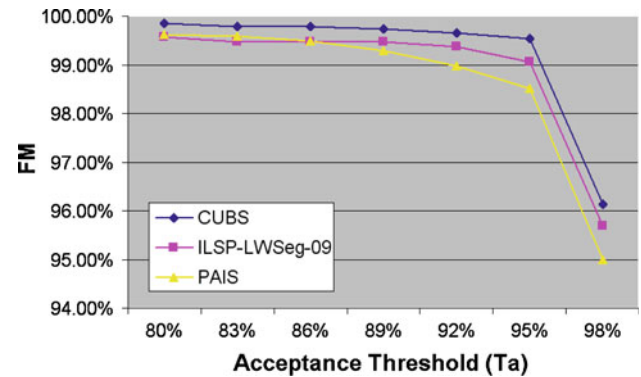


Fig. 6 Graphical detection of the comparison results for text line segmentation using various acceptance threshold values

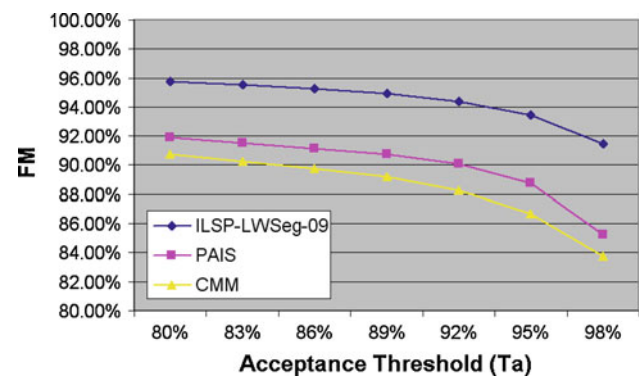


Fig. 7 Graphical detection of the comparison results for word segmentation using various acceptance threshold values

6 Winning algorithms

According to the evaluation results, presented in Sect. 5, the best performance in text line segmentation is achieved by CUBS method which has been submitted by Z. Shi, S. Setlur and V. Govindaraju. Concerning word segmentation, the ILSP-LWSeg-09 method, which has been submitted by V. Papavassiliou, T. Stafylakis, V. Katsouros and G. Carayannis, obtained the highest results. The strong point

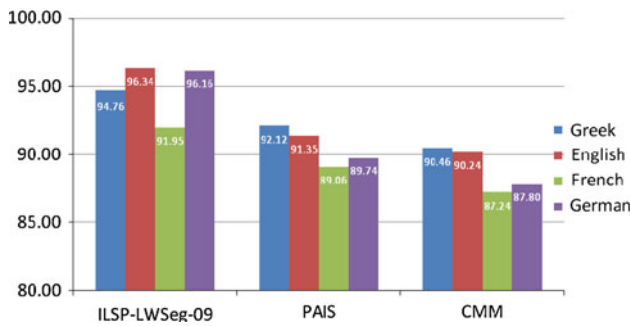


Fig. 8 Evaluation performance of the first three methodologies concerning word segmentation using documents of the same language

of the reported as best text line segmentation method (CUBS) is the use of an adaptive local connectivity map (ALCM) generated using a steerable direction filter. This method results to a very high accuracy results (FM = 99.53%). The main idea of the reported as best word segmentation method (ILSP-LWSeg-09) is the use of a gap metric that exploits the objective function of a soft-margin linear SVM in order to separate successive connected components. The performance of this method is reported to be 94.77% (FM) which means that is still room for improvements. Detailed descriptions of the winning algorithms are given in this section.

6.1 CUBS: text line segmentation algorithm

Text line segmentation is based on an adaptive local connectivity map (ALCM) generated using a steerable direction filter [8]. The algorithm is designed for solving the particularly complex problems seen in handwritten documents including fluctuating, touching or crossing text lines.

The text line segmentation method consists of the following steps: (1) Applying a steerable directional filter, a down-sampled version of the input document image is converted into an adaptive local connectivity map (ALCM), which is also a gray scale image. (2) A local adaptive thresholding algorithm is applied on the ALCM to reveal the text line patterns in terms of connected components. (3) A grouping algorithm is used to easily group the connected components into location masks for each text line. (4) Extraction of the text lines is done by collecting the connected components corresponding to the location masks on the original binary document image.

6.1.1 ALCM using steerable filter

Let $f : R^2 \rightarrow R$ represent any given signal. The document image is the discrete version of this signal with the domain limited to $\{0, 1, \dots, n - 1\} \times \{0, 1, \dots, m - 1\}$ and values in $\{0, \dots, 255\}$. Then, the adaptive local connectivity map is

defined as a transform

$$ALCM: f \rightarrow A \tag{4}$$

by the convolution:

$$A(x, y) = \int_{R^2} f(x, y) G_{a,b}^{\theta_0}(x - t, y - s) dt ds \tag{5}$$

where

$$G_{a,b}^{\theta_0}(x, y) = \begin{cases} 1 & \text{if } (x, y) \in E_{a,b}^{\theta_0} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $E_{a,b}^{\theta_0}$ is an ellipse with semi-minor axis a, b and rotated by an angle θ_0 :

$$E_{a,b}^{\theta_0} = \left\{ (x, y) \mid \begin{cases} x < a \cos(\theta - \theta_0) \\ y < b \cos(\theta - \theta_0) \end{cases} \text{ and } 0 \leq \theta \leq 2\pi \right\} \tag{7}$$

When we choose a longer than b , the ellipse $E_{a,b}^{\theta_0}$ is an elongated mask aligned with its long axis in θ_0 direction. Intuitively, using the steerable directional filter $G_{a,b}^{\theta_0}(x, y)$, ALCM is a convolution that aggregates the pixel intensities within the mask centered at (x, y) . When the long axis of the filter is aligned with the direction of a text line, the ALCM value $A(x, y)$ at the pixel location inside the text line will be greater than the value along any other direction.

6.1.2 Location of text lines

Each pixel value in an ALCM image represents the cumulative intensity of the foreground pixels in an elliptical neighborhood around the pixel in the original document image. A pixel with higher value in the ALCM implies that the pixel is in a dense text region. Therefore, ALCM is binarized for separation of highly-likely text areas from the background [16]. The binarization algorithm determines a pixel's binary value by considering the pixel intensity distribution in a 5 neighborhood block structure. The 5 neighborhood blocks are $5n \times n$ windows with one in the middle centered at the pixel under consideration and 4 other blocks adjacent to the corner of the center block. A weighted difference between the average pixel intensity in the middle block and that in the other 4 blocks is used to decide the center pixel's binary value. The binarized ALCM consists of connected components which represent either the entire line or part of a line.

6.1.3 Extraction of text lines

The line patterns extracted from the ALCM are location masks for the actual text lines. The text lines in the original document image are extracted by collection and grouping of connected components. The connected components for the text in the original document image are generated.

After up-sampling the line patterns in ALCM to the scale of the original image, we superimpose the line patterns on the document image. For each text line pattern, all the connected components of text touching the pattern are collected and these components together make up the text line. If there are some connected components that do not touch any line pattern, they are grouped with the closest line. Some connected components may belong to more than one line pattern. These components represent characters that cross multiple text lines. For a touching piece, a reference line is drawn between the line patterns. The segmentation algorithm segments the contours of the piece into contour segments. Based on the location of the center of mass of the contour segments relative to the reference line, they are grouped into the corresponding text lines. The text images are recovered using the contour segments.

6.2 ILSP-LWSeg-09: word segmentation algorithm

Word segmentation is based on a gap metric that exploits the objective function of a soft-margin linear SVM that separates successive connected components [9]. 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.

6.2.1 Gap metric

Considering the pixels of two successive CCs as elements of two distinct classes, the margin of an SVM classifier is used to measure their separability. 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. Figure 9b illustrates the group of pixels on the left of “I” and Fig. 9c on the right of “t”. The variable $x_m \in X_k \subseteq \mathbb{R}^2$ corresponds to the 2-d coordinates of the m -th foreground pixel and $y_m \in Y_k = \{-1, 1\}$ denotes 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 [17], they 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. Z_k^c is obtained by splitting each group into non-overlapping horizontal zones of height equal to 2-pixels. For each zone they keep the q right-most pixels for the “left” group and the q left-most pixels for the “right” group. In this approach, they set q equal to 4. In Fig. 9d, 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.

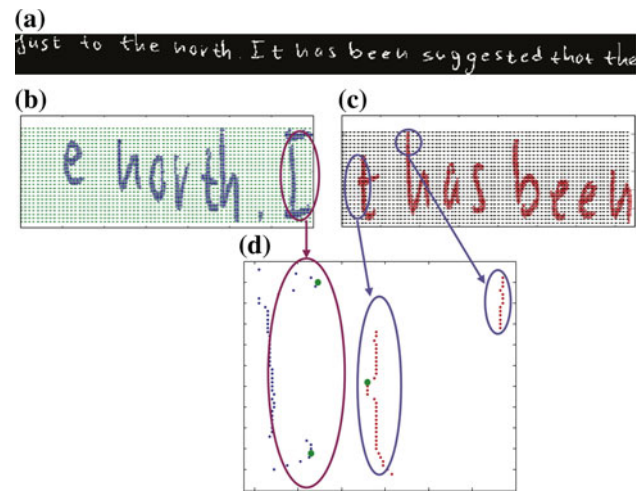


Fig. 9 Estimation of the gap metric for the candidate word separator between “I” and “t” in a text line: **a** the text line; **b** the zones of the left group; **c** the zones of the right group; **d** the pixels of $|Z_k^c|$ for each group and the resulting support vectors (in green)

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

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

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 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. Therefore, the gap metric between the k -th and the $(k+1)$ -th CCs of the ℓ -th text line is defined as:

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

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.

6.2.2 Threshold estimation

Using the above mentioned procedure, the gap metrics for every selected pair of successive CCs in the whole document page are calculated. 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. Therefore, the threshold is calculated by taking into account all gap metric values within a given document page.

Let $G = \{g_m\}_{m=1}^M$ denotes 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. A nonparametric approach for estimating the probability density function [18] of the gap metrics is calculated as follows:

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

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

7 Conclusions

ICDAR 2009 Handwriting Segmentation Contest was organized in order to record recent advances in off-line handwriting segmentation. As it is shown in the evaluation results section, the best performance considering an overall ranking for text line and word segmentation as well as a ranking only for word segmentation, was achieved by the ILSP-LWSeg-09 method of the Institute for Language and Speech Processing (ILSP) with overall global performance metric $SM = 96.91\%$ and word segmentation performance metric $FM = 94.77\%$. Considering only text line segmentation, the best performance was achieved by the CUBS method of the Center for Unified Biometrics and Sensors (CUBS) with performance metric FM equal to 99.53% . It is also recorded that the selection of the acceptance thresholds values (T_a) in order to define a one-to-one matching does not affect this ranking. Our plans for the construction of the test set of the next contest include the involvement of several writers that will all write in their native language. This will lead to more realistic and representative paradigms for testing.

References

1. Gatos, B., Antonacopoulos, A., Stamatopoulos, N.: ICDAR2007 handwriting segmentation contest. In: 9th International Conference on Document Analysis and Recognition (ICDAR'07), pp. 1284–1288. Curitiba, Brazil, September (2007)
2. Antonacopoulos, A., Gatos, B., Bridson, D.: ICDAR2007 page segmentation competition. In: 9th International Conference on Document Analysis and Recognition (ICDAR'07), pp. 1279–1283. Curitiba, Brazil, September (2007)
3. Antonacopoulos, A., Gatos, B., Bridson, D.: ICDAR2005 page segmentation competition. In: 8th International Conference on Document Analysis and Recognition (ICDAR'05), pp. 75–79. Seoul, Korea, August (2005)
4. <http://www.iit.demokritos.gr/~bgat/HandSegmCont2009/Benchmark>
5. Phillips, I., Chhabra, A.: Empirical performance evaluation of graphics recognition systems. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(9), 849–870 (1999)
6. Yin, F., Liu, C.-L.: Handwritten text line segmentation by clustering with distance metric learning. In: Proceedings of 11th International Conference on Frontiers in Handwriting Recognition, pp. 229–234. Montreal, Canada, August (2008)
7. Shi, Z., Setlur, S., Govindaraju, V.: Text extraction from gray scale historical document images using adaptive local connectivity map. In: 8th International Conference on Document Analysis and Recognition, (ICDAR'05), pp. 794–798. Seoul, Korea, August (2005)
8. Shi, Z., Setlur, S., Govindaraju, V.: "A steerable directional local profile technique for extraction of handwritten arabic text lines", to appear in 10th International Conference on Document Analysis and Recognition (ICDAR'09), Spain, July (2009)
9. Stafylakis, T., Papavassiliou, V., Katsouros, V., Carayannis, G.: Handwritten document image segmentation into text lines and words. *Pattern Recognit.* **43**(1), 369–377 (2010)
10. <http://olena.lrde.epita.fr/ModuleIcdar>
11. Kavallieratou, E., Dromazou, N., Fakotakis, N., Kokkinakis, G.: An integrated system for handwritten document image processing. *Int. J. Pattern Recognit. Artif. Intell.* **17**(4), 101–120 (2003)
12. Kavallieratou, E., Fakotakis, N., Kokkinakis, G.: An Off-line unconstrained handwriting recognition system. *Int. J. Document Anal. Recognit.* (4), 226–242 (2002)
13. Cardoso, J.S., Capela, A., Rebelo, A., Guedes, C.: A connected path approach for staff detection on a music score. In: Proceedings of the International Conference on Image Processing (ICIP 2008), pp. 1005–1008. San Diego, USA, October (2008)
14. Konidaris, T., Gatos, B., Ntzios, K., Pratikakis, I., Theodoridis, S., Perantonis, S.J.: Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. *Int. J. Document Anal. Recognit. (IJ DAR)* **9**(2–4), 167–177 (2007)
15. Antonacopoulos, A., Karatzas D.: Semantics-based content extraction in typewritten historical documents. In: Eighth International Conference on Document Analysis and Recognition, pp. 48–53 (2005)
16. Giuliano, E., Paitra, O., Stringa, L.: Electronic character reading system. U.S. Patent No. 4047152, September (1977)
17. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.* **2**(2), 121–167 (1998)
18. Alpaydin, E.: Introduction to Machine Learning. The MIT Press, Cambridge, USA (2004)