Noname manuscript No.
(will be inserted by the editor)

A Graph based Approach for Segmenting Touching Lines in Historical Handwritten Documents

David Fernández-Mota · Josep Lladós · Alicia Fornés

Received: date / Accepted: date

Abstract Text line segmentation in handwritten documents is an important task in the recognition of historical documents. Handwritten document images contain text-lines with multiple orientations, touching and overlapping characters between consecutive text-lines and different document structures making line segmentation a difficult task. In this paper we present a new approach for handwritten text line segmentation solving the problems of touching components, curvilinear text lines and horizontally-overlapping components. The proposed algorithm formulates line segmentation as finding the central path in the area between two consecutive lines. This is solved as a graph traversal problem. A graph is constructed using the skeleton of the image. Then, a path-finding algorithm is used to find the optimum path between text lines. The proposed algorithm has been evaluated on a comprehensive dataset consisting of five databases: ICDAR2009, ICDAR2013, UMD, the George Washington and the Barcelona Marriages Database. The proposed method outperforms the state of the art considering the different types and difficulties of the benchmarking data.

Keywords Text line segmentation · Handwritten documents · Document image processing · Historical document analysis

D. Fernández-Mota · J. Lladós · A. Fornés
Computer Vision Center - Computer Science Department,
Universitat Autònoma de Barcelona, Edifici O
08193 Bellaterra (Cerdanyola) Barcelona, Spain
E-mail: {dfernandez, josep, afornes}@cvc.uab.es

1 Introduction

There is an increasing interest to digitally preserve and provide access to historical document collections in libraries, museums and archives. This kind of documents are valuable cultural heritage, as they provide insights into both tangible and intangible cultural aspects. Historical archives usually contain handwritten documents. Examples are manuscripts written by well known scientists, artists or writers; as well as letters, trade forms or administrative documents kept by parishes or councils that help to reconstruct historical sequences in a given place or time. While machine printed documents, under minimum quality conditions, are easy to be read by OCR systems, handwritten document recognition is still a scientific challenge.

Layout segmentation and, in particular, line segmentation is a key step to guarantee a good performance of handwriting recognition. Not only for text transcription, but the segmentation of documents into text lines is an important process for several document analysis tasks, such as word spotting [42,44,51], or text alignment [26]. However, line segmentation is not a trivial process. Historical documents have several difficulties that can complicate the segmentation of text lines. First, the physical lifetime degradation of the original documents, related to the frequent handling and careless storage, produces holes, spots, broken strokes, ink bleed, winkles, etc. Second, if the scanning process has not been rigorous, it might introduce difficulties such as non stationary noise due to illumination changes, show-through effect, low contrast, warping, etc. Third, the inherent irregularity of handwriting is also a problem. Besides these general difficulties, the characteristics of the handwriting and the configuration of the text lines may provoke additional difficulties. First, a curvilinear

baseline due to the non-straight pen movement. Second, lines of crowded writing styles, which are more difficult to segment because they are close to each other and increase the overlapping. Third, the presence of touching and horizontally-overlapped components [17] when ascenders and descenders exceed the lower and upper bounds. And finally, punctuation and diacritic symbols, which are located between lines and introduce confusion in the decoding process of the physical structure. In Fig. 1 we illustrate some of the difficulties described above.

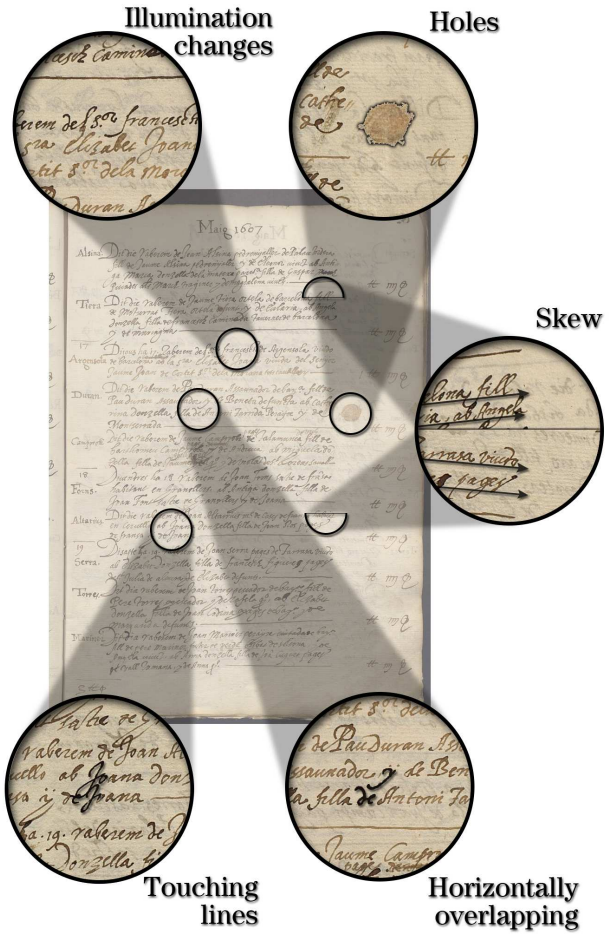


Fig. 1: Some difficulties in historical handwritten documents: illumination changes, holes, skew, horizontally-overlapping, and touching lines.

Although accurate algorithms for locating text lines in machine printed documents have been proposed [16, 38], they have showed some drawbacks in handwritten documents and there is still room for improvement. Several text-line segmentation algorithms for handwritten documents have been proposed (see Section 2). The methods can be classified as: projection-based [34], Hough based [33], grouping-based [11], morphology-based [48, 10] or other methods [21].

Text segmentation in handwritten documents can be divided in two tasks: localization and segmentation. Localization means to find the position of the text line, for example by its baseline, or central axis. Segmentation refers to a pixel-wise labelling. Localization has a good performance in highly structured documents, when text lines are isolated (as they follow rule lines or form boxes [31]) or when the methods are designed ad-hoc to a particular layout and document type. But when the stated difficulties of handwritten documents are present: touching lines, curvilinear text lines and horizontally-overlapping components, the performance decreases and the accurate segmentation is very difficult. Finer analysis processes are performed, especially in touch parts. Some methods use connected components to group the touching parts to the closest text lines [51,25], while other methods are more accurate and analyse touching parts [28,41]. The segmentation is done taking in account the properties and the shape of the studied area [52]. In this paper we present a line segmentation approach that in addition to the general difficulties of historical documents, tackles with these problems without loosing the generality, so the approach is writer-independent, layout-independent, and is able to cope with skew and warping disturb.

The main idea of our algorithm is, first, to estimate the localization of the text lines, and second, to segment the text lines. The accuracy of the location is not primordial because our algorithm is focused in finding the optimal path with minimum cost in-between two consecutive lines in the image background. This is solved as a graph traversal problem. Hence, the skeleton of the background image is converted to a graph. After finding potential starting and ending nodes, minimum cost paths between pairs of starting and ending nodes are searched. Local cost functions associated to the graph nodes are defined to find the best continuation path in terms of the potential configurations. The problem of touching text lines is solved adding virtual edges between the candidate nodes that are around the involved characters.

The rest of the paper is structured as follows. In section 2 the state-of-art is reviewed. Section 3 describes the proposed method. Section 4 shows the experimental results. Finally, we present the conclusions in the last section of the paper.

2 Related Work

Handwritten text line segmentation has received high attention over the last years [31]. In addition to rele-

Table 1: Comparative of text line detection methods.

	Printed documents	Handwritten documents	Skewed documents	Curved lines	Over-line	Touching lines
Projection-based	++	−−	+	−−	−−	−−
Hough-based	++	−	++	−−	−−	−−
Grouping-based	++	+	++	+	+	−−
Morphology-based	++	+	++	+	++	−−
Our method	++	++	++	+	++	+

Table 2: Classification of the methods according to the input images.

	Gray level	Binary
Projection-based	[34]	[2] [22] [23] [41] [55]
Hough-based		[33]
Grouping-based		[57] [58]
Morphological operations	[1] [35] [51]	[7] [24] [48] [10] [50]
Other methods	[4] [5] [21] [52]	[29] [32] [53]

vant publications, a series of competitions on this topic has been organized in international events (e.g. the ICDAR2009 Handwritten Segmentation Contest [13], with 12 participants, or ICDAR2013 Handwritten Segmentation Contest [54], with 14 participants). Observing the existing methods to segment handwritten documents, we propose a classification into 5 categories: projection-based, Hough-based, component grouping, morphology-based operations and other methods.

2.1 Taxonomy of methods

Projection-based methods are based in projection profiles. Black pixels are projected on the vertical axis. The maxima and the minima of the resulting histogram correspond to regions with large and low horizontal density of pixels. The lines are obtained computing the average distance between the peaks of the histogram [34,56]. Some techniques [2,41] can deal with variations in the text orientation, but they are sensitive to the size of characters and the gaps between successive words. To solve these problems, some methods [22,23] detect areas where two lines are merged due to long ascenders or descenders and compute local histograms to split the lines. The *PAIS* method [13] improves a line segmentation approach based in projections applying the knowledge of estimated line-distance and reasonable black-to-white traversal numbers.

Most of these techniques are simple and easy to implement, but they do not work efficiently with multi-skewed text lines, touching components and horizontally-overlapping component configurations.

Hough-based methods [33] describe parametric geometric shapes (straight lines, circles and ellipses are the most usual) and identify geometric locations that suggest the existence of the sought shape. They are proper methods to detect lines because text lines are usually parallel, and consequently in the Hough space they generate a configuration consisting of aligned peaks at a regular distance. Although these methods handle documents with variations in the skew angle between text lines, they are not accurate when the skew varies along the same text line, i.e. curvilinear lines. In addition, these methods can not achieve an accurate segmentation of touching or overlapping lines.

Grouping-based methods, also known as bottom-up strategies, group components according to a specific property. Most of the works belonging to this category are based in searching for components that are horizontally aligned. In [58,59] connected components are organized in a tree structure in terms of a metric distance, and grouped by a minimal spanning tree (MST) algorithm. The *CMM* method [13] groups components that are horizontally aligned. The *JadavpurUniv* method [13] analyzes dimension features of the components to determine the handwriting style and to set the threshold values for inter-word spacing. Yi et al. [30] propose an approach based on density estimation criteria to cluster components. Although grouping methods usually present problems to segment touching text lines, the method described in [30] includes a post-process that detects and splits them. Feldbach and Tonnies [11] join baselines segments, computed in a pre-process step, in historical church registers, similar to the main experimental focus of this paper. Kumar et al. [27] present a method that computes the similarity between text components based on local orientation detection and shortest path in graphs. The proposed

method can handle with printed documents and complex layouts in handwritten documents, however like the other grouping-based methods, it fails to segment touching text lines.

Morphology-based methods have been used in many works for layout analysis, especially when documents contain text blocks that are strictly oriented horizontally or vertically, i.e. columns and lines. Smearing-based operators can be seen as morphological methods with horizontal structuring elements. Particular examples are the methods described in [48,10]. They combine the two fundamental morphological operations (dilation and erosion) with horizontal projections and run-length smearing algorithm (RLSA) respectively. Other methods [35,50] use anisotropic Gaussian kernel or local estimation count map.

In the *LRDE* method [13], a morphological watershed transform is computed once the document is smoothed using an anisotropic Gaussian filter. Shi et al. [52] propose a technique based on a generalized adaptive local connectivity map which uses a steerable directional filter. In the *ETS* method [13], the text is smeared using a modified version of Weickest's coherence-enhancing diffusion filter to segment lines. Alaei et al. [1] use strip-like structures to decompose the text block in vertically parallel structures. Each one is labelled using their grey intensity and applying a morphological dilation operation. Nicolau et al. [37] shred text images into strips along the white gaps in between text lines. Saabni et al. [49] propose a method that computes an energy map of the input text block image and determines the seams that pass across text lines.

These kind of methods also have problems in documents with overlapping of adjacent text lines. To overcome this problem, some morphology-based works define ad-hoc heuristics [7] or min-cut/max-flow graph cut algorithm [24].

Graph based: Some approaches use graphs to compactly represent the image structure keeping the relevant information on the arrangement of text lines. Energy (or cost) functions are used to establish the optimal path between nodes that segment the lines. In [29] the segmentation is posed as a graph cut problem. The graph is built using either the pixels or the connected components of the image as nodes, which are linked to its neighbours through edges.

The *PortoUniv* method [13] represents the image as a graph, which is used to find the minimum energy paths between the borders of the page using an efficient dynamic programming approach. The robustness of projection based methods is combined with the flexibility of graph-based methods by Wahlberg et al. in [55]. The graph is constructed using the foreground of the image.

Other methods: There is a miscellanea of other methods that can not be classified into any of the main categories described above. Kass et al. [21] use active contours to explore the borders of the image objects with relevant differences between the foreground and the background in characteristics like brightness or colour. Bukhari et al. [4,5] adapt active contours (snakes) over the ridges of the geometry of the gray level image to detect the central axis of parts of text lines. The method properly localizes the text lines in the documents, even with the difficulties explained above. In case of touching components lying in two different text lines (a connected component lies over two text lines), they are horizontally of vertically cut depending on the slope of underlying ridges into equal number of parts. However, this can split words into different lines (e.g. ascenders or descenders of the words are split in the above or below text line). Liwicki et al. [32] use dynamic programming to find text lines, computing minimum cost paths in between consecutive text lines. Stafylakis et al. use a Viterbi algorithm to segment the text-lines [53].

2.2 Discussion

In order to summarize the above described methods for segment lines in handwritten documents, Table 1 overviews their strengths and weaknesses and Table 2 shows the type of documents (binary or grey level) usually used as input. The graph based methods are not included in the taxonomy because the methodologies used in these kind of approaches are too diverse and, therefore they are unable to be generalized under a common assessment. The rest of the methods are compared (Table 1) according to the following criteria: if the method works in printed and/or handwritten documents; if the method handles variations in the skew angle between text lines and when the skew varies along the same text line (curved lines) or not; if the method can solve the problem of horizontally-overlapping components; and if the method can properly split touching lines or not.

Text line segmentation in printed documents is a problem that has been solved from different approaches with satisfactory results. However, when dealing with handwritten documents, state-of-the-art methods, specially projection and Hough-based, present some difficulties to properly segment the lines. Errors in segmentation are usually due to noise and the non-rigid structure of this kind of documents. These irregularities

lead to the three main problems that are present in the handwritten text line segmentation: curved lines, horizontally-overlapping lines and touching lines. Methods based on morphological operations and grouping-based methods are able to deal with curved and horizontally-overlapping lines. However, the segmentation of two touching lines still remains as an unsolved problem among state-of-the-art methods.

Besides the taxonomy presented in Table 1, an additional criterion that is worth to be considered is whether the approach requires a learning process [20,60] or not [6]. The methods based on projection profiles have good accuracy when they are applied to documents with the same structure layout and style. The main problem of these approaches is their adaptability. They have to learn their models for every new document, for those that present a new structure layout, or a new handwriting style or a different time period. This kind of methods need some samples of every type of documents to learn a model. However, such samples are not always easily available. In addition to this drawback, learning-based methods have a higher computational cost, even though the learning process is an off-line process. The methods without a learning process are more adaptive. They can robustly extract the lines of any kind of documents and the computational cost is lower. However, the performance decreases when dealing with a close collection because they are not adapted to the specificities of that set.

Most of the methods presented above localize text lines with a high accuracy, but only a few of them focus their methods to solve the problem of overlapping and touching components [18,40,47]. Even so, Kang et al. [18] require a learning process to define local configurations of touching components. Ouwayed et al. [40] split touching components following the descending parts of the characters, which is a common property of most characters in the Arabic alphabet. Rohini et al. [47] localize touching components extracting the core region (space between consecutive text lines) using horizontal projections profiles. Then, the method needs a preprocess to deskew the curved text lines.

We have also classified the above methods according to the kind of input image: binary or grey-level. Usually, projection-based methods use binary images as input, except the method of Manmatha et al. [34] that uses a modified version of [14] extended to grey-level images. Hough-based and Grouping-based methods use binary images in their approaches because they perceptually group basic primitives (key-points or connected components). Morphological-based and other methods have a large diversity of algorithms and each one uses a different type of images.

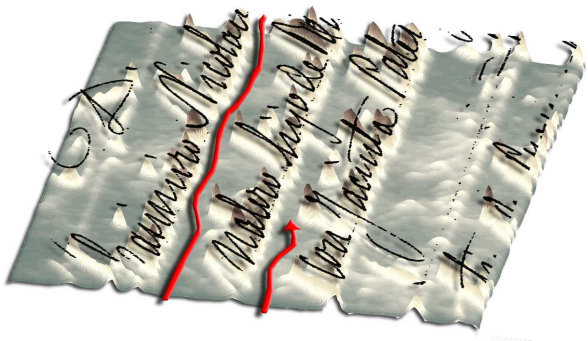


Fig. 2: In a 3D-view, text lines can been seen as peaks and the space between them as valleys.

From the comparative shown in Table 1, we can conclude that the main challenges are the segmentation of touching, horizontally-overlapping and curvilinear lines. The main contribution of the approach proposed in this paper is its robustness and high performance when segmenting lines under the above mentioned problems. From the comparative in Table 2, we conclude that most methods use binary images as input. In our approach, we assume that images have been previously binarized so it simplifies the process. However we will show how the method is robust to binarization noise so this process is not critical in the pipeline.

2.3 Contribution

We present an approach inspired by graph representation methods. Graphs are a useful tool to capture the structure of the image objects (lines and words in our case). In addition, graph theory offers solid and elegant methods. Graph vertices are usually constructed from pixels or connected components. Graph edges represent spatial relations between connected components and are usually weighted by the distance between the connecting vertices [3,57].

The main objective of our work is to localize text lines and to solve the problem of touching lines adding new *virtual edges* to the graph. These characters are split using some heuristics which evaluate the spatial information around the area involved. This technique is not oriented to a specific writer, style or alphabet, and it is able to cope with multi-oriented text lines and historical documents. The approach presented in this work belongs to the group of methods which do not need a learning process to segment lines, therefore it does not need labelled samples. Next, we explain this approach in detail.

3 Line Segmentation Approach

Humans tend to write text in blocks, and they usually use the same space between lines. In a 3D-view of the intensity image, this characteristic can be seen as a valley: if we compute the distance function and hence see the topography of the image, the in-line space is seen as a valley and the words as crests. Using this observation, we first compute the distance function in the input image, which corresponds to the skeleton of the background. Afterwards, we detect paths through the valleys of the document. The paths consist of background points at equal distance to the words above and below. We use a path-finding algorithm to select which paths are the best to segment the lines. In Fig. 2 we can observe a representation of this characteristic.

The goal of our method is to automatically locate and segment text line regions in handwritten documents. The system consists of two big stages, as shown in Fig. 3. The first stage is the enhancing of the documents and the localization of the text lines. Then, the second stage is the line segmentation. We compute the skeleton of the background image. All the possible pixel-paths are computed using an iterative thinning function. Then the paths are converted to a graph, which will be used to find the optimal paths that segment the text lines. Then, the best paths that segment the text lines are found. For this purpose, we adapt the A-star path finding algorithm. Finally, the consistency checking step is applied. Let us further describe the different steps.

3.1 Localization

The localization step includes an enhancing process of the documents. The main idea is to use the valleys that appear between the text lines to segment them. The words of the text lines represent the crest of the mountains, and the noise of the documents can introduce hills that produce the diversification of the valleys. This fact introduces new possible paths and the computational cost increases proportionally to the noise. The number of valleys is reduced applying morphological operations to smooth the image and to reduce the hilltops.

First, the image is binarized using the Otsu's method [39]. Several binarization methods have been tested in our document images: Niblack (Fig. 4a) generates noisy images, Sauvola (Fig. 4c) loses important information and the characters are thinned, Bernsen (Fig. 4b) generates good results but the computational time is very high. Otsu (Fig. 4d) obtains a clear image, without noise, the characters do not loose pixels and are well defined, and in addition, it is the fastest method.

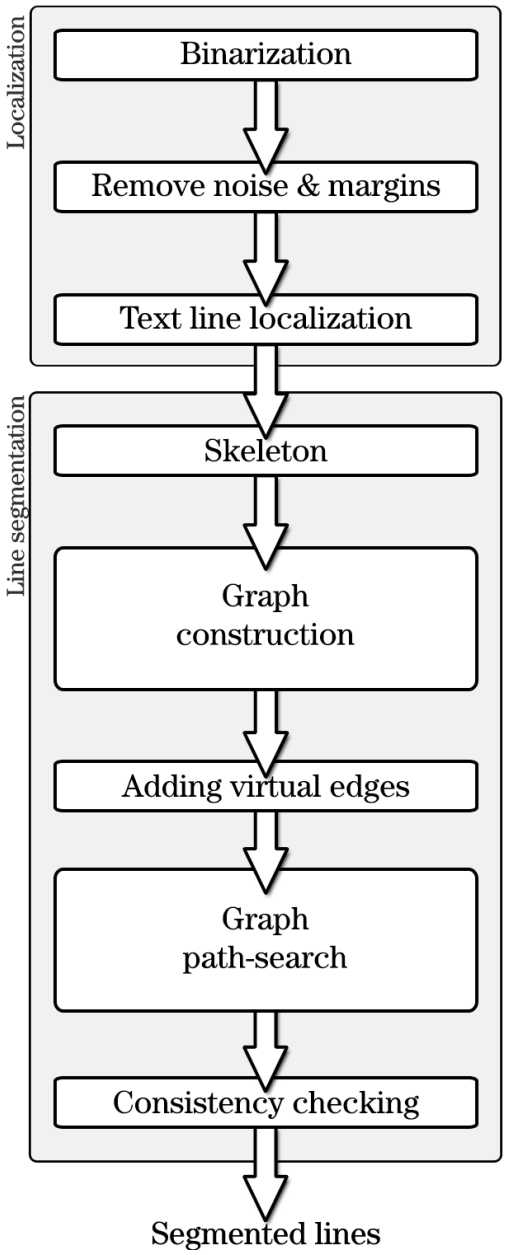
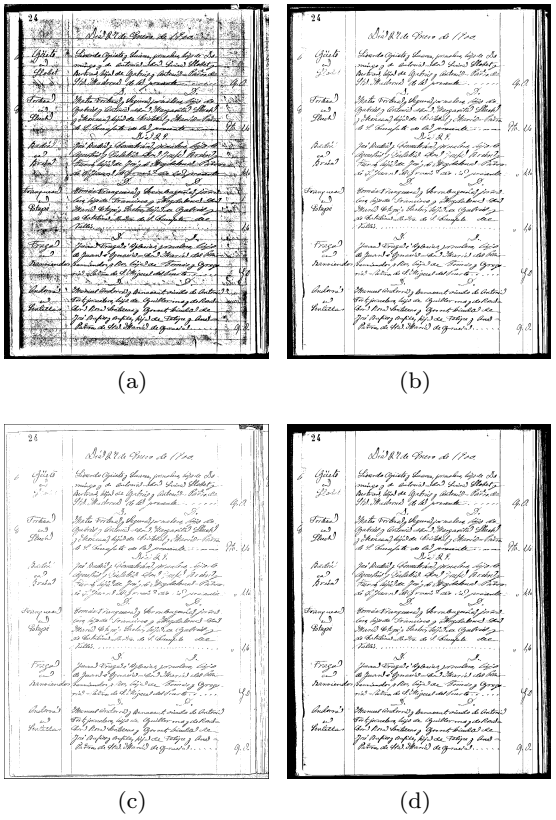


Fig. 3: Flowchart of the proposed approach.

The skeleton is a simplification of the topology of the image. In this work, the skeleton of the distance function applied to the background allows to obtain the seams between text lines (valleys of the distance transform image). Since the images are originally binary (white paper and black ink), computing the skeleton in the grey-level image would not give the same result so it would obtain distorted skeleton with extra-segments in images due the scanning process. Consequently, the path search algorithm has to analyze all these extra paths, so the computation cost increases exponentially. For this reason, we prefer to binarize the input images.



(a) (b) (c) (d)

Fig. 4: Binarization of the documents using several methods. (a) Niblack's method. (b) Bernsen's method. (c) Sauvola's method. (d) Otsu's method.

The scanning process introduces some distortions. One of them is the set of page margins (a black area around the page image). We delete these margins by applying morphological operations and selecting the biggest blobs in the periphery of the document. Then, a median filter is applied with a mask of size $4x4$ to remove the speckle noise. The problem of using this kind of filters is that they provoke a thinning of the characters. An alternate sequential filter of opening and closing operations is applied to correct this problem.

Once the image is binarized, text lines are localized using a projection-base method. To get a best accuracy, a rough estimation of the skew of the document is computed using the Wigner-Ville distribution [40].

3.2 Graph construction

The proposed method for segmenting lines in handwritten documents is based on searching the pixel paths of minimum cost on the skeleton of the background image between the left and the right margins of the previously localized text lines.

For the sake of efficiency, instead of directly processing the skeleton at pixel level, it is approximated by a graph $G = (V, E)$ that preserves its structure. The set of vertices V of the skeleton graph represents characteristic points (terminal and intersection points), and the set of edges E represents sequences of consecutive skeleton points between vertices. Formally a skeleton graph G is represented as an attributed graph $G = (V, E, L_V, L_E)$ where L_V and L_E are two labelling functions that assign attributes to nodes and edges respectively. The labelling functions L_V and L_E are defined as follows.

Given a vertex $v \in V$, the attributes assigned to it are denoted as:

$$L_V(v) = [N_v, x_v, y_v, t_v]$$

where N_v denotes the number of neighbours, $(x_v,\ y_v)$ are the coordinates of the pixel and t_v is the type of the node out of $\{\gamma_i,\ \gamma_f,\ \gamma_e,\ \gamma_c,\ \gamma_{cb},\ \gamma_{ct}\}$ (Fig. 5). These types of nodes represent the following configurations:

- γ_i: an *initial node* is a terminal pixel (with only one neighbour) of the skeleton located at the left margin of the image (first column of the image pixels).
- γ_f: a *final node* is a terminal pixel of the skeleton located at the right margin of the image (last column of the image).
- γ_e: an *ending node* defines the end of a path in the central part of the image. It is a terminal pixel of the skeleton located at any place of the image except the first and the last column of the image pixels.
- γ_c: a *connection node* represents a corner in a path, i.e. it has two incident edges (skeleton paths) with an important change in the orientation.
- γ_{cb}: a *bifurcation node* defines a branch (three incident edges). It is a pixel of the skeleton which has three neighbours.
- γ_{ct}: a *trifurcation node* defines a crossing (four incident edges). It is a pixel of the skeleton which has four neighbours.

An edge $e = (v_s, v_t) \in E$ stores the current path of chain pixels joining the source vertex $v_s \in V$ and the target vertex $v_t \in V$; the Euclidean distance between v_s and v_t; and the type of edge: *true edge* (when there is a true path of pixels between v_s and v_t) or *virtual edge*.

The problem of touching text lines is solved by adding virtual edges. Due to the geometry of the distance function image, two touching words provide a discontinuity in the path, i.e. the skeleton computed in this area creates two or more *ending points* around the place where the touching problem appears. These

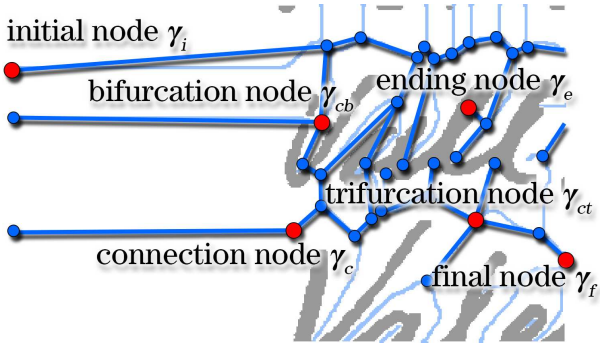
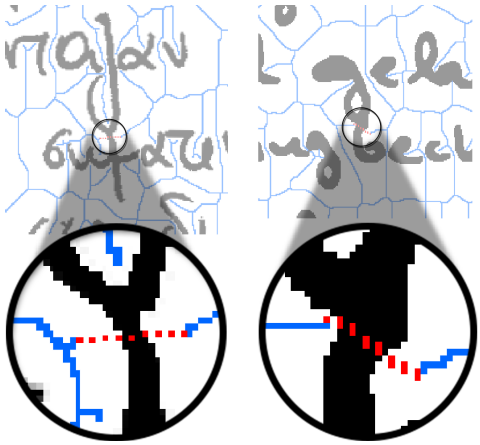


Fig. 5: Types of graph nodes computed from the skeleton image.



(a) (b)

Fig. 6: Adding virtual edges between two *ending nodes*.

ending nodes are used to solve this problem. We connect these nodes using new edges. These new edges are known as *virtual edges* (Fig. 6). These *virtual edges* are sub-path candidates. So they allow to reconstruct the broken path traversing the touching characters through the minimum path. A virtual edge is created between two ending nodes γ_e when they are very close. The threshold radius R_v (see Eq. 1) of the area around an ending node to search for other connecting nodes is experimentally set proportional to the size of the image. The mean of the separations between the text lines is estimated in the localization process. When t_e is a virtual edge, then p_e is empty.

$$R_v = \frac{\sum_{i=2}^{n} y_i - y_{i-1}}{n} \tag{1}$$

3.3 Graph path-search

Once the skeleton of the background image has been converted to a graph, the problem of text line finding is translated into searching for shortest paths in the graph according to some considerations. A-star (A*) is a computer algorithm that is widely used in path finding and graph traversal. Hart et al. [15] described the algorithm as an extension of the Dijkstra's 1959 algorithm [9].

The algorithm proposed in this work is a modified version of the classical A-star algorithm. In the classical algorithm the starting and the target point should be established. In our problem, we do not know a priori which node, among the final nodes, is the target node. For each *initial node* γ_i of the graph, the algorithm iteratively searches a minimum cost path until a final node γ_f is reached.

The objective of this step is to find the best path in the valley between two crests, or text lines, (Eq. 2). The solution is found as a minimum energy path in the graph G between an initial node $v_1 \in \{\gamma_i\}$ [1] and a final node $v_z \in \{\gamma_f\}$. A path P contains a sorted list of nodes and edges. The path P can be seen as a representation of the valley path between two text lines calculated based on the skeleton. We denote a path P^j as follows:

$$P^j = [v_1^j, e_1^j, v_2^j, e_2^j \ldots, e_{z-1}^j, v_z^j] \tag{2}$$

where $v_1^j \in \{\gamma_i\}$ and $v_z^j \in \{\gamma_f\}$.

The cost of a path P^j is the accumulated cost of the local transitions (local paths) between consecutive nodes. Formally:

$$p(P^j) = c(v_1^j, v_2^j, e_1^j) + c(v_2^j, v_3^j, e_2^j) + \cdots + c(v_{z-1}^j, v_z^j, e_{z-1}^j). \tag{3}$$

Given an initial node $v_1 \in \{\gamma_i\}$, the algorithm searches for the minimum cost path that reaches a final node $v_z \in \{\gamma_f\}$ in the opposite side of the page.

The algorithm searches the best path in the state-space S, where each state represents a partial path explored to this point. The algorithm explores in a state S_i all the possibles states $[S_i^1, S_i^2, \ldots, S_i^m]$ to go. An intermediate state S_i corresponds to a graph node v_n^j, and the next possible states correspond to all the possible next graph nodes v_{n+1}^j that are connected to the node v_n^j.

[1] For the sake of understanding we denote $v \in \{\gamma_i\}$ to represent a node belonging to the category of initial nodes (equally for the rest of types).

The transition from a state S_i to the next state S_{i+1} is computed in terms of some heuristic functions that model local configurations (explained with more details in the next paragraphs). Each transition has a cost of moving from a state to the next state according to a weighted combination of four predefined heuristics corresponding to four possible local configurations. Briefly, the heuristics give the cost of a path taking into account its trend, the bound of each text line, and also to avoid the possible backward paths and to solve the problems of touching-components and horizontally-overlapping objects. The next state corresponds to the minimum cost transition from the node v_n^j to the neighbour v_{n+1}^j through the edge e_n^j (chosen among all the possible nodes v_{n+1}^j connected to v_n^j). The cost of the path-step v_n^j to v_{n+1}^j through the edge e_n^j is denoted as $c(v_n^j, v_{n+1}^j, e_n^j)$. Formally:

$$c(v_n^j, v_{n+1}^j, e_n^j) = \alpha_1 * h_1(v_n^j, v_{n+1}^j, e_n^j) + \dots \\ \dots + \alpha_4 * h_4(v_n^j, v_{n+1}^j, e_n^j) \quad (4)$$

where α_i are the corresponding weights computed experimentally ($\alpha_1 = 1, \alpha_2 = 0.5, \alpha_3 = 0.01$ and $\alpha_4 = 0.2$), $v_n^j \in \{\gamma_i, \gamma_e, \gamma_c, \gamma_{cb}, \gamma_{ct}\}$, $v_{n+1}^j \in \{\gamma_f, \gamma_e, \gamma_c, \gamma_{cb}, \gamma_{ct}\}$ and e_n^j is the edge between v_n and v_{n+1}, which contains information as if it is virtual or real. To simplify the notation $x_{v_{n+1}^j}$ is denoted as x_{n+1}^j and $y_{v_{n+1}^j}$ is denoted as y_{n+1}^j.

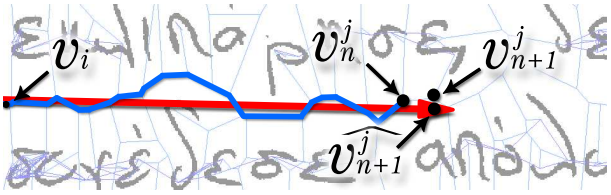
Let us further describe the four heuristics that are considered to model the cost function.

H1.- Trend Heuristic. Humans write text lines following a uniform direction, without abrupt orientation changes. Although a text line presents a curvilinear orientation, the local orientation trend predicts the smoothest continuation path. This property is used to fix the path to this trend and to avoid the possibility of sharp curves in the computed paths.

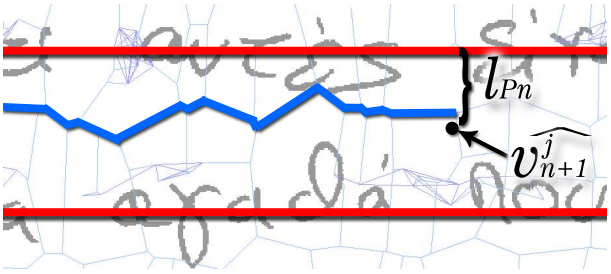
We use the trend of the path, computed by a linear regression, to compute the cost of the new node in the path taking into account the nearby nodes in the y axis. Given the estimated trend point $\widehat{v_{n+1}^j}$,the starting node $v_i \in \{\gamma_i\}$ and the source node v_n^j, the cost is the sum of the respective differences between those three points and the target node v_{n+1}^j. Formally:

$$h_1(v_n^j, v_{n+1}^j, e_n^j) = |f_{lr}(v_{n+1}^j) - y_{n+1}^j| \\ + |y_i^j - y_n^j| + |y_{n+1}^j - y_n^j| \quad (5)$$

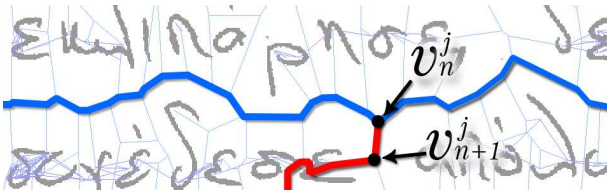
where $f_{lr}(v_{n+1}^j) = \widehat{y_{n+1}^j}$ is a linear regression obtained from all the nodes that compose the temporal path $[v_1^j, e_1^j, v_2^j, e_2^j \dots, v_n^j]$, where v_1^j is a starting node and v_n^j is the source node and $\widehat{v_{n+1}^j}$ is an estimation of v_{n+1}^j.



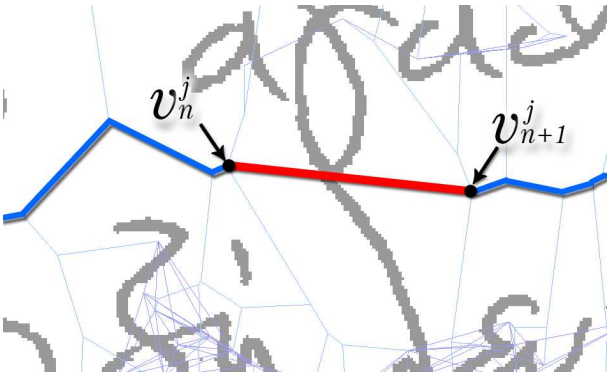
(a) **Trend Heuristic.** The red line represents the trend of the computed path (blue line).



(b) **Bounds Heuristic.** The red lines are the bounds for the computed path (blue line).



(c) **Back Heuristic.** The blue line represents the correct path computed. Red line is a wrong possible path going back.



(d) **Virtual Paths Heuristics.** A virtual path (red line) connects to ending nodes to solve the problem of touching lines.

Fig. 7: Illustration of the different heuristics.

H2.- Bounds Heuristic. Humans tend to write text lines parallel each other. We also take into account that it is not usual to cross lines when writing. The objective of this heuristic is to fix the path inside the upper and lower bounds of two text lines, defining a band

along which the path can not surpass (Fig. 7b). We fix the path between the upper and the bottom limit of each line. Some documents contain multi-skewed lines. To correct this problem, the bounds of each line are adapted dynamically at each iteration in terms of the current trend of the path. The graph edges located between the words of the same text line have a higher cost than the edges located between different text lines. Formally:

$$h_2(v_n^j, v_{n+1}^j, e_n^j) = \begin{cases} 0 & \text{if } f_{hb}(v_n^j, v_{n+1}^j) \leq l_{P_n} \\ f_{hb}(v_n^j, v_{n+1}^j) & \text{otherwise} \end{cases} \tag{6}$$

where $f_{hb}(v_n^j, v_{n+1}^j) = |f_{lr}(v_{n+1}^j) - y_{n+1^j}|$ and l_{P_n} is the limit estimated previously of this path.

H3.- Back Heuristic. Following the premises of the last two heuristics (H1 and H2), paths cannot go back abruptly. They have to follow a trend and it has to be inside a bound. Another premise is that the target of our path-finding algorithm is located on the right margin of the document, so paths that go backwards are penalized with a high cost.

The direction of the path is checked, and if it goes back, we increase the cost directly proportional to the retracted distance (Fig. 7c). Although the cost of this path is high, in some cases the algorithm chooses this option because there is no alternative or it is too expensive. Formally:

$$h_3(v_n^j, v_{n+1}^j, e_n^j) = \begin{cases} 0 & \text{if } x_{n+1}^j - x_n^j \geq 0 \\ d_e(v_n^j, v_{n+1}^j) & \text{otherwise} \end{cases} \tag{7}$$

where d_e is the Euclidean distance.

H4.- Virtual Paths Heuristic. As we have explained before, the problem of touching text lines is solved by adding virtual edges to the graph. In the construction process of the graph, we introduce virtual edges between intermediate ending points (Fig. 7d). We use this kind of edges when there is no alternative path (or the cost of the other paths is too high), or the alternative path has a high deviation passing through the words of the text lines above or below. Formally:

$$h_4(v_n^j, v_{n+1}^j, e_n^j) = d_e(v_n^j, v_{n+1}^j) \tag{8}$$

if e_n^j is a virtual path.

Algorithm 1 Path finding.

```
1: LIST_PATHS = NULL;
2: for all v_i^j ∈ V do
3:    OPEN_LIST = null;
4:    CLOSE_LIST = null;
5:    Insert v_i^j node in OPEN_LIST
6:    P* = null;
7:    while OPEN_LIST ≠ empty do
8:       Select the first node v_n^j ∈ V from OPEN_LIST,
         remove it from OPEN_LIST, and put it on
         CLOSED_LIST
9:       if v_n^j ∈ {γ_f} then
10:         exit
11:      end if
12:      Expand node v_n^j, generating the set V_M ⊆ V, of its
         successors that are not already ancestors of v_n in P*
13:      for all v_{n+1}^j ∈ V_M do
14:         Cost = getHeuristic(v_n, v_{n+1}^j, v_i^j, e_n)
15:         if v_{n+1}^j is not in OPEN_LIST then
16:            Insert v_{n+1}^j node in OPEN_LIST
17:         else
18:            v_eq = getOpenListNode(v_{n+1}^j);
19:            if Cost(v_{n+1}^j) < Cost(v_eq) then
20:               Update v_eq with new cost
21:            end if
22:         end if
23:      end for
24:   end while
25:   the path P* is obtained by tracing a path along the
      pointers from v_n^j to v_i^j ∈ {γ_i}.
26:   Add P* to LIST_PATHS
27: end for
```

In summary, the algorithm computes the best path in the valley between two crests, or text lines, for each initial node $v_i^j \in \{\gamma_i\}$. For each node $v_n \in V$, its branches $v_{n+1} \in V$ are expanded, and the *heuristic cost* h is computed from v_n^j to v_{n+1}^j. The branch with the minimum cost (sum of the *real cost* and the *heuristic cost*) is chosen. The *real cost* is the cost computed from the initial node v_i^j to the current node v_n^j. Contrary, the *heuristic cost* is an estimation of the cost from v_n^j to v_{n+1}^j. The algorithm ends when it finds an ending node $v_f^j \in \{\gamma_f\}$. The algorithm is summarized in **Algorithm 1**.

3.4 Consistency checking

Although the proposed method is able to cope with noise, some documents may present high levels of degradation. It results in wrongly segmented lines, in particular two consecutive paths may be overlapped (Fig. 8). To avoid this problem, the consistency checking process is a post-process that looks for the overlapped paths and splits them accordingly. During the graph path-search step, the algorithm checks the overlapped edges. Then, paths that share edges are split.

(a) Two paths are overlapped.

(b) One path is overlapped on two other paths.

Fig. 8: Examples of the different types of overlapping between paths.

Two kinds of overlapping can appear. The first one occurs when two paths are overlapped (Fig. 8a). The second one occurs when a path is overlapped with two other paths (Fig. 8b). In the first case, the overlapping is solved taking in account two facts: the high variability (in Y axis) in its nodes and the distance between the starting node and the closest text line estimated in the localization step. Text lines that start in the middle of an estimated text line, will be penalized. In the second case, the overlapping is solved removing the path which is overlapped in the other two paths.

4 Experimental Results and Discussions

We have experimented with five databases with increasing level of difficulty: the datasets from ICDAR2009 [13] and ICDAR2013 [54] Handwritten Segmentation Contest, the UMD Database [19], George Washington's manuscripts [43, 45] and the Barcelona Marriages Database [12]. The metrics used to evaluate the performance of our approach are the ones used in the ICDAR2013 Handwritten Segmentation Contest.

4.1 Metrics

To make the results comparable, the performance evaluation method used in this work is the same that the used in ICDAR2013 Handwritten Segmentation Contest [54]. It is based on counting the number of matches between the entities detected by the algorithm and the entities in the ground truth. A *MatchScore(i,j)* table was used, representing the matching results of the *j-th* ground truth region and the *i-th* resulting region.

$$MatchScore(i,j) = \frac{T(G_j \cap R_i \cap I)}{T((G_j \cup R_i) \cap I)} \quad (9)$$

Let I be the set of all images 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 points of set s.

A region means the set of foreground pixels likely to belong to a text line in each segment. A match is only considered if the matching score is equal to or above a specific acceptance threshold. Let N and M be the amount of pixels of the ground-truth and result elements respectively, and *o2o* is the number of one-to-one matches, we calculate the detection rate DR and recognition accuracy RA as in ICDAR2013 Handwritten Segmentation Contest [54]. Formally:

$$DR = \frac{o2o}{N}, \quad RA = \frac{o2o}{M} \quad (10)$$

A performance metric, called F-Measure FM, is computed combining the values of the detection rate (DR) and recognition accuracy (RA):

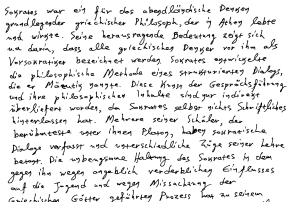
$$FM = \frac{2DR * RA}{DR + RA} \quad (11)$$

4.2 Datasets

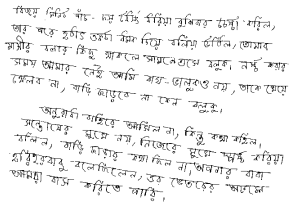
ICDAR2009 and ICDAR2013. The documents of the ICDAR2009 text line segmentation contest, consisting of 200 document binary images with 4034 text lines, came from several writers that were asked to copy a given text. None of the documents include any non-text elements (such as graphical lines, drawings, etc.), and were written in several languages (English, French, German, and Greek). The documents of the ICDAR2013 text line segmentation contest are similar to the previous one. The main difference is that there are more languages involved by including an Indian language. It consists in 150 document binary images with 2649 text lines.

These databases were specifically created for a competition on line segmentation. The text lines are roughly straight and horizontal. There are only few documents that present multi-skewed text. The text lines are well separated and, in a few cases, we observe overlapping between ascenders and descenders from adjacent lines. The touching line problem only appears in some few cases. A sample of a handwritten document image of these datasets can be seen in Fig. 9a and 9b.

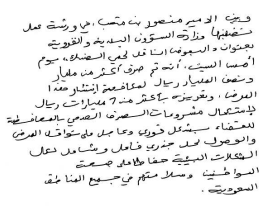
UMD. The UMD data set was collected by the members of the Language and Media Processing Laboratory at the University of Maryland. It consists in 123 Arabic documents, containing 1670 text lines. The images in this dataset present complex layouts and different



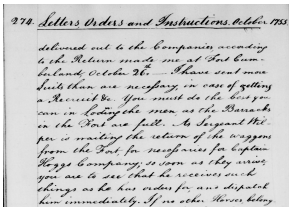
(a) ICDAR2009



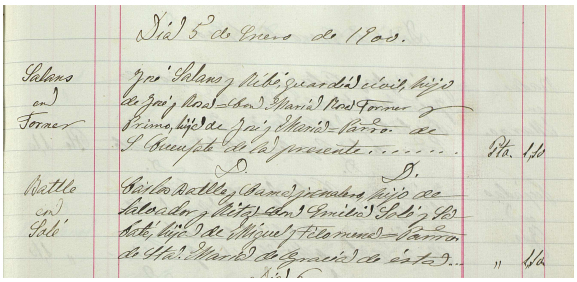
(b) ICDAR2013



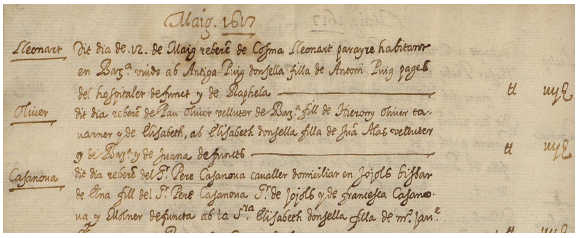
(c) UMD



(d) *George Washington*



(e) *Barcelona Marriages (19th century)*



(f) *Barcelona Marriages (17th century).*

Fig. 9: Samples of the datasets used in this work.

levels of noise and degradation. We can observe an example of this database in Fig. 9c.

George Washington. This dataset consists of 20 pages and 715 text lines, from the George Washington collection. It is sampled from different parts of the original collection (at the Library of Congress). The images are in grey-level and scanned from microfilms. There are two writers in the 20 selected document pages. It is a well-known real historical handwritten database and, although the documents present good quality, some of the typical difficulties appear in some cases. The handwriting style in the Washington dataset is roughly straight and horizontal, and it contains ascenders and descenders from adjacent lines which are touching each other. We can observe an example of this dataset in Fig. 9d.

Barcelona Marriages[2] It is a collection of books written from the 15th to the 19th century. It contains the marriage licenses celebrated in Barcelona and surroundings. There are approximately $90,500$ pages, written by 244 different writers. The documents of this collection are in colour and they are degraded by lifetime and frequent handling. We show two examples of these documents in Figs. 9e and 9f. Information extraction from these manuscripts is of key relevance for scholars in social sciences to study the demographical changes over the five centuries. This collection presents an old handwriting style with all the difficulties of a real historical handwritten database (see section 1): touching lines, large and big strokes with many overlapped characters between lines, horizontally-overlapping components and multi-skewed text lines. We have used two datasets in order to show the performance of our method with different handwriting styles. The first one consists of 30 documents with 964 text lines from the 19th century. The second one contains 94 documents with 3252 text lines from the 17th century.

4.3 Experiments and Results

We have performed five different experiments with the five datasets explained above, plus a synthetic dataset. Each experiment includes the results of the method presented in this work and the results of a classic method based in projections [46]. The objective of this comparison is to show the difference between localizing and segmenting text lines. We prove that, even our localization is coarse, we obtain a high accuracy in the text line segmentation. The parameters used in our method have been experimentally computed, but it is important to notice that we have used the same configuration for all the experiments. Therefore, we show how robust is the method to different collections using the same configuration. The values are $\alpha_1 = 0.61$, $\alpha_2 = 0.369$, $\alpha_3 = 0.001$ and $\alpha_4 = 0.20$.

ICDAR2009 & ICDAR2013 experiments. In the first experiment we have compared the results of our approach with the results of the participants of the ICDAR2009 and ICDAR2013 Handwritten Line Segmentation Contests and a classical line segmentation

[2] This database is available upon request to the authors of the paper.

method based in projections [46] as baseline. The performance obtained using the ICDAR2009 and the ICDAR2013 datasets are shown in Table 3 and 4 respectively.

First, to have a baseline reference, the performance of a classical algorithm based on projections has been measured. It obtains a FM of 85.86% using the ICDAR2009 and 76.51% using the ICDAR2013 database. This low performance is because projection-based methods have difficulties segmenting handwritten documents with skew or touching components, and do not work properly when ascenders and descenders of two consecutive lines are horizontally-overlapped.

To better assess the performance of our method regarding the state of the art, in Table 3 we can see the comparison with all the methods presented in the ICDAR2009 Handwritten Line Segmentation Contest. These results are reprinted from the contest report [13]. Here we focus on the analysis of the most outstanding methods. The *CUBS* method is based on an improved directional run-length analysis. The *ILSP-LWSeg-09* method uses a *Viberti* algorithm to segment lines. The *PAIS* method is based on horizontal projections. And the *CMM* method uses labels to identify words of the same line. For more details on the rest of the methods the reader is referred to Section 2.

The main problem of the *CUBS* method is the overlapping of adjacent text lines. The *ILSP-LWSeg-09* method has the problem with the function that minimizes the distance is that it is different depending on the document. The *PAIS* is a projection-based method, and this kind of methods have a problem in highly multi-skewed text lines. Finally, the *CMM* method is a grouping-based method, so it fails to distinguish touching text lines.

Table 4 shows the comparison with all the methods presented in the ICDAR2013 Handwritten Line Segmentation Contest. These results are reprinted from the contest report [54]. Here we focus again on the analysis of the most outstanding methods. The *INMC* method is based in an algorithm of energy minimization using the fitting errors and the distances between the text lines. The *NUS* method uses a seam carving algorithm to segment lines. The *GOLESTAN* method divides the document in regions to compute the text lines using a 2D Gaussian filter.The *CUBS* method is based in a connectivity mapping using directional run-length analysis. And the *IRISA* method combines blurred images and connected components to segment the lines.

The *INMC* method has as main problem that needs a learning process to estimate a cost function that imposes the constrains on the distances between text lines and the curvilinearity of each text line. The *GOLESTAN* method localize text lines, the segmentation is done obtaining the dilation of synthetic paths, which represents the localization of the text lines. Difficulties as touching lines and overlapping are not solved. The *IRISA* method localize properly the handwritten text lines, the problems of crossing lines and overlapping are considered, but the touching lines problem is omitted.

Our method has obtained a FM of 96.67% and a 95.43% ir respective databases. It is worth noticing that the performance of some methods, having a high performance, decreases when they are applied to other document collections, especially historical ones. An example is the *CUBS* method which in the ICDAR2009 got a

Table 3: Evaluation results using the ICDAR 2009 Database, where M is the count of result elements, *o2o* is the number of one-2-one matches, *DR* is the Detection Rate, *RA* is the Recognition Accuracy and *FM* is the harmonic mean. The number of ground-truth elements N is 4034.

	M	o2o	DR(%)	RA(%)	FM(%)
CUBS	4036	4016	99.55	99.50	99.53
ILSP-LWSeg-09	4043	4000	99.16	98.94	99.05
PAIS	4031	3973	98.49	98.56	98.52
CMM	4044	3975	98.54	98.29	98.42
CASIA-MSTSeg	4049	3867	95.86	95.51	95.68
PortoUniv	4028	3811	94.47	94.61	94.54
PPSL	4084	3792	94.00	92.85	93.42
LRDE	4423	3901	96.70	88.20	92.25
Jadavpur Univ	4075	3541	87.78	86.90	87.34
ETS	4033	3496	86.66	86.68	86.67
AegeanUniv	4054	3130	77.59	77.21	77.40
REGIM	4563	1629	40.38	35.70	37.90
Proposed	**4176**	**3971**	**98.40**	**95.00**	**96.67**
Base line (Project.)	4081	3834	86.36	86.37	85.86

Table 4: Evaluation results using the ICDAR 2013 Database. The number of ground-truth elements N is 2649.

	M	o2o	DR(%)	RA(%)	FM(%)
INMC	2650	2614	98.68	98.64	98.66
NUS	2645	2605	98.34	98.49	98.41
GOLESTAN-a & -b	2646	2602	98.23	98.34	98.28
CUBS	2677	2595	97.96	96.94	97.45
IRISA	2674	2592	97.85	96.93	97.39
TEI (SoA)	2675	2590	97.77	96.92	97.30
LRDE	2632	2598	96.94	97.57	97.25
ILSP (SoA)	2685	2546	96.11	94.82	95.46
QATAR-b	2609	2430	91.73	93.14	92.42
NCSR (SoA)	2646	2447	92.37	92.48	92.43
QATAR-a	2626	2404	90.75	91.55	91.15
MSHK	2696	2428	91.66	90.06	90.85
CVC[3]	2715	2418	91.28	89.06	90.16
Proposed	**2697**	**2551**	**96.30**	**94.58**	**95.43**
Base line (Project.)	2430	1836	77.50	75.55	76.51

[3] This method was a preliminary version of the approach presented in this paper. The method proposed in this paper increases the accuracy of the touching-components segmentation with an improved version of the heuristics.

Table 5: Evaluation results using several approaches in *UMD* Documents. The number of ground-truth elements N is 1951.

	M	o2o	DR(%)	RA(%)	FM(%)
Proposed	**2189**	**1814**	**92.97**	**82.86**	**87.63**
Base line (Project.)	2314	1270	65.09	54.88	59.55

Table 6: Evaluation results using several approaches in *George Washington* Documents. The number of ground-truth elements N is 715.

	M	o2o	DR(%)	RA(%)	FM(%)
Proposed	**693**	**653**	**91.30**	**94.20**	**92.70**
Base line (Project.)	727	338	47.20	46.40	46.70

Table 7: Evaluation results using several approaches in *Barcelona Marriages* Documents (19th century). The number of ground-truth elements N is 964.

	M	o2o	DR(%)	RA(%)	FM(%)
Proposed	**981**	**964**	**83.00**	**81.60**	**82.20**
Base line (Project.)	1276	630	65.30	49.30	56.10

Table 8: Evaluation results using several approaches in *Barcelona Marriages* Documents (17th century). The number of ground-truth elements N is 3252.

	M	o2o	DR(%)	RA(%)	FM(%)
Proposed	**1013**	**865**	**87.40**	**85.30**	**86.30**
Base line (Project.)	1064	651	66.40	61.10	63.60

FM of 99.53% and a 97.45% in the ICDAR2013. But nevertheless, the method presents robustness in front of different databases. In the following paragraphs we will show how our method is robust under different conditions, and how the performance keeps in a reasonable level even when the distortion is high in some other historical datasets.

The objective of the next experiments is to show that our method is addressed to segment lines accurately and not only the localization.

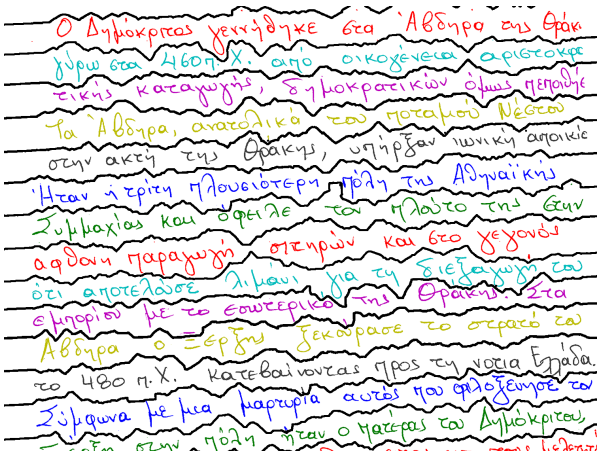
UMD experiment. In this experiment we have evaluated the performance of our method with respect the classical approach based in projections, using the UMD dataset. The performance obtained using the UMD dataset is shown in Table 5. Our approach obtained a FM of 87.63% and the baseline method based in projections obtained a FM of 59.55%. The performance of simple projection-profile analysis methods fails also in Arabic documents, even more if the present touching lines in their documents. The results of the UMD database can be compared with the results of the work [5]. They compare their work with several datasets of the literature and with different databases (included the UMD database). We observe low performance results (73.52%) using the UMD database. This method makes an accurate localization of the lines, but fails in the segmentation when the lines present touching components.

George Washington experiment. In this experiment we have evaluated the performance of our method with respect the classical approach based in projections, using the George Washington dataset. The performance obtained using the George Washington Dataset is shown in Table 6. Our approach obtained a FM of 92.70%, slightly lower than in the ICDAR2009 and in the ICDAR2013 dataset. The baseline method based in projections obtained a poor FM of 46.70%.

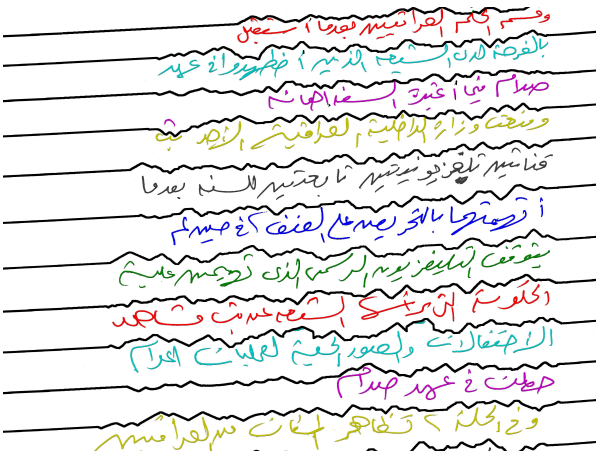
Barcelona Marriages experiment. In the last experiment, we have compared the performance of our method with the results obtained from the classical approach based in projections, using the two Barcelona Marriages datasets. Tables 7 and 8 show the results obtained in the datasets of the 19th and 17th century respectively. The results using this dataset are good taking into account the quality of the documents. We have obtained a *FM* of 82.20% using the dataset of 19th century and a FM of 86.3% with the dataset of 17th century. These FM rates are lower than the obtained using the George Washington dataset. We have also computed the FM rate using the classic method based in projections and the results are poor (56.10% and 63.60% respectively). As it was expected, the performance of simple projection-profile analysis methods, that are standard techniques in machine-printed documents, is very poor in historical manuscripts with variations in the script styles, lines that touch and overlap ones to the others, noise, etc.

In Fig. 10 the reader can observe some qualitative results obtained in the five databases. As we can observe the Barcelona Marriages datasets are more complex than the other two datasets, and sometimes touching components are not properly segmented. However, the problem of horizontally-overlapping components is solved in most of the cases (Fig. 10f). Our method can find a path through the ascenders and descenders of the text lines without any problem. We can observe that our method properly segments documents containing text lines of different length (Fig. 10d and 10e) and skew (Fig. 10a and 10c).

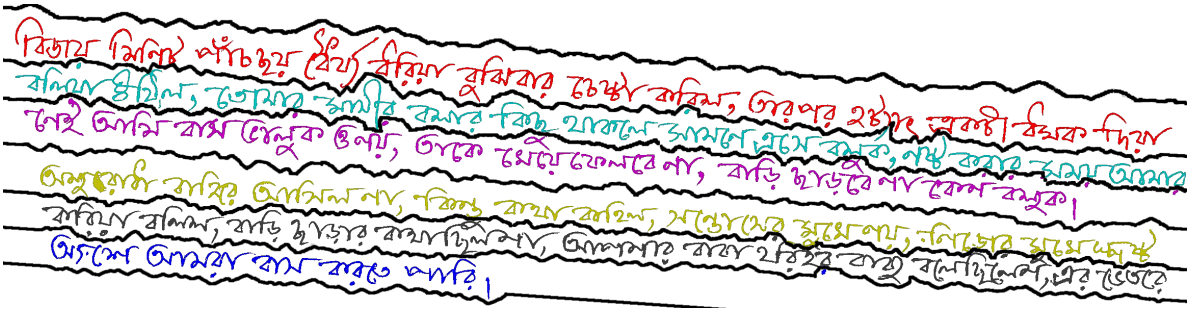
Other experiments. To evaluate the robustness of our method in front of the typical difficulties of handwritten documents, we have generated some synthetic



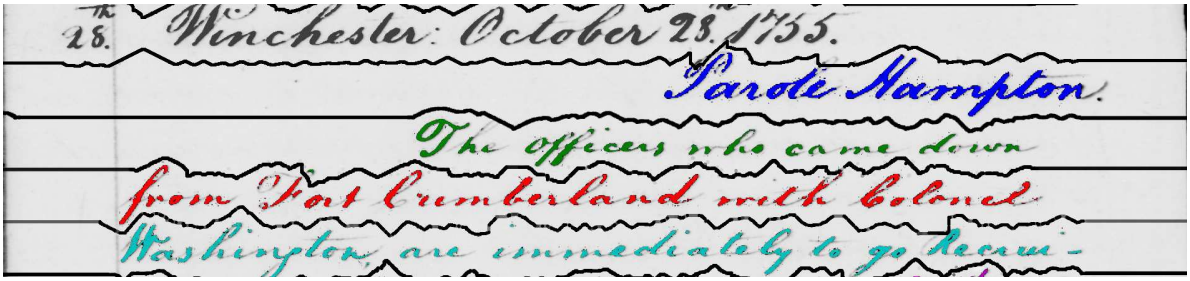
(a) ICDAR2009.



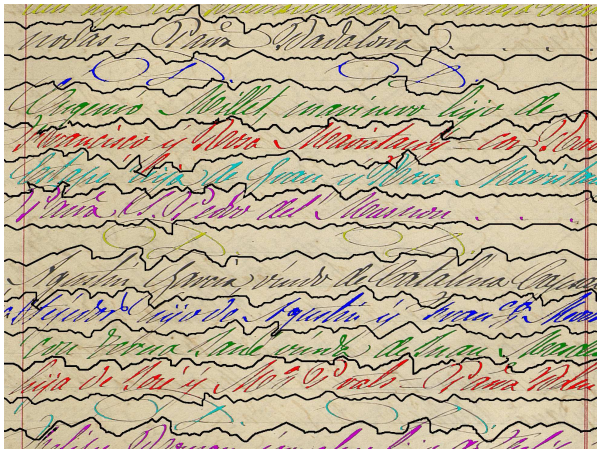
(b) *UMD.*



(c) *ICDAR2013.*



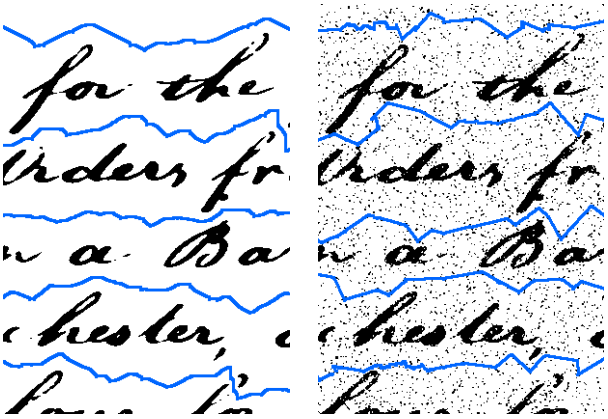
(d) *George Washington.*



(e) *Barcelona Marriages (19th century).*

(f) *Barcelona Marriages (17th century).*

Fig. 10: Qualitative results obtained using our approach.



(a) (b)

Fig. 12: The influence of noise in line segmentation. (a) Clean image. (b) Salt & pepper noise image.

(a) (b) (c) (d)

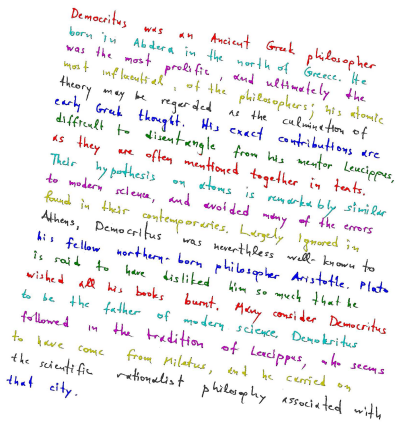
Fig. 13: The influence of noise in line segmentation: spots, blobs, graphical lines and show-through problems.

images (Fig. 11) from the ICDAR2009 dataset. The first difficulty appears when the document presents a high skew (Fig. 11a). We have rotated -5 degrees document. The second one appears when the line spaces between text lines are not homogeneous (Fig. 11b). We have spaced the lines using different sizes. Sometimes the text lines present a falling curvature (Fig. 11c), or text lines have different orientation between them (Fig. 11d). Finally, some documents present several text blocks, even in different orientations (Fig. 11e). These three images have been manually modified. As the objective of this work is the line segmentation in text blocks, and not the layout segmentation, we have used the approach developed by Cruz et al. [8] which is oriented to extract text blocks in historical documents. So, the segmentation method has been applied after segmenting each text block. In the same image we have introduced text with 0, 90 and 45 degrees. We can observe that in all these cases our approach properly segments the documents.

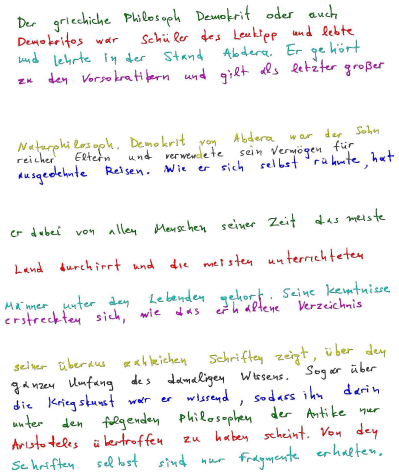
Our method is also able to cope with noisy documents as we can observe in Fig. 12. The presence of noise influences on the computation of the skeleton of the background image. However the variation in the skeleton does not influence on the segmentation of the text lines. We can observe the results obtained using a clean image in Fig. 12a and a salt-and-pepper noisy image in Fig. 12b. In this figure we have simulated the case when the document is too much noise and the preprocess cannot clean the document completely. The result of the pre-process is a noisy image, but nonetheless, the segmentation is done in a proper way.

Some documents contain spots and show-through that can be a problem in the segmentation process. As we can observe in Fig. 13 our method allows to solve this problem thanks to the pre-processing step. In Fig. 13a and 13c we observe some noise in the document (spots), which can slightly modify the path, as we can observe in the Fig. 13a, but, in both cases, our approach finds a path to segment the lines. There are some cases where the documents present drawings and graphical lines (Fig. 13b). Our method does not remove this kind of noise, but segments the lines searching a path between these graphical elements. The problem of show-through is showed in the Fig. 13d. The preprocess solves this problem and the paths are properly computed. For more complex cases we can use specific pre-processing methods to remove spots, graphical lines, drawings and show-through. We have removed all the steps to clean the image included in the pre-process in all the experiments presented above. We have only done the binarization of the images. The objective is to simulate the noisy documents where the pre-process cannot remove the noisy completely. We show the good performance of the approach presented in this paper in this kind of documents.

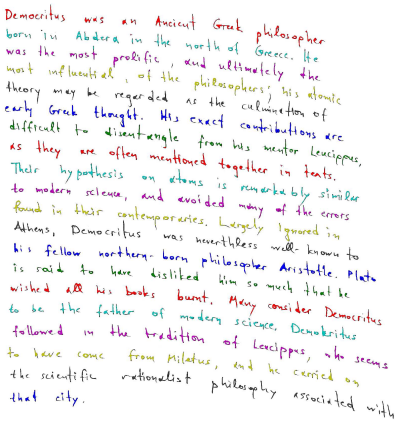
There are two main problems that can vary the number of initial nodes regarding the number of text lines (Fig. 14). The first problem appears when there are comments between the text lines. They are usually smaller than the size of the text lines and they are completely touching the above and below text lines (see Fig. 14a). In these cases the paths to segment the lines (upper, bottom and in-line text line) are almost over-



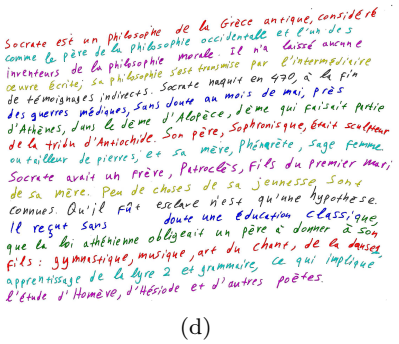
(a)



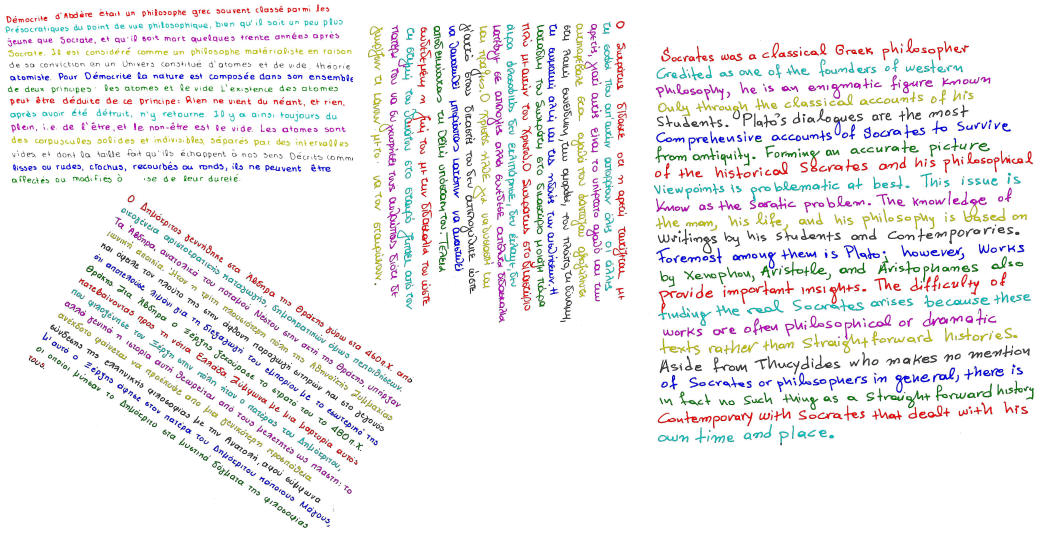
(b)



(c)

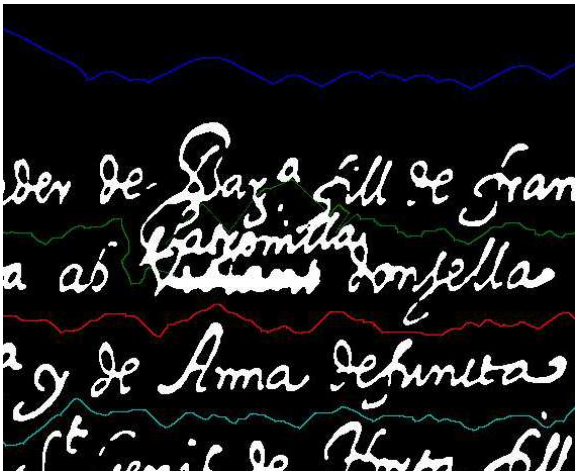


(d)

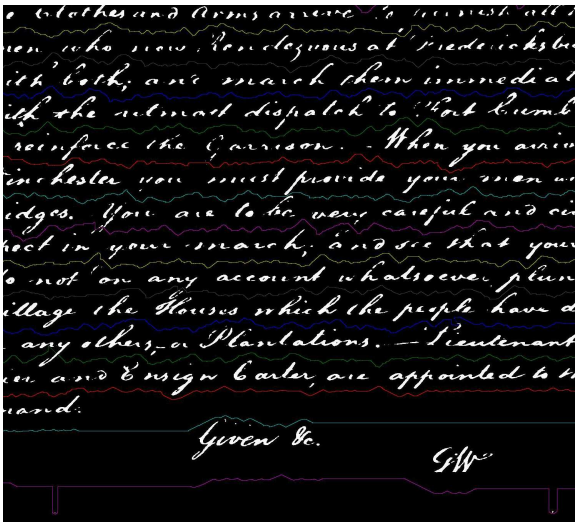


(e)

Fig. 11: Qualitative results over synthetic images generated from ICDAR2009 dataset. (a) Document with a pronounced skew. (b) Document with different line spacing between text lines. (c) Document where the text lines fall. (d) Document with multi-oriented text lines. (e) Document with several text blocks, each one with a different orientation.



(a) Text comments between the text lines.



(b) Short horizontally overlapped text lines.

Fig. 14: Difficulties that can vary the number of initial nodes.

lapped, and the consistency checking process will remove the most overlapped path. Consequently, it joins one of the text lines with the in-line text. The second problem appears when the length of the text line is very short (compared to the rest of the lines) and it is horizontally overlapped (Fig. 14b). Then the segmentation of the text lines becomes ambiguous because it is difficult to determine whether text fragments belong to the same text line or not. These two problems increase the difficulty of segmenting text lines and it can be objective of specific works. However, since the number of these specific cases is very low in the databases used in our experiments, this is not a critical issue.

Finally, the presence of non-text elements like graphical lines or drawings is not a handicap for our method because it does not really depend on a script or text

Fig. 15: Segmenting non-text elements.

like writing style, but only on the structure. In the Barcelona Marriages dataset it is usual to end text lines with an horizontal stroke, or to cross out wrong registers. Our approach is able to tackle with this difficulty as we can observe in Fig. 15.

4.4 Discussion

The configuration used in the experiments has been computed experimentally using the ICDAR2009 dataset. This configuration has been used in all the experiments independently of the dataset. This fact shows the robustness of our method in front of different types of handwritten documents, whether they are historical or modern.

We can observe the evolution of the performance of the methods using the five databases: the two first ones are datasets where the problem of touching, skew and horizontally-overlapping text lines is scarce (ICDAR-2009 and ICDAR2009 dataset). The second one is dataset written in Arabic. It is written uing a left justification and presents touching and overlapping in a few cases. The third one is a historical dataset with some noise introduced by the life-time, and with some of the problems explained before (George Washington dataset) The last one is a dataset with noisy and degraded documents, where the percentage of touching components and horizontally-overlapping is very high (Barcelona Marriage dataset).

The last experiment shows the robustness of our method in front of the typical difficulties in handwritten documents. The method is able to cope with noisy

documents and with documents that contain drawing lines or comment lines.

The overall conclusion is that a baseline method based on classical projection profile analysis does not work well with manuscripts. The main difficulties for this kind of methods were anticipated in the introduction: the more is the skew and degree of touching and overlapping between consecutive lines, the lower is the accuracy of the segmentation. In general, projection profiles detect the approximate position of the text lines, but do not allow an accurate segmentation. When analyzing other methods designed for handwritten line segmentation, the proposed approach is ranked in the top positions. The robustness of the method was proved when it was applied to databases with increasing levels of difficulty. Our method kept its performance over 80% of *FM*. Hence we can conclude that the proposed method is highly able to cope with the different types of problems that appear in historical documents, in particular with multi-oriented lines, overlapped components and touching lines. Conversely, as it is observed in [36], the winner method of the *ICDAR2009* Handwritten Segmentation Contest with a *FM* of 99.5%, drastically decreases the performance to a *FM* of 56.1% using low resolution and noisy handwritten documents.

5 Conclusions

In this paper we have presented a robust line segmentation approach, which segments lines in any kind of handwritten documents. It is not designed for a specific category of documents, coping with both historical and modern ones. The proposed approach finds the path which is located at the same distance in the area between two consecutive text lines. The skeleton of the background image is used to convert it to a graph. This graph is used to find the best path to segment text lines using a minimum cost path-search algorithm.

One of the key contributions is the ability of the method to segment lines pixel-wise and not only locate then, as some approaches in the literature.

We have tested the method in five databases with different difficulties: ICDAR2009, UCDAR2013, UMD, George Washington and Barcelona Marriages database. With this extensive experimentation, we have proved that our method is able to deal with different conditions of degradations and in front of modern and historical documents. Even in the worst scenario, such as the Barcelona Marriage datasets that present many difficulties, our approach obtains a FM of 82.20%, while other state of the art methods dramatically decrease their performance when the documents contain skewed or multi-oriented, touching and overlapping lines.

In summary, this paper has presented a robust method to segment handwritten lines that outperforms the state-of-art (Table 1) in historical documents. The method works in an unsupervised framework so it is writer invariant. It tackles with multi-skewed, touching and horizontally-overlapped lines.

Acknowledgements The authors thank the *CED-UAB* and the Cathedral of Barcelona for providing the images. This work has been partially supported by the Spanish projects TIN2011-24631 and TIN2012-37475-C02-02, by the EU project ERC-2010-AdG-20100407-269796.

References

1. Alaei, A., Nagabhushan, P., Pal, U.: Piece–wise painting technique for line segmentation of unconstrained handwritten text: a specific study with persian text documents. Pattern Analysis and Applications pp. 381–394 (2011)
2. Arivazhagan, M., Srinivasan, H., Srihari, S.: A statistical approach to line segmentation in handwritten documents. In: Document Recognition and Retrieval XIV SPIE, pp. 6500T–1–11 (2007)
3. Boykov, Y., Veksler, O.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1222–1239 (2001)
4. Bukhari, S., Shafait, F., Breuel, T.: Script-independent handwritten textlines segmentation using active contours. In: International Conference on Document Analysis and Recognition, pp. 446–450 (2009)
5. Bukhari, S., Shafait, F., Breuel, T.: Towards Generic Text-Line Extraction. International Conference on Document Analysis and Recognition pp. 748–752 (2013)
6. Bukhari, S.S., Breuel, T.M.: Layout analysis for arabic historical document images using machine learning. In: International Conference on Frontiers in Handwriting Recognition, pp. 635–640 (2012)
7. Cohen, E., Hull, J., Srihari, S.: Control structure for interpreting handwritten addresses. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1049–1055 (1994)
8. Cruz, F., Ramos, O.: Handwritten line detection via an em algorithm. International Conference on Document Analysis and Recognition (2013)
9. Dijkstra, E.: A note on two problems in connexion with graphs. Numerische mathematik pp. 269–271 (1959)
10. Dos Santos, R., Clemente, G.S.G., Ren, T.T.I., Cavalcanti, G.G.D., Santos, R.P.D.: Text line segmentation based on morphology and histogram projection. International Conference on Document Analysis and Recognition pp. 651–655 (2009)
11. Feldbach, M., Tonnies, K.: Line detection and segmentation in historical church registers. In: International Conference on Document Analysis and Recognition, pp. 743–747 (2001)
12. Fernández, D., Lladós, J., Fornés, A.: Handwritten word spotting in old manuscript images using a pseudo-structural descriptor organized in a hash structure. In: Pattern Recognition and Image Analysis, pp. 628–635 (2011)

13. Gatos, B., Stamatopoulos, N., Louloudis, G.: Icdar 2009 handwriting segmentation contest. International Conference on Document Analysis and Recognition pp. 1393–1397 (2009)
14. Ha, J., Haralick, R.M., Phillips, I.T.: Document page decomposition by the bounding-box project. In: International Conference on Document Analysis and Recognition, p. 1119 (1995)
15. Hart, P., Nilsson, N.: A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems, Science, and Cybernetics pp. 100–107 (1968)
16. Hull, J.: Document image skew detection: Survey and annotated bibliography. Series in Machine Perception and Artificial Intelligence pp. 40–66 (1998)
17. Jindal, M., Sharma, R., Lehal, G.: Segmentation of horizontally overlapping lines in printed indian scripts. In: International Journal of Computational Intelligence Research, pp. 277–286 (2007)
18. Kang, L., Doermann, D.: Template based Segmentation of Touching Components in Handwritten Text Lines. In: International Conference on Document Analysis and Recognition, pp. 569–573 (2011)
19. Kang, L., Kumar, J., Ye, P., Dermann, D.: Learning textline segmentation using codebooks and graph partitioning. In: International Conference on Frontiers in Handwriting Recognition (2012). 63-68
20. Kang, L., Kumar, J., Ye, P., Doermann, D.: Learning Text-line Segmentation using Codebooks and Graph Partitioning. In: International Conference on Frontiers in Handwriting Recognition, pp. 63–68 (2012)
21. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of computer vision pp. 321–331 (1988)
22. Kavallieratou, E., Dromazou, N., Fakotakis, N., Kokkinakis, G.: An integrated system for handwritten document image processing. International Journal of Pattern Recognition and AI pp. 617–636 (2003)
23. Kavallieratou, E., Fakotakis, N., Kokkinakis, G.K.: An unconstrained handwriting recognition system. International Journal on Document Analysis and Recognition pp. 226–242 (2002)
24. Kennard, D., Barrett, W.: Separating lines of text in free-form handwritten historical documents. In: Document Image Analysis for Libraries, pp. 12–23 (2006)
25. Koo, H., Cho, N.: Text-line extraction in handwritten chinese documents based on an energy minimization framework. Trans. Img. Proc. pp. 1169–1175 (2012)
26. Kornfield, E., Manmatha, R., Allan, J.: Text alignment with handwritten documents. In: Document Image Analysis for Libraries, pp. 195–209 (2004)
27. Kumar, J., Abd-Almageed, W., Kang, L., Doermann, D.: Handwritten arabic text line segmentation using affinity propagation. In: IAPR International Workshop on Document Analysis Systems, pp. 135–142 (2010)
28. Kumar, J., Kang, L., Doermann, D., Abd-Almageed, W.: Segmentation of Handwritten Textlines in Presence of Touching Components. International Conference on Document Analysis and Recognition pp. 109–113 (2011)
29. Kumar, K.S., Namboodiri, A.: Learning segmentation of documents with complex scripts. Indian conference on Computer Vision, Graphics and Image Processing pp. 749–760 (2006)
30. Li, Y., Zheng, Y., Doermann, D., Jaeger, S.: Script-independent text line segmentation in freestyle handwritten documents. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1313–1329 (2008)
31. Likforman-Sulem, L., Zahour, A., Taconet, B.: Text line segmentation of historical documents: a survey. International Journal of Document Analysis and Recognition pp. 123–138 (2006)
32. Liwicki, M., Indermuhle, E., Bunke, H.: On-line handwritten text line detection using dynamic programming. International Conference on Document Analysis and Recognition pp. 447–451 (2007)
33. Louloudis, G., Gatos, B., Pratikakis, I., Halatsis, C.: Text line detection in handwritten documents. Pattern Recognition pp. 3758–3772 (2008)
34. Manmatha, R., Rothfeder, J.: A scale space approach for automatically segmenting words from historical handwritten documents. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1212–1225 (2005)
35. Manmatha, R., Srimal, N.: Scale space technique for word segmentation in handwritten documents. In: Scale-Space Theories in Computer Vision, vol. 1682, pp. 22–33 (1999)
36. Manohar, V., Vitaladevuni, S., Cao, H., Prasad, R., Natarajan, P.: Graph clustering-based ensemble method for handwritten text line segmentation. In: International Conference on Document Analysis and Recognition, pp. 574–578 (2011)
37. Nicolaou, A., Gatos, B.: Handwritten text line segmentation by shredding text into its lines. In: International Conference on Document Analysis and Recognition, pp. 626–630 (2009)
38. O'Gorman, L.: The document spectrum for page layout analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1162–1173 (1993)
39. Otsu, N.: A Threshold Selection Method from Gray-level Histograms. IEEE Transactions on Systems, Man and Cybernetics pp. 62–66 (1979)
40. Ouwayed, N., Belaid, A.: A general approach for multi-oriented text line extraction of handwritten document. International Journal on Document Analysis and Recognition (2011)
41. Papavassiliou, V., Stafylakis, T., Katsouros, V., Carayannis, G.: Handwritten document image segmentation into text lines and words. Pattern Recognition pp. 369–377 (2010)
42. Rath, T., Manmatha, R.: Word image matching using dynamic time warping. IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp. 521–527 (2003)
43. Rath, T., Manmatha, R.: Word image matching using dynamic time warping. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. II–521 – II–527 vol.2 (2003)
44. Rath, T., Manmatha, R., Lavrenko, V.: A search engine for historical manuscript images. International conference on Research and development in information retrieval pp. 369–376 (2004)
45. Rath, T.M., Manmatha, R.: Word spotting for historical documents. International Journal On Document Analysis and Recognition pp. 139–152 (2007)
46. Rodríguez-Serrano, J.A., Perronnin, F.: Handwritten word-spotting using hidden markov models and universal vocabularies. Pattern Recognition pp. 2106–2116 (2009)
47. Rohini, S., Uma Devi, R., Mohanavel, S.: Segmentation of touching, overlapping, skewed and short handwritten text lines. International Journal of Computer Applications pp. 24–27 (2012)
48. Roy, P., Pal, U., Lladós, J.: Morphology based handwritten line segmentation using foreground and background information. In: International Conference on Frontiers in Handwriting Recognition, pp. 241–246 (2008)

49. Saabni, R., El-Sana, J.: Language-independent text lines extraction using seam carving. In: International Conference on Document Analysis and Recognition, pp. 563–568 (2011)
50. Sarkar, R., Moulik, S., Das, N., Basu, S., Nasipuri, M., Kundu, M.: Suppression of non-text components in handwritten document images. In: International Conference on Image Information Processing, pp. 1–7 (2011)
51. Shi, Z., Setlur, S., Govindaraju, V.: Text extraction from gray scale historical document images using adaptive local connectivity map. In: International Conference on Document Analysis and Recognition, pp. 794–798 (2005)
52. Shi, Z., Setlur, S., Govindaraju, V.: A steerable directional local profile technique for extraction of handwritten arabic text lines. In: International Conference on Document Analysis and Recognition, pp. 176–180 (2009)
53. Stafylakis, T., Papavassiliou, V., Katsouros, V., Carayannis, G.: Robust text-line and word segmentation for handwritten documents images. In: International Conference on Acoustics, Speech and Signal Processing, pp. 3393–3396 (2008)
54. Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., Alaei, A.: Icdar 2013 handwriting segmentation contest. International Conference on Document Analysis and Recognition pp. 1402–1406 (2013)
55. Wahlberg, F., Brun, A.: Graph based line segmentation on cluttered handwritten manuscripts. International Conference on Pattern Recognition pp. 1570–1573 (2012)
56. Yanikoglu, B.: Segmentation of off-line cursive handwriting using linear programming. Pattern Recognition pp. 1825–1833 (1998)
57. Yin, F.: Handwritten text line extraction based on minimum spanning tree clustering. International Conference on Wavelet Analysis and Pattern Recognition pp. 1123–1128 (2007)
58. Yin, F., Liu, C.: Handwritten text line segmentation by clustering with distance metric learning. International Conference on Frontiers in Handwriting Recognition pp. 229–234 (2008)
59. Yin, F., Liu, C.: Handwritten chinese text line segmentation by clustering with distance metric learning. Pattern Recognition pp. 3146–3157 (2009)
60. Zagoris, K., Pratikakis, I., Antonacopoulos, A., Gatos, B., Papamarkos, N.: Handwritten and machine printed text separation in document images using the bag of visual words paradigm. In: International Conference on Frontiers in Handwriting Recognition, pp. 103–108 (2012)