

Learning-free text line segmentation for historical handwritten documents

Berat Kurar Barakat · Rafi Cohen · Ahmad Drobay · Irina Rabaev ·
Jihad El-Sana

Received: date / Accepted: date

Abstract We present a learning-free method for text line segmentation of historical handwritten document images. This method relies on automatic scale selection together with second derivative of anisotropic Gaussian filters to detect the blob lines that strike through the text lines. Detected blob lines guide an energy minimization procedure to extract the text lines. Historical handwritten documents contain noise, heterogeneous text line heights, skews and touching characters among text lines. Automatic scale selection allows automatically adapt to the heterogeneous nature of handwritten text lines in case the character height range is correctly estimated. In the extraction phase, the method can accurately split the touching characters among the text lines. We provide results investigating various settings, and compare the model with recent learning-free and learning-based methods on the cBAD competition dataset.

Keywords Text line segmentation · Text line detection · Text line extraction · Learning-free · Historical handwritten documents

1 Introduction

Digital handwritten documents are not easily explorable in their raw form, but need to be transcribed further into machine readable text. Certainly, manual transcription of large number of documents is not feasible in a

First author
first address
E-mail: fauthor@example.com

S. Author
second address
E-mail: fauthor@example.com



Fig. 1 Examples of challenges in a recent historical handwritten text line detection dataset, cBAD.

reasonable time. Hence, there is a practical need for reliable handwritten document image processing algorithms. Text line segmentation is an essential operation and prerequisite for many document image analysis tasks. Advancement in text line segmentation performance will boost the performance of other tasks, such as word segmentation [28, 48] and word recognition [20, 27].

Text line segmentation consists of text line detection and text line extraction. Text line detection locates each text line by its baseline or x -height representation. Text line extraction in turn leads to polygonal or pixel level representation of text lines. Extraction level representation is more precise and useful for higher level document image analysis tasks. With the advances in deep learning, numerous learning-based methods have been proposed for text line segmentation of handwritten documents. Learning-based methods [38, 21, 34, 25] can inherently handle the problems arising from complex layout of text lines and heterogeneity of documents. Henceforth, the recent competition datasets [13] (Fig. 1) are more challenging than the prior ones [19, 46, 3, 14].

In the last decade several learning-free methods for text line extraction of handwritten documents have been proposed [6, 10, 41, 11]. However, we observe only limited attempts where learning-free algorithms have been used for text line detection of challenging historical documents [13]. This is particularly interesting as learning-based algorithms require vast amount of labeling effort.

A learning-free algorithm for text line extraction is proposed in [11] where text lines are detected using multi-scale second derivative of Gaussian filters and extracted using an energy minimization (EM) function. The present article extends this approach for detecting text lines of challenging historical documents. The new method proposes a robust character height estimation even with the presence of noise and chains of touching characters among multiple text lines. In addition, the extraction phase includes a mechanism to split the consequently touching text lines. Apart from these, we mathematically formulate automatic scale selection together with second derivative of anisotropic Gaussian and thoroughly present and discuss the evaluation of parameters.

2 Related Work

Text line extraction approaches can be classified into three categories: top-down, bottom-up, and hybrid. Top-down approaches partition document images into text lines based on global features. Bottom-up approaches group pixels or components based on local features to form text lines. Hybrid approaches combine top-down and bottom-up techniques.

2.1 Top-down approaches

Top-down approaches are mainly based on projection profile, Hough transform, smearing or seam carving. Projection profile sums pixel values along the horizontal axis for each y axis value of a binary image to determine locations of text lines.

Projection profile is commonly used for simple document images [2] but can also be adapted for gray scale images [22] and slightly skewed lines [36].

Hough transform calculates parameters of linear structures produced by the text components in the document image. [43] used Hough transform to find the global orientation of a document image and apply projection profile along this orientation. Hough transform can also be applied to the centroids of connected components and directly align them as text lines [18]. Hough transform is robust for dealing with skewed straight lines, but entails high computational cost.

Smearing fills the white space between the consecutive black pixels along the same direction if their distance is within a predefined threshold [50]. Smearing is sensitive to overlapping text strokes, therefore, [44] smeared background pixels running through the overlapping strokes to build line separators. Later on, [1] adapted smearing to skewed lines by applying it in a strip-wise fashion. Main difficulty of smearing is determining the optimum threshold and is dealt in [47] by using steerable directional filters.

Seam carving computes the path of minimum energy cost from one end of the image to another. [42] employed medial seams for text line extraction of binary document images using signed distance transform as the energy map. Subsequently, they improved the method for gray scale documents using geodesic distance transform as energy map [41].

2.2 Bottom-up approaches

Top-down approaches process the document images at global level, which is problematic when the document does not have a Manhattan layout. Therefore, bottom-up approaches process document images at local level and does not assume straight lines. They group the elements into text lines. Elements can be pixels, super-pixels or connected components. The counter-part of this is isolation of local elements, which is complicated for touching components across consecutive lines. Bottom-up approaches are mainly based on clustering or classification.

Clustering algorithms group elements according to their features in an unsupervised manner [51]. They can be applied both to the binarized document images [40, 14] as well as to the gray scale document images [17, 16]. The above mentioned works clustered super pixels into words and locally join them to form the text lines. Recently, [21] clustered super pixels into text lines in a greedy manner. As a result, this approach is applicable to different datasets without parameter tuning. Clustering algorithms are suitable for heterogeneous document collections; however, the number of clusters has to be selected sensitively.

Classification algorithms classify the elements according to their features in a supervised manner. They are robust to noisy and transformed images, but require a large amount of annotated data for training. Early methods used non-convolutional classifiers with hand crafted features [3, 29, 7]. Recent methods are inspired by convolutional neural networks, which have proven to be efficient. [31] used recurrent neural network to segment isolated paragraphs. [37] used convolutional neural network, first to classify page pixels as paragraph,

and then to classify paragraph pixels as text line or non text line. Text line extraction from full page is studied as a problem of predicting the bounding box around the text lines [32, 30]. Sequential nature of these methods limit them to be used by sliding windows on horizontal text lines. Pixel classification in a sliding window fashion is not desirable due to redundant and expensive computation of overlapping areas in the sliding windows. As a remedy, dense prediction has been successfully used for text line segmentation of handwritten documents [38, 25].

2.3 Hybrid approaches

Hybrid approaches combine the strengths of top-down and bottom-up approaches while reducing the weaknesses of each one. [15] calculates the starting point and the skew of text lines with projection profile, and then uses them to search a piecewise linear separating path. [23] globally estimates coarse text lines and locally re-assigns misclassified elements to split the touching text lines. [9] focuses on solving skewed and touching text lines using a combination of clustering connected components and projection profile analysis.

3 Scientific Background

In this section we present notations and definitions of scale-space representation with automatic scale selection, component tree and energy minimization via graph cuts.

3.1 Scale-space representation with automatic scale selection

The notion of scale is important when processing unknown image structures by automatic methods. This problem can be approached by representing image structures at different scales, so called scale-space representation [49]. However, scale-space representation does not address the problem of how to select local appropriate scales. Therefore, we use automatic scale selection [26] that adapts to the local scales of image structures.

Scale-space representation together with automatic scale selection apply to a large class of differential image descriptors. We adapt this for detecting blobs, regions brighter than their surround, from text lines and term this procedure as blob line detection with automatic scale selection.

Given n scale values $(\sigma_1, \sigma_2, \dots, \sigma_n)$, scale-space representation is a stack of n layers where the i th layer is

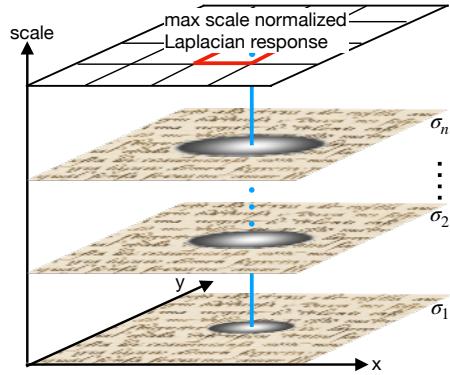


Fig. 2 Scale-space representation with automatic scale selection is achieved by convolving the image with different scales ($\sigma_1, \sigma_2, \dots, \sigma_n$) and assigning the maximum scale-normalized convolution response to each pixel of the image.

the convolution of image by Laplacian with the scale of σ_i (Fig. 2). Spatially, the Laplacian response will be maximum at the center of blob when the scale is matched with the scale of the text line. However, Laplacian response decreases as the scale σ increases. To eliminate this decay, Laplacian response is multiplied by σ^2 and is called scale-normalized Laplacian. Then, automatic scale selection assigns the maximum scale-normalized convolution response to each pixel of the image.

Normally, the differential descriptor for forming a blob is the Laplacian of an isotropic Gaussian (LoG) [26]. However, we define a scale-space representation using Laplacian of anisotropic Gaussians, because a text line is elongated and has different scales along the two coordinate directions. In the following sections we define the mathematical formulas for scale-space of Laplacian of an anisotropic Gaussians and automatic scale selection using scale-normalized Laplacian.

3.1.1 Scale-space of anisotropic Gaussians

Given a continuous image $I : \mathbb{R}^2 \rightarrow \mathbb{R}$, its scale-space of anisotropic Gaussians $L : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$ is defined by following convolution:

$$L(x, y; \sigma_x, \sigma_y) = I(x, y) * g(x, y; \sigma_x, \sigma_y), \quad (1)$$

where $g : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$ denotes the anisotropic Gaussian kernel

$$g(x, y; \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2}\right)}. \quad (2)$$

In this representation $*$ is the convolution operator, σ_x is the scale parameter in horizontal direction, and σ_y is the scale parameter in vertical direction.

3.1.2 Scale-space of Laplacian of anisotropic Gaussians

Scale-space of Laplacian of anisotropic Gaussians is computed by differentiating the scale-space of anisotropic Gaussians with respect to x and y two times

$$L_{x^2y^2}(x, y; \sigma_x, \sigma_y) = \partial_{x^2y^2} L(x, y; \sigma_x, \sigma_y), \quad (3)$$

or, equivalently, by convolving the image with Laplacian of anisotropic Gaussians

$$L_{x^2y^2}(x, y; \sigma_x, \sigma_y) = I(x, y) * [g_{x^2y^2}(x, y; \sigma_x, \sigma_y)]. \quad (4)$$

Given the anisotropic Gaussian in Eq. 2, its Laplacian is defined by

$$g_{x^2} = g_{x^2} + g_{y^2}, \quad (5)$$

where

$$g_{x^2} = \left(\frac{x^2 - \sigma_x^2}{\sigma_x^4} \right) \cdot g(x, y; \sigma_x, \sigma_y), \quad (6)$$

and

$$g_{y^2} = \left(\frac{y^2 - \sigma_y^2}{\sigma_y^4} \right) \cdot g(x, y; \sigma_x, \sigma_y). \quad (7)$$

3.1.3 Automatic scale selection

In scale-space representation the amplitude of the Laplacian in Eq. 3 decreases with scale. Based on this phenomena, automatic scale selection states that local extreme over scales of scale-normalized Laplacian

$$L_{x^2y^2}^{\sigma-norm} = (\sigma_x \sigma_y) \cdot L_{x^2y^2}(x, y; \sigma_x, \sigma_y) \quad (8)$$

corresponds to the significant structures that are regions brighter than their surround.

3.2 Component tree

Component tree [33] organizes the connected components of level sets in a tree structure. Let \mathcal{C}_t be the set of connected components obtained by thresholding with threshold t . The nodes in a component tree correspond to the components in \mathcal{C}_t for varying values of the threshold t . The root of the tree is the member of $\mathcal{C}_{t_{min}}$, where t_{min} is chosen such that $|\mathcal{C}_{t_{min}}| = 1$. Level ℓ in the tree correspond to $\mathcal{C}_{t_{min}+\ell d}$, where d is a parameter that determines the step size for the tree. We use $d = 1$ in all the experiments. There is an edge between $C_i \in \mathcal{C}_t$ and $C_j \in \mathcal{C}_{t+1}$ if and only if $C_j \subseteq C_i$. The maximal threshold t_{max} used in the tree construction is simply the maximal value in the map. Fig. 3 illustrates the above definitions.

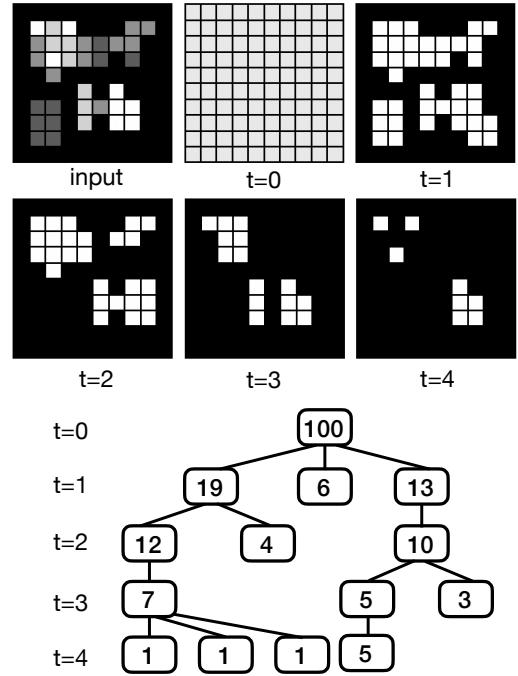


Fig. 3 A gray-level input and its successive threshold sets from $t = 0$ to $t = 4$, where $d = 1$. Below is the component tree with each node shows the size of its connected component.

4 Method

The proposed approach utilizes scale-space representation with automatic scale selection to detect blob lines that strike-through the text lines. The detected blob lines are then binarized by component tree algorithm. Finally, energy minimization via graph cuts extracts the text lines with the help of the detected blob lines.

4.1 Blob line detection using automatic scale selection

We informally define a blob to be a connected region that is significantly brighter than its neighbourhood. Text line detection aims to derive the blob lines that strike-through the text lines. These blob lines are derived by convolving text lines with the Laplacian of anisotropic Gaussians in a range of scales corresponding to the height range of the characters (Eq. 5).

Character height range σ_x is computed automatically using either way: 1) Component Evolution Map (CEM) [5] estimates character height range by analyzing the height distribution of connected components for each possible grayscale threshold; 2) Mean height of components estimates character height range as $[\mu, (\mu + \sigma)]$, where μ and σ are the average and standard deviation of components' heights in the document.

Rotation angle	0°	2°	4°	6°	8°	10°
Rotated image						
Grayscale blobs						
Binary blobs						

Fig. 4 The effectiveness of horizontally elongated filters on text lines with varying inclination. The visual results show that as the inclination increases the effectiveness of the method decreases.

These Gaussian filters are elongated along the horizontal direction, as such for every value of σ_x , $\sigma_y = e \times \sigma_x$, where e is the elongation rate and its optimal value is experimentally determined in Section 6.2.6. Then, for each pixel we chose the strongest response along the scale-normalized Laplacian given by Eq. 8. We investigate the effectiveness of horizontally elongated filters on text lines with varying inclination. The results suggest that this approach appears to be effective in detecting almost horizontal text lines (Fig. 4).

4.2 Blob line binarization

The blob lines detected by automatic scale selection are represented by a grayscale image as illustrated in Fig. 5(a). We investigate component tree algorithm to gather the binary blob lines that strike through the text lines.

4.2.1 Component tree

Component tree [33] binarizes the grayscale blob lines image by fitting k knots linear splines of least squares to each of the candidate blob line. A candidate blob line is labeled valid if it satisfies the both fitting scores, FS1 and FS2. FS1 requires the average 1-norm of each blob pixel from the spline to be less than the maximum character height. This condition eliminates the fat blobs that include two consecutive lines. FS2 requires the ratio of the blob area and the sum of the distances of the contour pixels from the spline to be less than 0.9. This

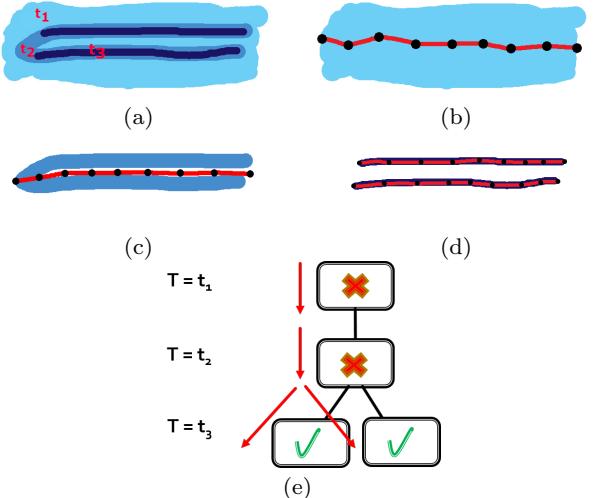


Fig. 5 Illustration of thresholding on a synthetic component. (a) Component is composed of three gray levels with ascending values t_1, t_2, t_3 . (b-d) Child components from thresholding the parent component with $T \geq t_1, t_2, t_3$ respectively and their approximating splines. (e) Breadth first traversal of the component tree.

condition eliminates the non-convex blobs with a partial merge (Fig. 5c). We traverse component tree in a breadth first search manner with $d = 1$. At each node, if the connected component is valid, it is taken as a binary blob line, and the search along this branch is complete. Otherwise, component is refined by recursively processing the children of the node. Fig. 5 illustrates component thresholding procedure which stops at valid blobs.

4.3 Text line extraction with energy minimization using graph cuts

Binary blob lines that are detected in Section 4.2 correspond to the text line detection phase. Extraction level representation requires further assigning a text line label to each pixel in the document image. We use energy minimization [12] for assigning connected components to text line labels with the help of detected blob lines. It urges to assign components to the label of the closest blob line, while straining to assign closer components to the same label, and not to assign any component to spurious blob lines. Let \mathcal{L} be the set of binary blob lines, and \mathcal{C} be the set of connected components in the binary document image. Energy minimization finds a labeling f that assigns each component $c \in \mathcal{C}$ to a label $l_c \in \mathcal{L}$, where energy function $E(f)$ has the minimum.

$$E(f) = \sum_{c \in \mathcal{C}} D(c, \ell_c) + \sum_{\{c, c'\} \in \mathcal{N}} d(c, c') \cdot \delta(\ell_c \neq \ell_{c'}) + \sum_{\ell \in \mathcal{L}} h_\ell \quad (9)$$

Energy function has three terms:

1. Data cost is the cost of assigning component c to label ℓ_c . For every $c \in \mathcal{C}$, $D(c, \ell_c)$ is defined as the Euclidean distance between the centroid of c and the blob line ℓ_c . Closer the component to a blob line, higher the chance the component to be assigned to the label of this blob line.
2. Smoothness cost is the cost of assigning closer components to different labels. Let N be the nearest component pairs. For every $\{c, c'\} \in \mathcal{N}$, $d(c, c') = \exp(-\alpha \cdot d_e(c, c'))$ where $d_e(c, c')$ is the Euclidean distance between the centroids of the components c and c' . $\alpha = (2 \langle d_e(c, c') \rangle)^{-1}$ where $\langle \cdot \rangle$ denotes expectation over all pairs of adjacent elements [39]. $\delta(\ell_c \neq \ell_{c'})$ is 1 if the condition inside the parentheses holds, and 0 otherwise.
3. Label cost For every blob line $\ell \in \mathcal{L}$, h_ℓ is defined as $\exp(2 \cdot r_\ell)$ where r_ℓ is the normalized number of foreground pixels overlapping with blob line ℓ . The higher the label cost, the higher the probability to discard the blob line as spurious line.

4.4 Merging broken blob lines

Once the EM removes the spurious blob lines, there may still exist the problem of broken blob lines. Given the binary image of blob lines, for each blob line we extract its left and right endpoints and define its direction as the vector connecting the left endpoint to the right endpoint. Two adjacent blob lines are merged if 1) the

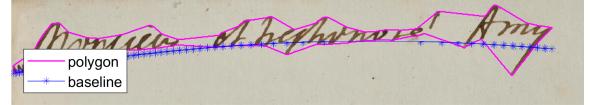


Fig. 6 Baseline extraction from bounding polygon.

direction of the vector connecting the right of the first component to the left of the second one falls between the direction of the two blob lines, 2) their vertical distance is less than the maximum character height.

4.5 Splitting touching characters

Text line extraction results contain unsegmented touching characters from adjacent text lines. To check whether a connected component c overlaps more than one blob line, we relabel c by assigning each pixel in c to the label of nearest blob line.

5 Evaluation

This paper proposes a learning-free text line segmentation method for challenging historical handwritten documents as such cBAD dataset. In addition, we also evaluated the method on another recent handwritten text line segmentation dataset, DIVA-HisDB. For each dataset we use only its test set because our method is not learning based. The proposed method's output is defined as pixel labels but it is manipulated according to the ground truth definition of each dataset.

5.1 cBAD dataset

cBAD dataset [13] contains 539 document images from 7 different archives. This dataset's ground truth is defined as baselines whereas our method extracts pixel labels. To find baselines from pixel labels, first we get tight polygons around the pixel labels of text lines. Then, for each bounding polygon we get its lower contour points and iteratively fit a regression line among these points by excluding the outliers (Fig. 6). Using the extracted baselines, the performance is measured by means of Precision (P), Recall (R) and F-measure (FM), as described in [13].

5.2 DIVA-HisDB dataset

DIVA-HisDB dataset [45] contains 30 pages from 3 medieval manuscripts. Ground truth is defined as bounding polygons. Therefore, we get the tight polygons around

the pixel labels of our output and measure the performance by means of the Intersection over Union (IU) as described in [45].

6 Experiments

6.1 Experimental setting

In this section we do ablation studies for adapting the learning-free algorithm proposed in [11] to challenging historical handwritten documents. Default parameters of the learning-free algorithm proposed in [11] are like the following:

1. Estimating character height range using CEM as described in Section 4.1
2. Blob line binarization using component tree with both of the fitting scores given in Section 4.2.1 and using number of knots $k = 20$.
3. Merging broken blob lines as described in Section 4.4
4. Binarizing the document image using Otsu thresholding, to be used with the EM function.

cBAD dataset is a relatively large dataset and the proposed algorithm takes around one day to run on the whole dataset. For this reason we randomly sample a representative subset of cBAD dataset, one page from each collection, and do the ablation studies on this subset.

6.2 Text line detection experiments

Text line detection stage is based on automatic scale selection by convolving the document image with second derivative of anisotropic Gaussian filters with a range of scales estimated as the character height range of the text in the document. Output of this filtering is, a response map, a grayscale image of the detected blobs. This grayscale image is binarized to gather the binary blob lines that strike-through the text lines.

6.2.1 Effect of character height range

Character height range that is used with the automatic scale selection inevitably effects the detected blob lines. [11] uses CEM for character height range estimation, but plenty number of touching text lines mislead the algorithm. To tackle this problem, we experiment with mean height of connected components (Section 4.1). Recognizing that the noise and long chain of touching text lines also misleading, we also use refined mean estimation by excluding the components that are bigger than a maximum threshold (100 pixels) and smaller

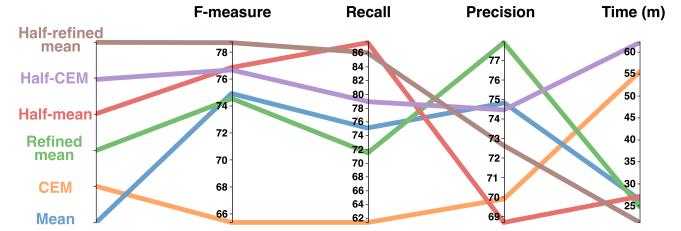


Fig. 7 Effect of height estimation method on the performance

Table 1 Effect of merging blob lines.

	Precision	Recall	F-measure	Time (m)
Merged	72.63	85.97	78.74	21.20
Non-merged	70.48	85.22	77.16	20.90

than a minimum threshold (10 pixels). Considering that the text is mostly located in the center region of a document, we also exclude the outer 2% of the document image. We found the results are less than ideal and further experiment with half of the above range estimations.

Fig. 8 shows that there are no sharp changes between the results of different height estimations. This is valid for such sample document without severe noise. As can be seen the half values of range values provide thinner blob lines that are more apart while provoking spurious blob lines. Qualitative analysis revealed a considerably higher performance and lower run time of the half-refined mean estimation for character height range (Fig. 7).

6.2.2 Effect of merging blob lines

In case of a big gap between words, a blob line can be disconnected. In this experiment we tested to which extent merging the blob line improves the results. We found that merging slightly increases the effectiveness (Table 1).

6.2.3 Effect of blob line fitting scores

The quality of detected blob lines is a key factor that influences the final performance. A blob line is preferable as plain, continuous and apart as possible. [11] defines this preferability by two fitting scores (4.2.1). The results using only the FS1 provide compelling evidence that non-convex blobs with a partial merge are negligible (Table 2).

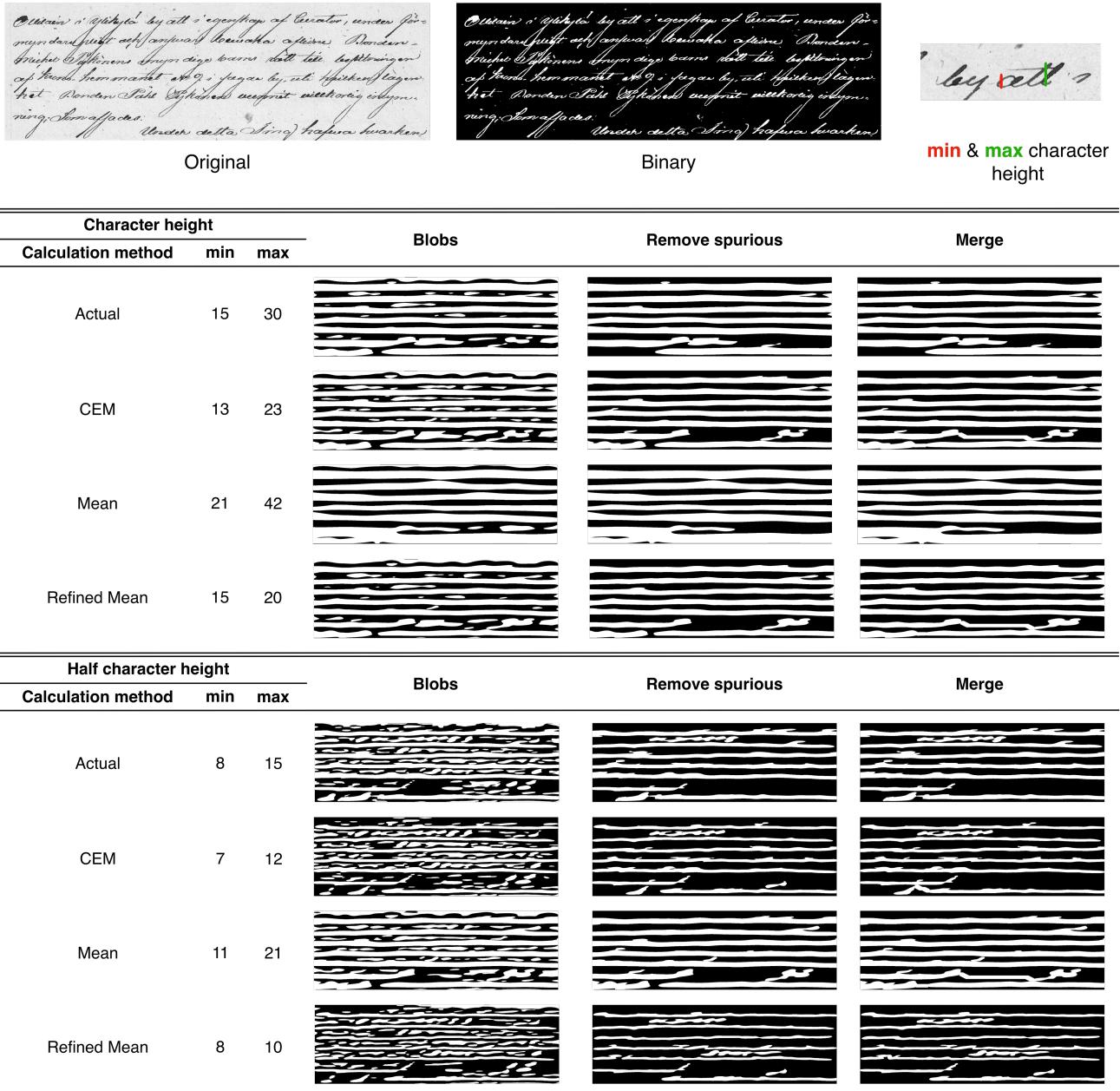


Fig. 8 Visualization of the effect of character height estimation method. Apparently the half values of range values provide thinner blob lines that are more apart while provoking spurious blob lines.

Table 2 Effect of using only the FS1.

Fitting score	Precision	Recall	F-measure	Time (m)
FS1 and FS2	72.63	85.97	78.74	21.20
FS1	74.66	85.50	79.71	19.96

6.2.4 Effect of fitting score threshold (t)

The results in Section 6.2.3 indicate that using only the FS1 is reasonable. FS1 eliminates the fat blobs that

include two consecutive lines. [11] conditions the FS1 to be less than the upper character height. Relaxing this condition decreases the run time whereas stressing it increases the run time without an evident performance gain (Fig. 9).

6.2.5 Effect of number of knots (k)

The aim of using splines is their ability to fit on curved blob lines as well as straight blob lines. [11] uses 20

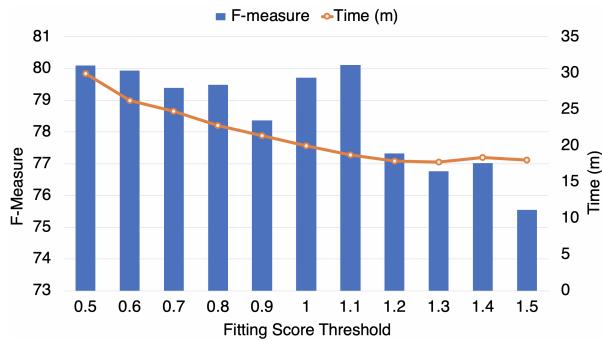


Fig. 9 Effect of fitting score threshold on performance and time.

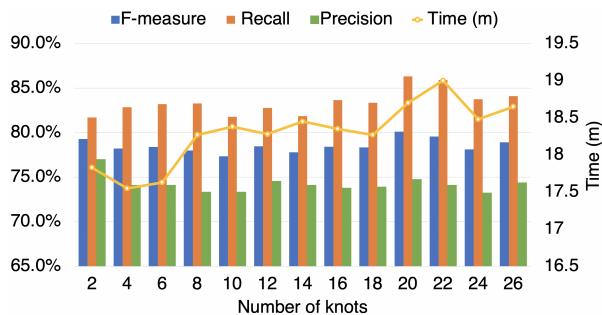


Fig. 10 Effect of number of knots (k) on performance and time.

knots to fit the linear splines. We investigate different number of knots and show their effect on the method's performance. Results are presented in Fig. 10. As expected, varying the number of knots does not impact the performance measure, because almost all text lines in the evaluation datasets are horizontal. However, the spline fitting makes the proposed method suitable for curved text lines ([24]). It is apparent that recall values follow a higher trend than the precision values, possibly due to the spurious blob lines that could not be eliminated by the label cost because of crowded ascenders and descenders overlapping these spurious lines.

6.2.6 Effect of elongation rate (e)

The proposed method assumes the text lines are almost horizontal. Therefore, convolution of a text line with the second derivative of an anisotropic Gaussian elongated along the horizontal direction generates a blob line that strike-through the text line. Figure 11 shows the blob lines resultant from two marginal cases: $e = 1$ and $e = 7$. There is a trade-off between the marginal cases. As e decreases as the blob lines are tight-fitting, as e increases as the blob lines are loose-fitting. Tight-fitting misses the spatial context and may consider two touching text lines as one (Fig. 11(b)). Loose-fitting grasps the unrelated spatial context and may generate spurious blob lines (Fig. 11(f)). From the performed ex-

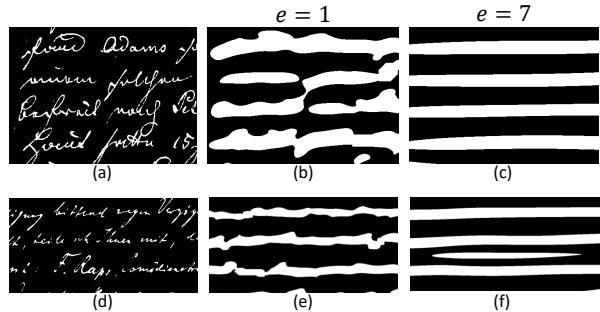


Fig. 11 Effect of elongation rate on blob lines.

Table 3 Effect of elongation rate on performance and time.

e	Precision	Recall	F-measure	Time (m)
1	75.38	82.09	78.59	22.96
2	75.71	84.64	79.93	21.11
3	74.19	85.62	79.49	22.80
4	71.92	87.31	78.87	23.30
5	72.31	86.97	78.96	23.05
6	71.09	86.30	77.96	25.83
7	71.44	86.18	78.12	29.76

periments, the optimal point seems to be when $e = 2$ (Table 3).

6.2.7 Conclusion about text line detection experiments

We have demonstrated that alternative values of parameters for text line detection produce slightly different results, with the exception of the character height estimation method. Half values of the refined-mean of the connected components achieves a superior performance over the default character estimation method, CEM.

6.3 Text line extraction experiments

In all the text line extraction experiments, we use half values of refined-mean for character height estimation. FS1 with a threshold of $t = 1.1$, merge the binarized blob lines. We set the number of knots to $k = 20$ and elongation rate to $e = 2$.

The energy function is formulated in the binary image domain, thereby, any noise pixel remained from the binarization process breaks down the accuracy. Therefore, we examined the effect of two binarization methods, Otsu [35] and Bar-Yosef *et al.* [4]. Both binarization results suffer either from losing parts of text lines or from noise parts of text lines. However, the performance is not sensitive to the binarization method (Table 4). On the other hand, splitting the touching characters among two text lines lead to a significant improvement (Table 4).

Table 4 Effect of using different binarization methods, Otsu and Bar-Yosef *et al.*, together with the effect of splitting the touching characters among text lines.

	Precision	Recall	F-measure	Time (m)
Non-split	75.71	84.64	79.93	21.11
Split and Otsu	81.74	92.25	86.68	23.95
Split and Bar-Yosef <i>et al.</i>	84.17	88.68	86.45	24.31

7 Results

We present results on two recent historical handwritten datasets, DIVA-HisDB and cBAD. For text line detection, we use half of refined-mean for character height estimation, FS1 with a threshold of $t = 1.1$, merge the binarized blob lines, and set the number of knots to $k = 20$ and elongation rate to $e = 2$. For text line extraction, we use Bar-Yosef *et al.* binarization and split the touching characters among text lines.

DIVA-HisDB is a historical document dataset with many number of consequently touching text lines and heterogeneous in terms of text line height. The input to our algorithm is the ground truth of layout analysis task of the competition. Table 5 shows our results compared to the best performing algorithm (System-9) in the competition which inputs the output of layout analysis. System-9 is also a learning-free algorithm. The proposed algorithm can reach a line IU of 100%. The degradation in pixel IU mostly comes from the coarse splitting of touching characters among text lines.

cBAD is a historical document dataset with a larger evaluation set and contains documents with varying layouts, originating from different time periods and locations. It is heterogeneous in terms of text line height and length. Input documents are gray-scale images. Some examples of pages with extracted bounding polygons can be seen in Fig. 12. Common error cases arise from group of touching characters that mislead the algorithm to merge the text lines (Fig. 12(b)). Another type of error occur due to the artifacts in the frame of the page images. We compare our results with the results obtained in cBAD competition. Table 6 shows precision, recall, and F-measure values of the participants. The methods are grouped as learning-based and learning-free. The best performance with an F-measure value of 97.70 is for a learning-based method, DMRZ. Two learning-free methods, our method and IRISA, achieve close F-measure values. The precision and recall values indicate that the IRISA method splits baselines more precisely but also misses more baselines compared to our method.

Finally we present the results obtained during the RASM2018 competition for historical scientific manuscripts

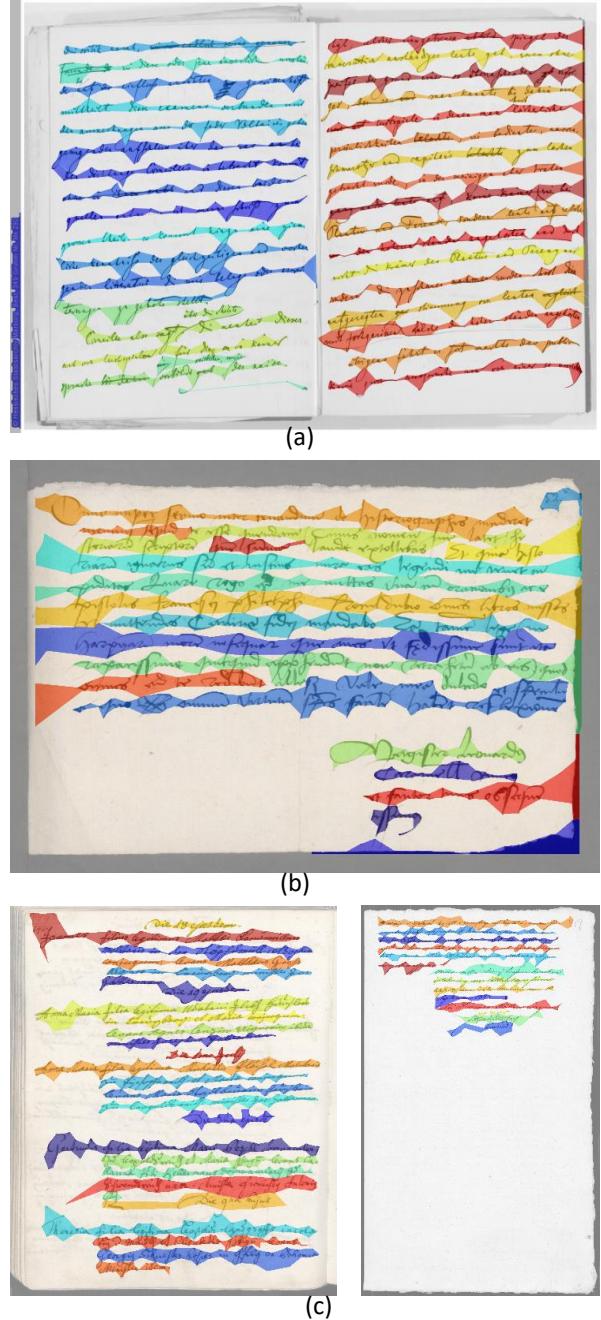


Fig. 12 Example results from cBAD dataset.

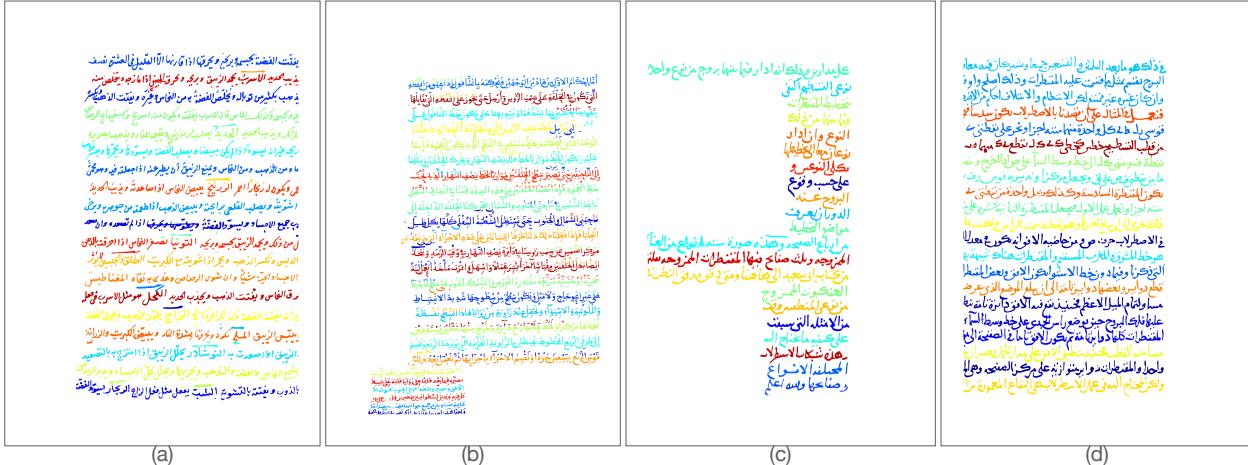
in Arabic. We submitted output text lines and results were evaluated by the competition committee using a success rate defined in [8]. The results and the comparison with other participator methods are presented in Table 7.

8 Conclusion

We presented a learning-free text line detection and extraction method for historical handwritten document

Table 5 Comparison of our method with state of the art results on DIVA-HisDB dataset.

Learning-free	CB55		CSG18		CSG863		Overall	
	Pixel IU	Line IU	Pixel IU	Line IU	Pixel IU	Line IU	Pixel IU	Line IU
[45]	96.67	98.36	96.93	96.91	97.54	99.27	97.05	97.86
Our method	97.04	100	98.14	98.38	97.31	100	97.50	99.46

**Fig. 13** Example results from RASM2018 dataset.**Table 6** Comparison of our results with the cBAD competition results.

	Method	P	R	FM
Learning-based	DMRZ	97.30	97.00	97.10
	UPVLC	93.70	85.50	89.40
	BYU	87.80	90.70	89.20
	LITIS	78.00	83.60	80.70
Learning-free	IRISA	88.30	87.70	88.00
	Our method	82.09	91.13	86.38

Table 7 RASM2018 competition results in terms of success rate defined by the competition.

Tesseract 3	Tesseract 4	FRE11	Our method	RDI
28.80	44.20	43.20	67.70	81.60

images. Historical handwritten documents contain complex layouts with varying text line heights and lengths, touching characters, crowd of ascenders and descenders. Learning-based algorithms are currently an increased trend in text line segmentation of historical handwritten documents; however, they require labeling effort for training. The proposed method can fairly detect the blob lines that strike-through text lines with arbitrary heights and lengths by convolving the input image with second derivative of anisotropic Gaussian using automatic scale selection. On the other hand, EM formulation removes spurious blob lines, assigns the connected components to the closest detected blob line or to the label of the closest component. Ablation study shows

that the method is not sensitive to the parameters except character height estimation. The results on three different datasets are not always the state of the art but achieved using the same experimental setting. In future, we plan to develop an unsupervised machine learning method for text line segmentation, using the blob lines detected by the proposed method as an annotation of the text lines.

Acknowledgements This research was supported by the Frankel Center for Computer Science in Ben-Gurion University of the Negev.

References

- Alaei, A., Pal, U., Nagabhushan, P.: A new scheme for unconstrained handwritten text-line segmentation. *Pattern Recognition* **44**(4), 917–928 (2011)
- Antonacopoulos, A., Karatzas, D.: Document image analysis for world war ii personal records. In: *Document Image Analysis for Libraries*, 2004. Proceedings. First International Workshop on, pp. 336–341. IEEE (2004)
- Baechler, M., Liwicki, M., Ingold, R.: Text line extraction using dmlp classifiers for historical manuscripts. In: *Document Analysis and Recognition (ICDAR)*, 2013 12th International Conference on, pp. 1029–1033. IEEE (2013)
- Bar-Yosef, I., Hagbi, N., Kedem, K., Dinstein, I.: Line segmentation for degraded handwritten historical documents. In: *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pp. 1161–1165. IEEE (2009)
- Biller, O., Rabaei, I., Kedem, K., El-Sana, J.J., et al.: Evolution maps and applications. *PeerJ Computer Science* **2**, e39 (2016)

6. Bukhari, S.S., Shafait, F., Breuel, T.M.: Script-independent handwritten textlines segmentation using active contours. In: 2009 10th International Conference on Document Analysis and Recognition, pp. 446–450. IEEE (2009)
7. Chen, K., Wei, H., Liwicki, M., Hennebert, J., Ingold, R.: Robust text line segmentation for historical manuscript images using color and texture. In: Pattern Recognition (ICPR), 2014 22nd International Conference on, pp. 2978–2983. IEEE (2014)
8. Clausner, C., Antonacopoulos, A., McGregor, N., Wilson-Nunn, D.: Icfhr 2018 competition on recognition of historical arabic scientific manuscripts – rasm2018. In: 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 471–476 (2018)
9. Clausner, C., Antonacopoulos, A., Pletschacher, S.: A robust hybrid approach for text line segmentation in historical documents. In: Pattern Recognition (ICPR), 2012 21st International Conference on, pp. 335–338. IEEE (2012)
10. Cohen, R., Asi, A., Kedem, K., El-Sana, J., Dinstein, I.: Robust text and drawing segmentation algorithm for historical documents. In: Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing, pp. 110–117. ACM (2013)
11. Cohen, R., Dinstein, I., El-Sana, J., Kedem, K.: Using scale-space anisotropic smoothing for text line extraction in historical documents. In: International Conference Image Analysis and Recognition, pp. 349–358. Springer (2014)
12. Delong, A., Osokin, A., Isack, H.N., Boykov, Y.: Fast approximate energy minimization with label costs. International journal of computer vision **96**(1), 1–27 (2012)
13. Diem, M., Kleber, F., Fiel, S., Grüning, T., Gatos, B.: cbad: Icdar2017 competition on baseline detection. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 1355–1360. IEEE (2017)
14. Diem, M., Kleber, F., Sablatnig, R.: Text line detection for heterogeneous documents. In: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on, pp. 743–747. IEEE (2013)
15. Fischer, A., Indermühle, E., Bunke, H., Viehhauser, G., Stolz, M.: Ground truth creation for handwriting recognition in historical documents. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, pp. 3–10. ACM (2010)
16. Garz, A., Fischer, A., Bunke, H., Ingold, R.: A binarization-free clustering approach to segment curved text lines in historical manuscripts. In: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on, pp. 1290–1294. IEEE (2013)
17. Garz, A., Fischer, A., Sablatnig, R., Bunke, H.: Binarization-free text line segmentation for historical documents based on interest point clustering. In: Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on, pp. 95–99. IEEE (2012)
18. Gatos, B., Louloudis, G., Stamatopoulos, N.: Segmentation of historical handwritten documents into text zones and text lines. In: Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on, pp. 464–469. IEEE (2014)
19. Gatos, B., Stamatopoulos, N., Louloudis, G.: Icdar2009 handwriting segmentation contest. International Journal on Document Analysis and Recognition (IJDAR) **14**(1), 25–33 (2011)
20. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. IEEE transactions on pattern analysis and machine intelligence **31**(5), 855–868 (2008)
21. Grueninger, T., Leifert, G., Strauss, T., Labahn, R.: A robust and binarization-free approach for text line detection in historical documents. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 236–241. IEEE (2017)
22. Kesiman, M.W.A., Burie, J.C., Ogier, J.M.: A new scheme for text line and character segmentation from gray scale images of palm leaf manuscript. In: 15th International Conference on Frontiers in Handwriting Recognition 2016, At Shenzhen, China, pp. 325–330 (2016)
23. Kumar, J., Kang, L., Doermann, D., Abd-Almageed, W.: Segmentation of handwritten textlines in presence of touching components. In: Document Analysis and Recognition (ICDAR), 2011 International Conference on, pp. 109–113. IEEE (2011)
24. Kurar Barakat, B., Cohen, R., El-Sana, J., Rabaev, I.: Vml-moc: Segmenting a multiply oriented and curved handwritten text line dataset. In: 3rd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), pp. 151–155. IEEE (2019)
25. Kurar Barakat, B., Drobly, A., Kassis, M., El-Sana, J.: Text line segmentation for challenging handwritten document images using fully convolutional network. In: 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 374–379. IEEE (2018)
26. Lindeberg, T.: Feature detection with automatic scale selection. International journal of computer vision **30**(2), 79–116 (1998)
27. Liwicki, M., Graves, A., Bunke, H.: Neural networks for handwriting recognition. In: Computational intelligence paradigms in advanced pattern classification, pp. 5–24. Springer (2012)
28. Manmatha, R., Srimal, N.: Scale space technique for word segmentation in handwritten documents. In: International conference on scale-space theories in computer vision, pp. 22–33. Springer (1999)
29. Mehri, M., Héroux, P., Gomez-Kramer, P., Boucher, A., Mullot, R.: A pixel labeling approach for historical digitized books. In: 12th International Conference on Document Analysis and Recognition, pp. 817–821. IEEE (2013)
30. Moysset, B., Kermorvant, C., Wolf, C.: Full-page text recognition: Learning where to start and when to stop. In: Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on, vol. 1, pp. 871–876. IEEE (2017)
31. Moysset, B., Kermorvant, C., Wolf, C., Louradour, J.: Paragraph text segmentation into lines with recurrent neural networks. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 456–460. IEEE (2015)
32. Moysset, B., Louradour, J., Kermorvant, C., Wolf, C.: Learning text-line localization with shared and local regression neural networks. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 1–6. IEEE (2016)
33. Naegel, B., Wendling, L.: A document binarization method based on connected operators. Pattern Recognition Letters **31**(11), 1251–1259 (2010)
34. Oliveira, S.A., Seguin, B., Kaplan, F.: dhsegment: A generic deep-learning approach for document segmentation. In: 2018 16th International Conference on Frontiers

- in Handwriting Recognition (ICFHR), pp. 7–12. IEEE (2018)
- 35. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics* **9**(1), 62–66 (1979)
 - 36. Ouwayed, N., Belaïd, A.: A general approach for multi-oriented text line extraction of handwritten documents. *International Journal on Document Analysis and Recognition (IJDAR)* **15**(4), 297–314 (2012)
 - 37. Pastor-Pellicer, J., Afzal, M.Z., Liwicki, M., Castro-Bleda, M.J.: Complete system for text line extraction using convolutional neural networks and watershed transform. In: Document Analysis Systems (DAS), 2016 12th IAPR Workshop on, pp. 30–35. IEEE (2016)
 - 38. Renton, G., Soullard, Y., Chatelain, C., Adam, S., Ker-morvant, C., Paquet, T.: Fully convolutional network with dilated convolutions for handwritten text line segmentation. *International Journal on Document Analysis and Recognition (IJDAR)* **21**(3), 177–186 (2018)
 - 39. Rother, C., Kolmogorov, V., Blake, A.: "grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)* **23**(3), 309–314 (2004)
 - 40. Roy, P.P., Pal, U., Lladós, J.: Text line extraction in graphical documents using background and foreground information. *International Journal on Document Analysis and Recognition (IJDAR)* **15**(3), 227–241 (2012)
 - 41. Saabni, R., Asi, A., El-Sana, J.: Text line extraction for historical document images. *Pattern Recognition Letters* **35**, 23–33 (2014)
 - 42. Saabni, R., El-Sana, J.: Language-independent text lines extraction using seam carving. In: Document Analysis and Recognition (ICDAR), 2011 International Conference on, pp. 563–568. IEEE (2011)
 - 43. Shapiro, V., Gluhchev, G., Sgurev, V.: Handwritten document image segmentation and analysis. *Pattern Recognition Letters* **14**(1), 71–78 (1993)
 - 44. Shi, Z., Govindaraju, V.: Line separation for complex document images using fuzzy runlength. In: null, p. 306. IEEE (2004)
 - 45. Simistira, F., Bouillon, M., Seuret, M., Würsch, M., Alberti, M., Ingold, R., Liwicki, M.: Icdar2017 competition on layout analysis for challenging medieval manuscripts. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 1361–1370. IEEE (2017)
 - 46. Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., Alaei, A.: Icdar 2013 handwriting segmentation contest. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 1402–1406. IEEE (2013)
 - 47. Swaileh, W., Mohand, K.A., Paquet, T.: Multi-script iterative steerable directional filtering for handwritten text line extraction. In: Document Analysis and Recognition (ICDAR), 2015 13th International Conference on, pp. 1241–1245. IEEE (2015)
 - 48. Varga, T., Bunke, H.: Tree structure for word extraction from handwritten text lines. In: Eighth International Conference on Document Analysis and Recognition (ICDAR'05), pp. 352–356. IEEE (2005)
 - 49. Witkin, A.P.: Scale-space filtering. In: Readings in Computer Vision, pp. 329–332. Elsevier (1987)
 - 50. Wong, K.Y., Casey, R.G., Wahl, F.M.: Document analysis system. *IBM journal of research and development* **26**(6), 647–656 (1982)
 - 51. Yin, F., Liu, C.L.: Handwritten chinese text line segmentation by clustering with distance metric learning. *Pattern Recognition* **42**(12), 3146–3157 (2009)