Text Line Detection in Corrupted and Damaged Historical Manuscripts

Irina Rabaev, Ofer Biller, Jihad El-Sana, Klara Kedem
Department of Computer Science
Ben-Gurion University
Beer-Sheva, Israel
rabaev,billero,el-sana,klara@cs.bgu.ac.il

Itshak Dinstein
Department of Electrical and Computer Engineering
Ben-Gurion University
Beer-Sheva, Israel
dinstein@ee.bgu.ac.il

***Abstract*—Most of the algorithms proposed for text line detection are designed to process binary images as input. For severely degraded documents, binarization often introduces significant noise and other artifacts. In this work we present a novel method designed to detect text lines directly in gray scale images. The method consists of two stages. Potential characters are detected in the first stage. This is done by analyzing the evolution maps of connected components obtained by a sliding threshold. The detected potential characters are grouped into text lines in the second stage using sweep-line approach. The suggested method is especially powerful when applied to torn and damaged documents that other algorithms are not able to deal with.**

I. Introduction

Various algorithms for historical document processing, such as indexing, word retrieval and recognition, assume that their input is given in the form of extracted text lines. The vast majority of procedures for text line extraction are designed to process binary images as input. Hence, they are unsuitable for highly damaged documents, as their binarization causes significant noise and may distort the information in the image. Despite considerable progress over the last decade, automatic text line segmentation of severely degraded documents remains an open problem.

In this work we present a novel method designed to detect text lines directly in gray scale images. The method utilizes evolution maps of connected components [1], which provide information about the location of potential characters in the processed documents. Then, a vertical sweep-line is moved across the document image accumulating the potential characters (elements) into lines. When the sweep-line encounters a new element, the algorithm determines whether to assign this element to one of the already discovered text lines, or to initiate a new one.

The suggested method is especially powerful when applied to damaged, torn and stained documents.

II. Related Work

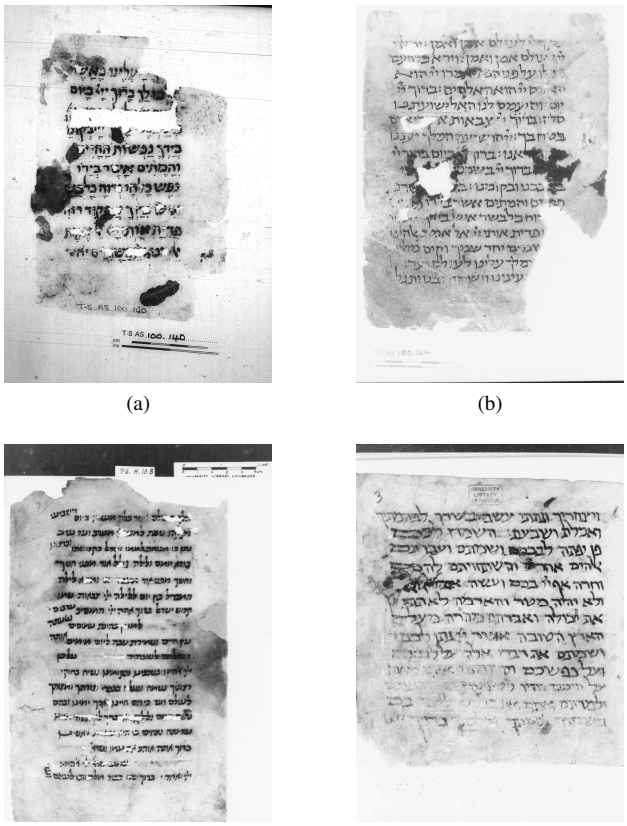
The vast majority of the text line extraction algorithms require the input to be binary (or they perform binarization at various stages of the algorithm) [2]–[12].

Very few methods address text line segmentation of gray scale images. Öztop *et al.* [13] introduced repulsive attractive (RA) network for baseline extraction on document images. Already extracted baselines act as repulsive forces, and pixels of the image act as attractive forces. However, the authors mention that RA method provides poor results when handwritten documents have high skew or/and large portion of overlapping between adjacent lines. Shi *et al.* [14] converted an input gray scale image into so called adaptive local connectivity map (ALCM), where the value of each pixel is defined to be the cumulative intensity of all pixel values inside a window of a predefined size. Then, the ALCM image is binarized, and text line patterns are extracted. The text line patterns allow to perform adaptive binarization of the input image. Nevertheless, for extremely degraded gray scale images even adaptive binarization is difficult. Bar-Yosef *et al.* [15] applied an approach based on oriented local projection profiles (LPP) of gray scale documents. LPP is calculated inside a sliding stripe. The average skew of the current stripe is calculated and the next stripe is projected in this skew direction. The authors reported accurate results on a set of historical documents in different skew angles and with curved text lines. Asi *et al.* [16] used seam-carving approach, where two types of seams, medial and separating, are calculated. A medial seam is a curve that crosses the text line and a separating seam is curve that separates two adjacent lines. Both types of seems propagate according to energy maps, which are defined based on the distance transform of the gray scale image. Although the algorithm achieves a good segmentation accuracy, the seams tend to diverge when big gaps between words or holes in the document are present. The method presented by Garz *et al.* [17] relies on interest points which represent letters. First, interest points are extracted from gray scale images. Next, word clusters are identified in hight-density regions. Finally, text lines are generated by concatenating neighboring word clusters.

Despite considerable progress over the last decade, automatic text line segmentation of severely degraded documents, as those presented in Fig. 1, remains an open problem.

III. Our Approach

In this paper we present an algorithm that gives satisfactory results for text line detection when applied to gray scale images of damaged historical documents which contain stains and holes.



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

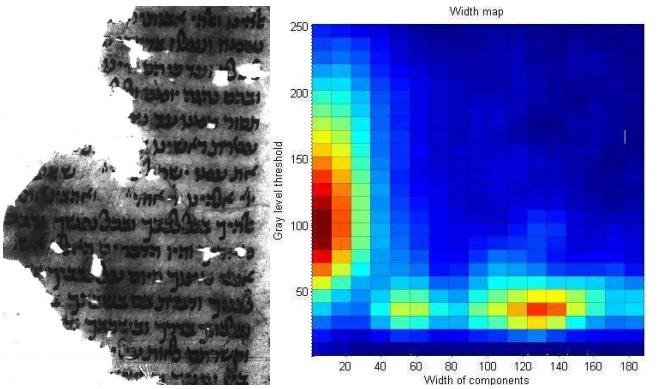
Fig. 1. Samples of the degraded documents on which we perform our tests.

In a preliminary step, the proposed method detects the locations of potential characters in the document, as presented in in Section III-A. Then, a vertical sweep-line moves across the document and the detected elements are accumulated into lines using sweep-line approach. This step is detailed in Section III-B.

A. Evolution maps of connected components

The proposed approach utilizes evolution maps (EMAPs) of connected components, introduced by Biller *et al.* [1]. Consider a gray scale image I with N levels of gray. Denote by B_I the binary image obtained by thresholding I with a threshold $T_g = g$. Let C_I be the set of the connected components in the binary image B_I. Let P_I be a measurable property of the connected components of C_I (e.g., area, width, height, etc.). The EMAP of the image I is defined to be the two dimensional histogram of the values of a property P_I over all the values of g. The histogram depicts the change in the distribution of P_I along the change in the gray scale threshold. For example, in Fig. 2b, we see the distribution of the connected components by width for the document in Fig. 2a. The Y-axis represents the threshold level, the X-axis represents width, and the Z-axis (color) represents the number of components for each width in the given threshold.

Since the EMAP is created for a text document, there is a high density in the range of character widths across the



(a) (b)

Fig. 2. (a) A document image, (b) the distribution of connected components along their width property (x-coordinate), for each possible gray scale threshold (y-coordinate) for document image in (a).

range of gray scale thresholds which would discriminate these characters from their background. In the map in Fig. 2b we notice such a blob centered around width of 130 pixels and ranges over gray scale threshold from approximately 30 to 50. This blob represents the document's characters. The high, dark red region along the Y-axis near width equal to zero represents the noise in the document.

The EMAP supplies details about the average size of the characters in the document, without the need to perform binarization. Knowing the average size of the character, we identify all the components with width within the range $[w-\varepsilon_1, w+\varepsilon_1]$ and height within the range $[h-\varepsilon_2, h+\varepsilon_2]$, where w, h are the average character width and height, respectively. For detailed explanation of this step, we refer the reader to [1]. Fig. 3 illustrates the set of elements identified for the documents in Fig. 1a and Fig. 2a. Each element is represented by its bounding box.

The set of identified elements does not contain all the characters present in the document. Furthermore, it might contain fragments of characters, couples of characters and noise. When the majority of the identified elements represent potential characters, we are able to detect text lines in the document, as shown in Section III-B.

B. Text line detection

The elements (potential characters) obtained in Section III-A are accumulated into lines using the sweep-line approach. A vertical sweep-line moves across the image in the direction of writing (in the case of Hebrew from right to left). When the sweep-line encounters an element, the algorithm determines whether to assign this element to one of the already discovered text lines, or to initiate a new line. The decision is based on the amount of vertical overlap between the boundaries of the already discovered text lines and the processed element, as detailed bellow.

The elements already assigned to lines are used to estimate the top and bottom boundaries of the discovered text lines at the location of the sweep-line. The upper boundary of a text

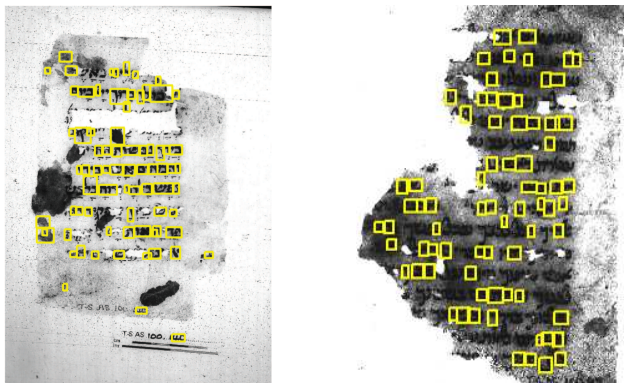


Fig. 3. The extracted set of potential characters for the documents in Fig. 1a and Fig. 2a; each potential character is represented by its bounding box.

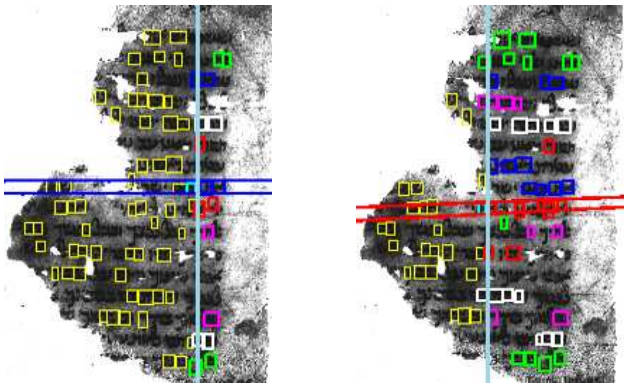
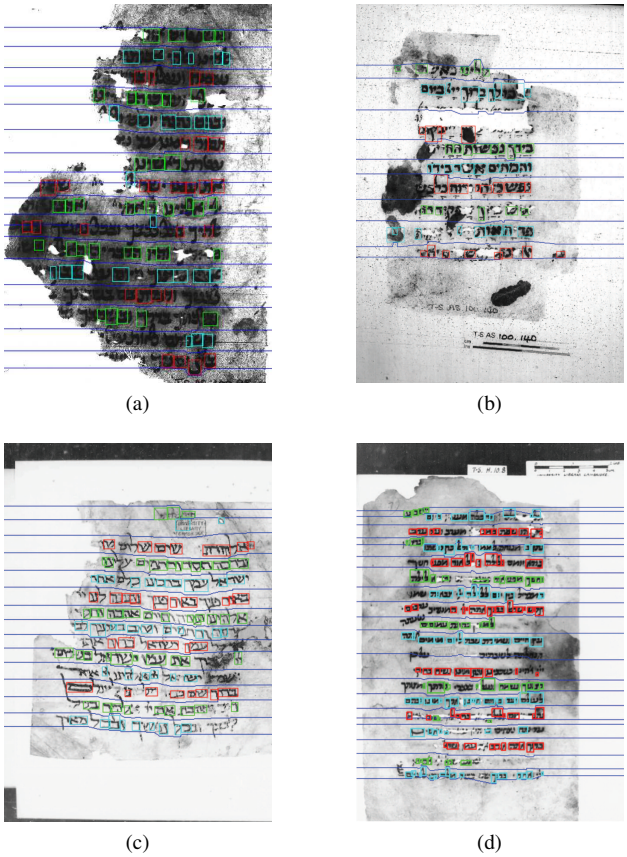


Fig. 4. The progress of line detection. The sweep line moves from right to left (cyan vertical line), the unprocessed elements are shown by yellow rectangles, the already processed elements within the same line have the same color, and the currently processed element is shown as a cyan rectangle on the sweep line. The top and bottom curves of the text line, to which the processed element is to be assigned, are shown in blue on the left and red on the right.

line is a curve that fits the upper vertices of the bounding boxes of the elements. Symmetrically, the lower boundary is a curve that fits the lower vertices of the bounding boxes. Each boundary curve is estimated by fitting first order polynomial with least squares error. Ascenders and descenders might cause boundary curves to deviate from their real direction. Fortunately, ascenders and descenders can be detected with a high probability. A tall component is a candidate to be an ascender or descender. We compare the location of a tall component to the locations of its recent neighbors in the same text line, and according to the result of the comparison the component is treated as an ascender or descender. While fitting the top boundary curve, if the component represents an ascender, we move slightly below the location of its top vertices. Similarly, while fitting the bottom boundary line, we move slightly above the locations of the bottom vertices of the descender.

After estimating the boundaries of so far discovered text lines, we calculate the amount of vertical overlap between these text lines and the currently processed element. If the vertical overlap with one of the lines is above a certain threshold, the processed element is assigned to that line, otherwise it initiates a new line. The text line detection progress is illustrated in Fig. 4. The sweep line moves from right to left (cyan vertical line), the unprocessed elements are shown by yellow rectangles, the already processed elements within the same line have the same color, and the currently processed element is shown as a cyan rectangle on the sweep line. The top and bottom curves of the text line, to which the processed element is to be assigned, are shown in blue on the left and red on the right.

At the end of the process all the elements are grouped into text lines and we calculate the segmentation border between consecutive text lines. First, the top and bottom envelopes of each text line are extrapolated in the regions where there are no elements. This is accomplished by linear extrapolation. The segmentation border of two consecutive text lines is determined to be the curve passing between the bottom envelope of the upper text line and top envelope of lower text line. The final results of the text line detection are illustrated in Fig. 5. The segmentation borders are shown as blue curves.



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

Fig. 5. Line detection results of the algorithm. The elements that belong to the same line have the same color; the segmentation borders are shown as blue curves.

IV. Experimental results and Discussions

To evaluate the performance of the proposed algorithm, we chose 58 severely degraded pages from Cairo Genizah

collection[1]. Fig. 1 shows few examples from our test set. There is no ground truth for the Genizah collection, thus we built it manually for our set using an interactive web-based system [18]. We marked the bounding boxes of all recognizable characters in the document, and indicated groups of characters belonging to the same text line.

We adapted the evaluation strategy as used in ICDAR handwriting segmentation contest [19]. The evaluation strategy presented in [19] is defined for binary images, so we slightly modified it for gray scale images. Two values, detection rate (DR) and recognition accuracy (RA), are calculated. The detection rate and recognition accuracy are based on the number of matches between the line regions detected by the algorithm and the line regions in the ground truth, and are calculated as follows:

$$DR = \frac{o2o}{N}, RA = \frac{o2o}{M},$$

where N and M are the number of text lines in the ground truth and the number of text lines detected by the algorithm respectively, and $o2o$ is the number of one-to-one matches. A text line pair, which consists of a detected line and a ground truth line, is considered as a one-to-one match if and only if both lines contain the same set of characters. Since each character in the ground truth is represented by its bounding box, we consider that the text line region detected by the algorithm contains a character if it contains at least 60% of its bounding box area. A measure that combines detection rate and recognition accuracy is the performance metric FM, which is defined to be their harmonic mean:

$$FM = \frac{2 \times DR \times RA}{DR + RA}$$

Our first experiment was devoted to the estimation of the boundary curves of the detected text lines. As we mention in section III-B, the elements already assigned to a text line are used for boundary curve estimation. Since the handwritten text is characterized by line fluctuation, if too few or too many elements within the line were to be considered, the boundary curves would not be estimated correctly. In order to assess this number accurately, the experiments had to be run on documents with relatively long text lines. We chose a set of 29 documents and ran the experiments with a varying number of elements. We determined that the ideal number of elements for boundary estimation is between seven to ten. So, for evaluating the performance of the proposed algorithm, we used up to ten recently assigned elements in a text line for estimating its boundary curves.

The results averaged over 58 documents from Cairo Genizah are $DR = 0.8823$, $RA = 0.8466$, and $FM = 0.8610$. Taking into consideration that the tested documents are torn, stained and highly damaged, the results are very encouraging. Besides this, the presented method does not require any preprocessing step, e.g. noise reduction or text zone location.

[1]The Cairo Genizah is the large collection of Hebrew medieval manuscripts, written between the 9th and 19th centuries AD. The documents are written in Judeo-Arabic, Hebrew, Aramaic and Yiddish languages using the Hebrew alphabet. Among the documents are daily business papers of merchants, the sacred texts of Judaism, ancient fragments of the Koran, and the Book of Wisdom of Jesus Son of Sirach.

(a) (b)

Fig. 6. Samples from Saint Gall (on the left) and Parzival (on the right) databases with superimposed results of our algorithm.

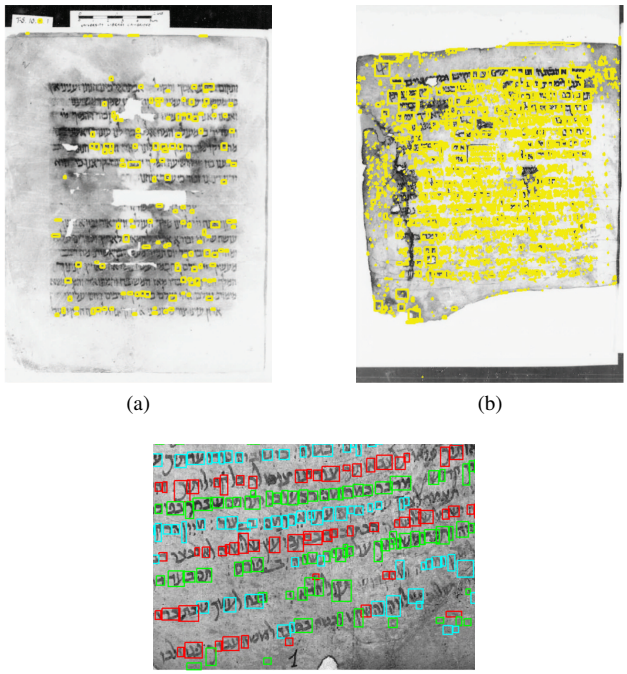
Recently, the image processing team of the Genizah project have announced that they developed an automatic web system for computing a number of features, including text line identification[2]. Unfortunately, they did not provide the details of the method. We experimented with their system. While it successfully detects text lines on the majority of the tested documents, we found a number documents for which the Genizah system failed, two of them appear in Fig. 5c and Fig. 5d. Genizah system announces that these documents are not suitable for computerized analysis. As seen in Fig. 5, our algorithm was able to correctly detect most of the text lines in these documents.

To test the applicability of the proposed approach to non-Hebrew documents, we used Saint Gall and Parzival databases. The Saint Gall database is presented in [20]. It contains 60 pages of a Latin manuscript from the 9th century written by single writer. The Parzival database is described in [21]. It contains 47 pages of a German manuscript from the 13th century written by three writers. Fig. 6 presents the samples of documents from Saint Gall and Parzival datasets with the superimposed results of our algorithm. The results of the proposed algorithms are $DR = 0.9784$, $RA = 0.8633$, and $FM = 0.9142$ on Saint Gall database,[3] and $DR = 0.8106$, $RA = 0.8652$, and $FM = 0.8363$ on Parzival database.

Although the proposed approach achieves encouraging results, it suffers from some limitations. The algorithm relies on the fact that the set of elements obtained from the evolution maps represents potential characters in the document. Once this set is not identified correctly, the algorithm does not provide meaningful results. An example of documents for which the set of potential characters was not identified correctly is shown in Fig. 7. In following research we plan to improve the use of the evolution maps in order to obtain

[2]http://www.genizah.org/

[3]Garz *et al.* [17] used slightly different evaluation criteria. Without getting into details, our results for Saint Gall dataset using their criteria is line accuracy 0.9784, while the results of Garz *et al.* is 0.9797. As can be seen, our results are almost the same.



(a) (b) (c)

Fig. 7. (a), (b) The set of potential characters (shown in yellow) was not identified correctly; (c) the high skew and the components resulted from noise (in the lower right corner) cause to wrong grouping of the elements.

a more reliable set of potential characters. This can be done by validating the set using additional information, such as the percentage of the bounding box occupied by the element, the size of the bounding box's diagonal, etc. Other limitations of the proposed method are when the skew of the text line is high or when the evolution map outputs many elements which are noise. In this case the boundaries of the text line might not be estimated correctly, due to wrong grouping of the elements. For example, in Fig. 7c the high skew and the components resulted from noise (in the lower right corner) cause wrong grouping of the elements. In future research we plan to find a more robust procedure to accurately estimate the boundaries of the text lines with a relatively high skew.

V. Conclusions and Future Directions

We proposed a novel algorithm for text line detection in damaged, torn and stained gray scale document images. An advantage of the method is that it does not require binarization or any other preprocessing, e.g., text area detection or noise reduction. In future research we plan to upgrade the proposed method in two directions: (1) refine the use of the evolution maps to obtain a more reliable set of potential characters in the document, and (2) find a more robust procedure for accurate estimation of the boundaries of the text lines with a relatively high skew.

References

[1] O. Biller, K. Kedem, I. Dinstein, and J. El-Sana, "Evolution maps for connected components in text documents," in *International Conference on Frontiers in Handwriting Recognition (ICFHR'12)*, 2012, pp. 403–408.

[2] L. Likforman-Sulem, A. Zahour, and B. Taconet, "Text line segmentation of historical documents: a survey," *International Journal on Document Analysis and Recognition*, vol. 9, no. 2, pp. 123–138, 2007.

[3] M. Arivazhagan, H. Srinivasan, and S. N. Srihari, "A Statistical Approach to Handwritten Line Segmentation," in *Document Recognition and Retrieval XIV, Proceedings of SPIE*, 2007, pp. 1–11.

[4] D. Kennard and W. Barrett, "Separating lines of text in free-form handwritten historical documents," in *Second International Conference on Document Image Analysis for Libraries (DIAL'06)*, 2006, pp. 12–23.

[5] Y. Li, Y. Zheng, and D. Doermann, "Detecting Text Lines in Handwritten Documents," in *The 18th International Conference on Pattern Recognition (ICPR'06)*, vol. 2, 2006, pp. 1030–1033.

[6] Y. Li, Y. Zheng, D. Doermann, and S. Jaeger, "Script-Independent Text Line Segmentation in Freestyle Handwritten Documents," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 8, pp. 1313–1329, 2008.

[7] A. Alaei, U. Pal, and P.Nagabhushan, "A new scheme for unconstrained handwritten text-line segmentation," *Pattern Recognition*, vol. 44, pp. 917–928, 2011.

[8] D. Fernández, J. Lladós, A. Fornés, and R. Manmatha, "On Influence of Line Segmentation in Efficient Word Segmentation in Old Manuscripts," in *International Conference on Frontiers of Handwriting Recognition (ICFHR'12)*, 2012, pp. 759–764.

[9] K. Zagoris, I. Pratikakis, A. Antonacopoulos, B. Gatos, and N. Papamarkos, "Handwritten and Machine Printed Text Separation in Document Images using the Bag of Visual Words Paradigm," in *International Conference on Frontiers of Handwriting Recognition (ICFHR'12)*, 2012, pp. 103–108.

[10] I. Ben-Messaoud, H. El-Abed, H. Amiri, and V. Märgner, "A multilevel text-line segmentation framework for handwritten historical documents," in *International Conference on Frontiers of Handwriting Recognition (ICFHC'12)*, 2012, pp. 513–518.

[11] L. Kang, J. Kumar, P. Ye, and D. Doermann, "Learning Text-Line Segmentation using Codebooks and Graph Partitioning," in *International Conference on Frontiers of Handwriting Recognition*, 2012, pp. 63–68.

[12] M. Liwicki, E.Indermuhle, and H.Bunke, "On-Line Handwritten Text Line Detection Using Dynamic Programming," in *The 9th International Conference on Document Analysis and Recognition (ICDAR'07)*, vol. 1, 2007, pp. 447– 451.

[13] E. Öztop, A. Mülayim, V. Atalay, and F. Yarman-Vural, "Repulsive attractive network for baseline extraction on document images," *Signal Processing*, vol. 75, pp. 1–10, 1999.

[14] Z. Shi, S. Setlur, and V. Govindaraju, "Text Extraction from Gray Scale Historical Document Images Using Adaptive Local Connectivity Map," in *The 8th International Conference on Document Analysis and Recognition (ICDAR'05)*, 2005, pp. 794–798.

[15] I. Bar-Yosef, N. Hagbi, K. Kedem, and I. Dinstein, "Text Line Segmentation for Degraded Handwritten Historical Documents," in *The 10th International Conference on Document Analysis and Recognition (ICDAR'09)*, 2009, pp. 1161–1165.

[16] A. Asi, R. Saabni, and J. El-Sana, "Text Line Segmentation for Gray Scale Historical Document Images," in *International Workshop on Historical Document Imaging and Processing (HIP'11)*, 2011, pp. 120–126.

[17] A. Garz, A. Fischer, R. Sablatnig, and H. Bunke, "Binarization-free text line segmentation for historical documents based on interest point clustering," in *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*, 2012, pp. 95–99.

[18] O. Biller, A. Asi, K. Kedem, and J. El-Sana, "WebGT: An Interactive Web-based System for Historical Document Ground Truth Generation," Computer Science Department, Ben-Gurion University of the Negev, Israel, Tech. Rep. 13–03, 2013.

[19] B. Gatos, N. Stamatopoulos, and G. Louloudis, "ICDAR2009 handwriting segmentation contest," *International journal on document analysis and recognition*, vol. 14, no. 1, pp. 25–33, 2011.

[20] A. Fischer, V. Frinken, A. Fornés, , and H. Bunke, "Transcription Alignment of Latin Manuscripts using Hidden Markov Models," in *1st International Workshop on Historical Document Imaging and Processing (HIP)*, 2011, pp. 29–36,.

[21] A. Fischer, A. Keller, V. Frinken, , and H. Bunke, "Lexicon-Free Handwritten Word Spotting Using Character HMMs," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 934–942, 2012.