Evolution maps and applications

Ofer Biller[1], Irina Rabaev[1], Klara Kedem[1], Its'hak Dinstein[2] and
Jihad J. El-Sana[1]

[1] Department of Computer Science, Ben-Guion University of the Negev, Beer-Sheva, Israel
[2] Electrical and Computer Engineering Department, Ben-Guion University of the Negev,
Beer-Sheva, Israel

Submitted 15 April 2015
Accepted 19 November 2015
Published 6 January 2016

Corresponding author
Ofer Biller, billero@cs.bgu.ac.il

Academic editor
Pinar Duygulu

Additional Information and Declarations can be found on page 18

DOI 10.7717/peerj-cs.39

ABSTRACT

Common tasks in document analysis, such as binarization, line extraction etc., are still considered difficult for highly degraded text documents. Having reliable fundamental information regarding the characters of the document, such as the distribution of character dimensions and stroke width, can significantly improve the performance of these tasks. We introduce a novel perspective of the image data which maps the evolution of connected components along the change in gray scale threshold. The maps reveal significant information about the sets of elements in the document, such as characters, noise, stains, and words. The information is further employed to improve state of the art binarization algorithm, and achieve automatically character size estimation, line extraction, stroke width estimation, and feature distribution analysis, all of which are hard tasks for highly degraded documents.

Subjects Computer Vision, Digital Libraries
Keywords Text document analysis, Historical documents, Degraded documents, Connected component analysis

INTRODUCTION

In recent years there has been a growing effort to digitize historical documents. Due to advanced technology, a high quality acquisition of large collections of documents became practical, saving them from physical decay. Many algorithms are being developed in order to extract information out of these documents, which are too numerous to analyze manually. However, some of these documents are severely degraded, and therefore do not provide acceptable performance when processed using common methods.

In our work, we provide a novel perspective on text documents which reveals key information about the various elements of the document such as letters, connected letters, noise, and stains. The extracted information can be used by higher level algorithms to significantly improve their performance. Various algorithms such as binarization, segmentation, word spotting, and recognition often require preliminary information such as character dimensions, stroke width, and noise characteristics of the input document. When dealing with severely degraded documents, this information is essential but is hard to obtain automatically. Less degraded documents parameters, such as stroke width and character size, are often obtained with relative accuracy by using a binarized version of the document (*Roy et al., 2009*; *Wen & Ding, 2004*; *Raju, Pati & Ramakrishnan, 2004a*). The binarization of degraded documents, however, is not reliable as it may introduce noise, falsely merged components, and other artifacts. For many algorithms, information regarding character dimensions is often expected to be provided by the user or determined in an ad-hoc manner (*New et al., 2006*; *Bar-Yosef et al., 2007*). Our method copes with severe degradations and non-uniform backgrounds.

In this paper, we introduce the component evolution maps, and demonstrate some of their applications. We use these maps to estimate letter dimensions and stroke width in degraded documents. We then utilize this information to improve state-of-the-art binarization, line segmentation, and characterizing feature behavior in a document collection.

For a given property of the connected components in a document, the evolution maps demonstrate the evolution of the histogram of the property, along with the change in the intensity threshold. To simplify the discussion, we start explaining the evolution map of the component's width property. For example, Fig. 1 shows the distribution of the width of the connected components for each possible intensity threshold. The y-axis represents the intensity level, the x-axis represents width, and the z-axis (color) represents the density of the components for each width in the given threshold: warm (red, orange, yellow, etc.)—high density, cold (blue, cyan, etc.)—low (this is the color coding of all the maps depicted in the figures in the paper). Since the map represents a text document, we expect to see high density in the range of character width and within the range of intensity thresholds that separates the characters from the background. The histogram in Fig. 1 shows a blob centered around a width of 28 pixels, ranging over gray levels approximately from 40 to 100. This blob represents the characters in the document. The noise, on the other hand, concentrates along the y-axis, and is depicted as a large number of low width connected components in a wide range of gray scale thresholds.

In our research we deal with severely degraded documents, where state-of-the-art document analysis algorithms, such as binarization and line extraction, provide poor results. A good example of such text documents are those from the Cairo Genizah. The Cairo Genizah is a collection of documents, most of them in handwritten Hebrew square letters, which where hidden in an attic of a synagogue in Cairo, Egypt for several hundreds of years. The Genizah collection contains hundreds of thousands of documents dated starting from the ninth century. The documents are characterized by different handwriting styles, document and letter sizes, and materials. Most documents are only textual, each document contains a single writing style with no titles, and many of them are severely degraded. The main motivation in this research is to provide a solid starting point for processing severely degraded historical documents.

In the rest of this paper we briefly overview related work, then describe in detail the component evolution maps, their construction and initial analysis ('Evolution maps'). In 'Applications' we demonstrate several applications for component evolution maps. Finally, in 'Summary,' we conclude our work and draw directions for future research.

RELATED WORK

Connected component analysis has attracted the interest of researchers, and intensive research was performed on document layout segmentation and text separation for

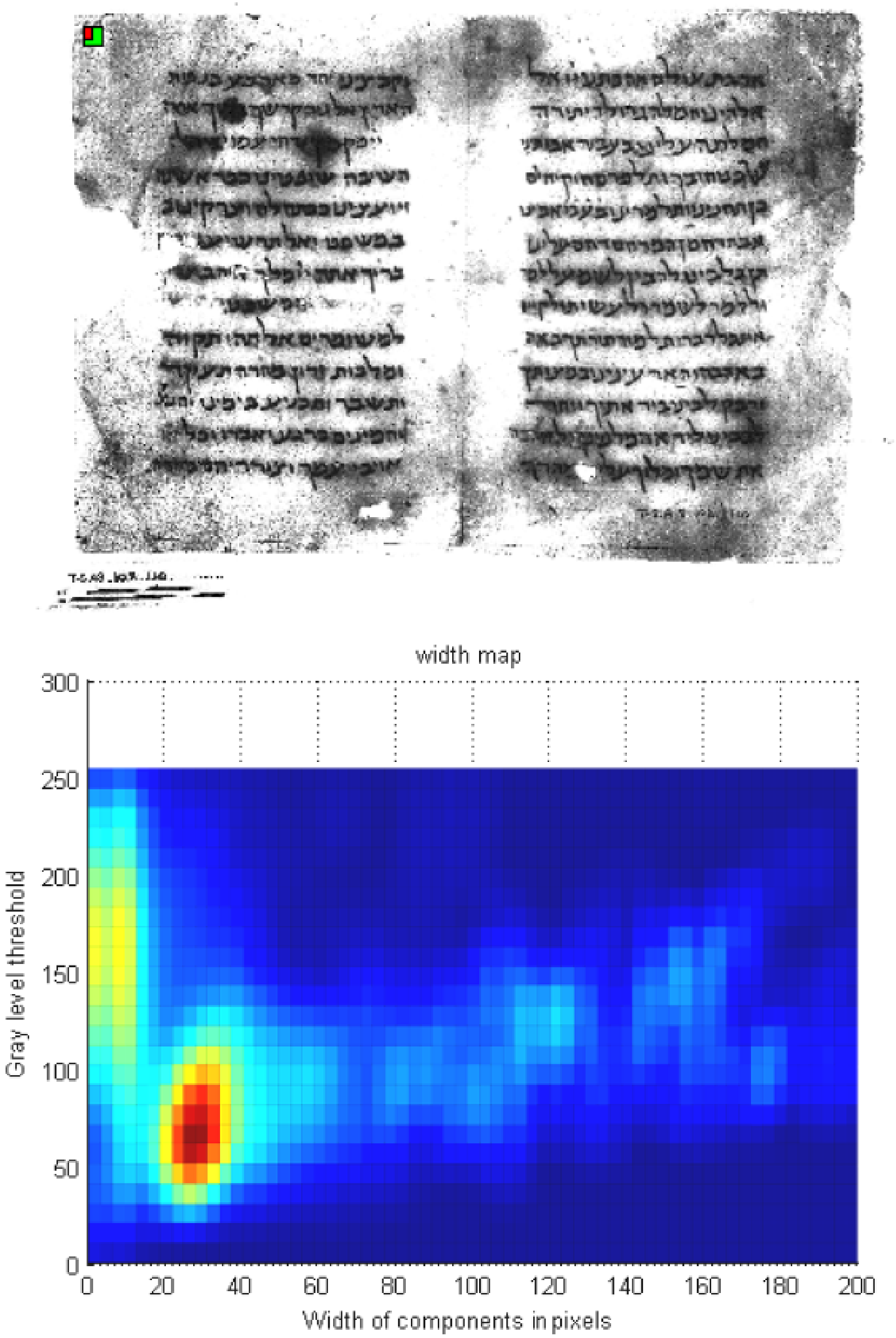


Figure 1 A document image, and the distribution of connected components along their width property (*x*-axis), for each possible gray scale threshold (*y*-axis). The *z*-axis is expressed by color where warm (cold) colors represent high (low) density of components.

binarized document images (*Jain & Zhong, 1996*; *Raju, Pati & Ramakrishnan, 2004b*; *Zagoris & Papamarko, 2010*; *Bukhari et al., 2010*).

Pikaz & Averbuch (1996) selected a threshold for text document by scanning the entire gray scale range, thresholding the image with each value, looking for the widest sub range of gray scale for which the number of the connected components remains stable. This method may be viewed as a special case of our approach.

The component tree is a graph representation computed from the cross-section decomposition of the gray levels of the image (*Mosorov & Kowalski, 2002*), and uses connected components of cross-sections at gray level of the image to assemble different perspectives of the image data. The component tree applies operators on a single component level for image segmentation (*De Carvalho et al., 2010*) and for document binarization (*Naegel & Wendling, 2010*). Similarly, the Maximally Stable Extremal Regions (MSER) descriptor (*Pajdla et al., 2002*) finds extremal regions which maximize stability over threshold change, and is used mostly for object recognition.

Most of these works analyze the change of thresholds in order to achieve local information such as local adaptive binarization, segmenting or locating objects in an image. In our work, we map connected component information for a range of different thresholds over the whole document in order to extract global information about the document and its elements.

EVOLUTION MAPS

We offer a novel perspective of the image data which reflects the evolution of connected components along the change in gray scale threshold. We term it as *Component Evolution Maps* (CEM). CEM brings to the surface underlying information about the sets of elements (e.g., letters, noise, connected letters) in the document image. Below we discuss in detail the construction of the evolution maps followed by their analysis.

Definition of the evolution map

The *evolution map* is a function of the intensity (I) and a property (P) of the connected components of the image into the occurrence level of this property in the image, i.e., $map : I \times P \longmapsto R$, where I is the intensity, P is an image property, and R is the occurrence (detailed below). For example, the width CEM, $map(g,w)$, is the number of connected components of width w in pixels, when thresholding the image at intensity g. To simplify the discussion, we refer in this section only to the width CEM. The CEM provides an intuitive visualization tool to analyze the distribution of an image property. Figure 2 represents the width CEM for a text document image, where 'hot' colors represent high density of components.

The horizontal cross-section at gray scale value g (y-axis) represents the histogram of the components' widths in the image binarized using g as a threshold. Figure 2 shows three cross-sections at different intensity thresholds, and the corresponding binary images. In Fig. 3 we illustrate a cross-section as histogram of component width for a specific threshold, and the corresponding components in the document for two ranges of width (see the squares on the graph).

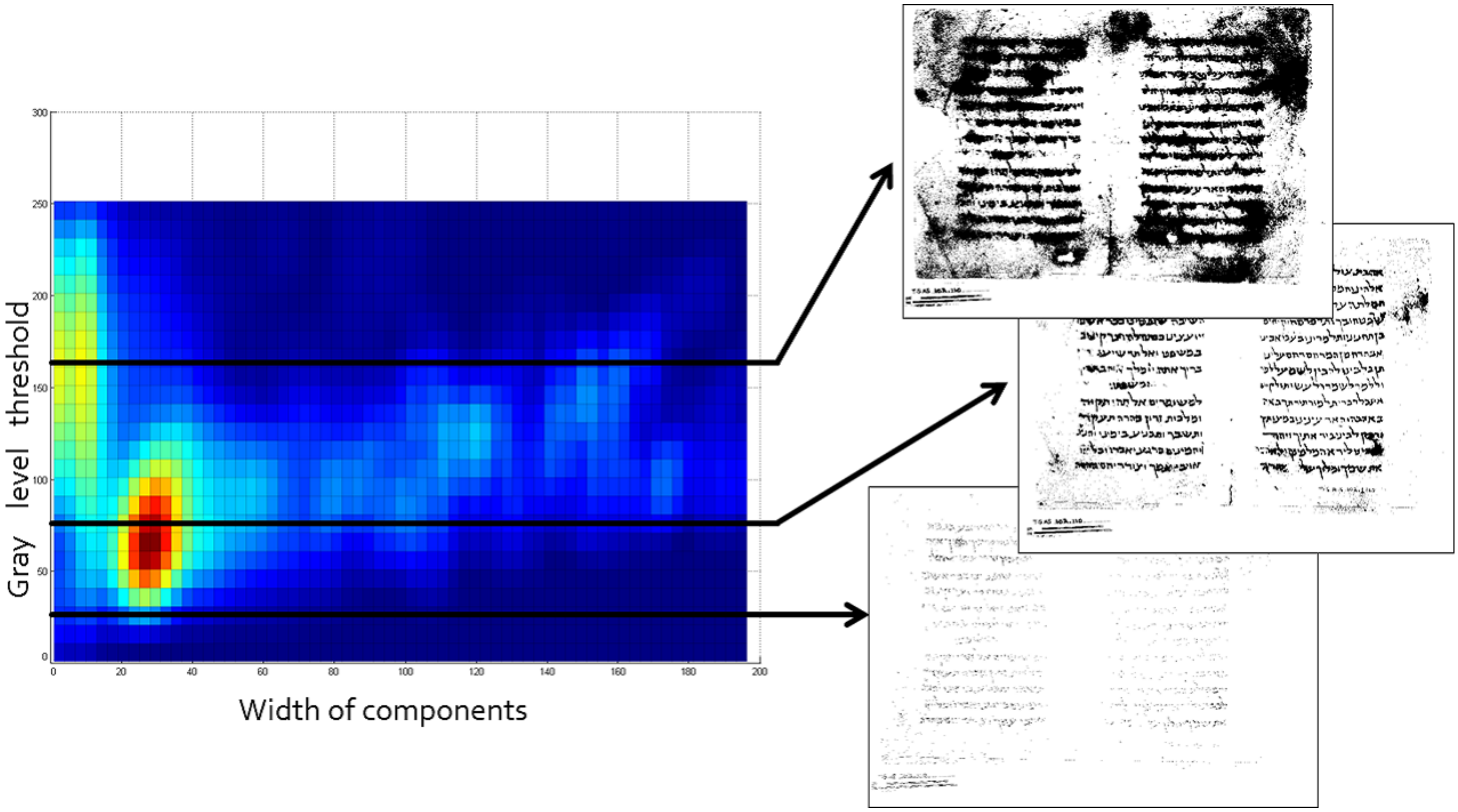


Figure 2 **Horizontal cross-section of the width CEM and the corresponding binary images.**

Construction of the evolution map

We construct the evolution map in a straightforward manner by thresholding the image over gray levels, and counting the number of resulting connected components for each width in the binary image. Since noise usually produces a vast number of small components and skews the histogram, we accumulate the *relative total area* of the components, which is defined as the area of the components normalized by the size of the whole image. This value is less sensitive to noise than component count, as shown in Fig. 4. As seen, the red blob on the top illustration emphasizes the noise, and the bottom illustration highlights the width of the sought elements. We store both the values count and the relative total area, and use each of them in different applications as detailed below. We smooth the CEM using a 2d Gaussian kernel to reduce the influence of local irregularities. Constructing the CEM, for one or few properties, is relatively straight forward and fast. For example, computing the CEMs of 5 properties for a document of resolution 1,600 × 2,300 took about 4.5 s on a standard desktop.

Analysis of the CEM

Some sets of elements in the gray scale documents (noise, characters, connected characters, etc.) have values of the mapped property which are distributed within a given range. Sets which are dominant in the document will form a blob of local maximum on the map. The

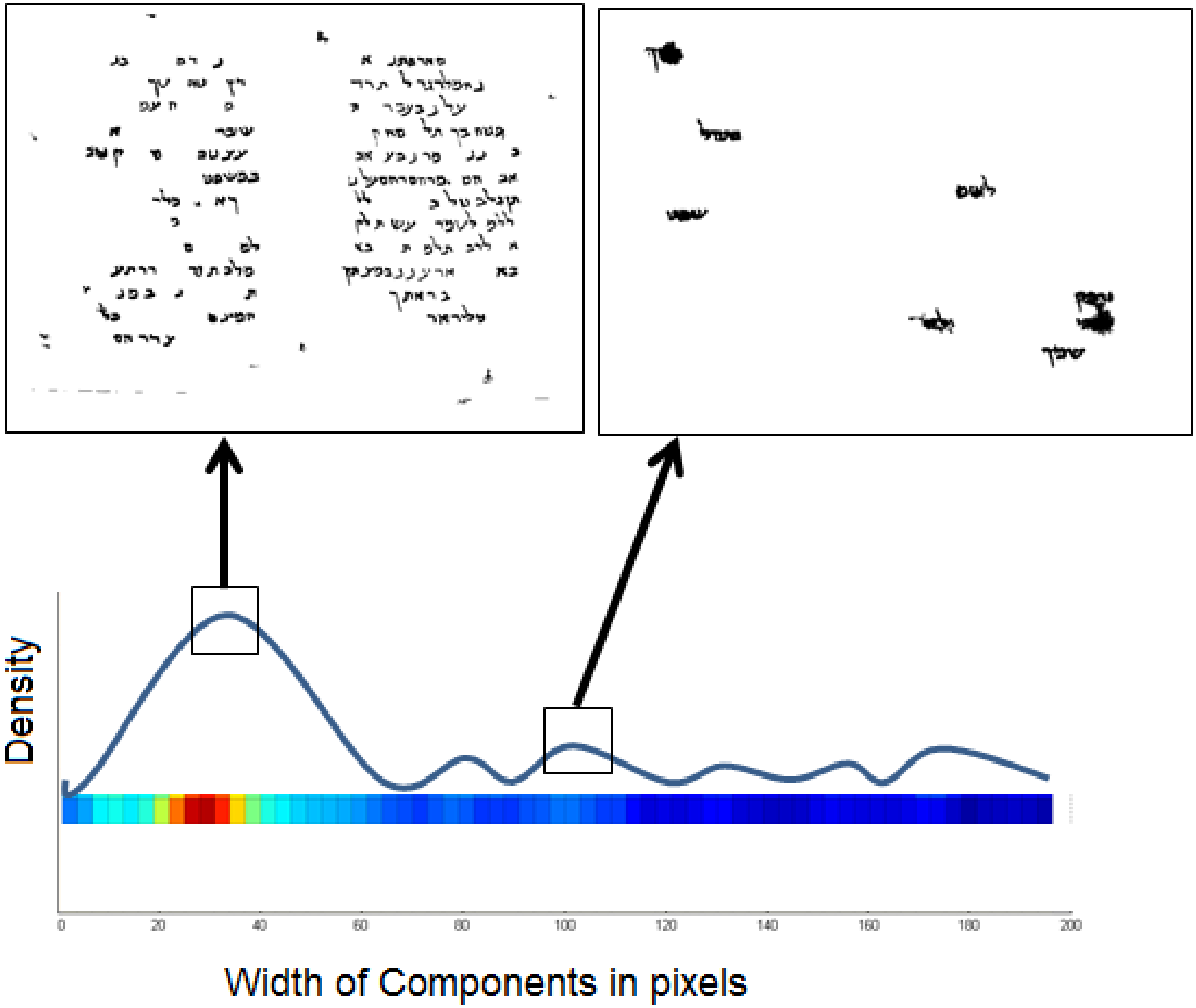


Figure 3 A horizontal cross-section of the densities in the width CEM at one gray level value. The squares on the graph depict the most popular widths for this intensity value. The images of components corresponding to the squares are pointed to by the arrows.

characteristics of the local maximum and its surroundings describe attributes of the set of objects. Due to the statistical nature of the map, it tends to be robust against various defects of the image such as blurriness, local degradations and deformations.

Figure 5 shows a document with its width CEM. Images a–e show the set of elements in the original image corresponding to the blobs a–e in the CEM. Blob a represents a set of noise stains in the document, which appear on relatively high range of threshold values and low width values (indicating small components). The dominance of blob b is supported by the existence of single character elements that occupy a substantial area of the image.

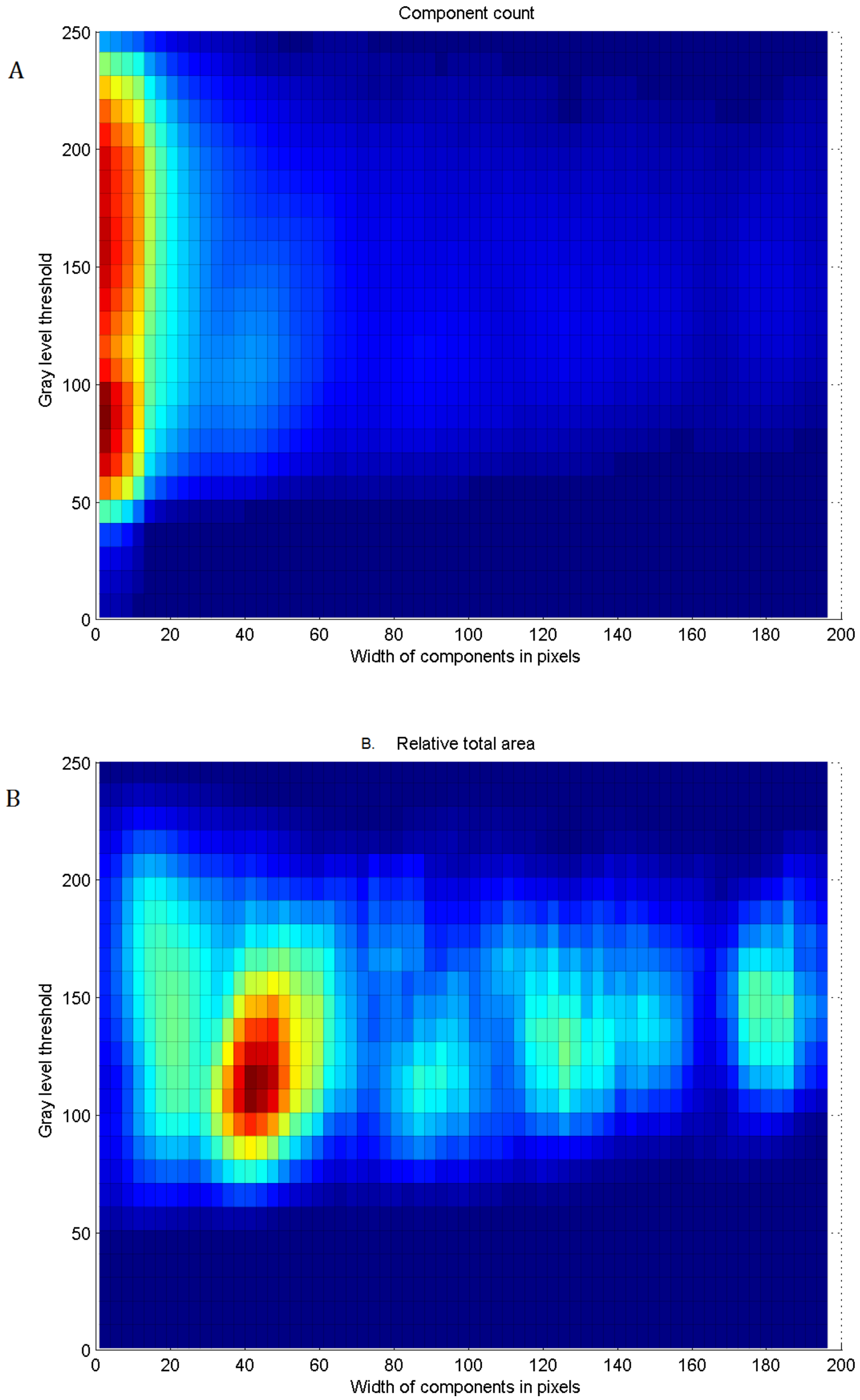


Figure 4 (A) displays the count of components per threshold and width, and (B) shows the relative total area of components per threshold and width.

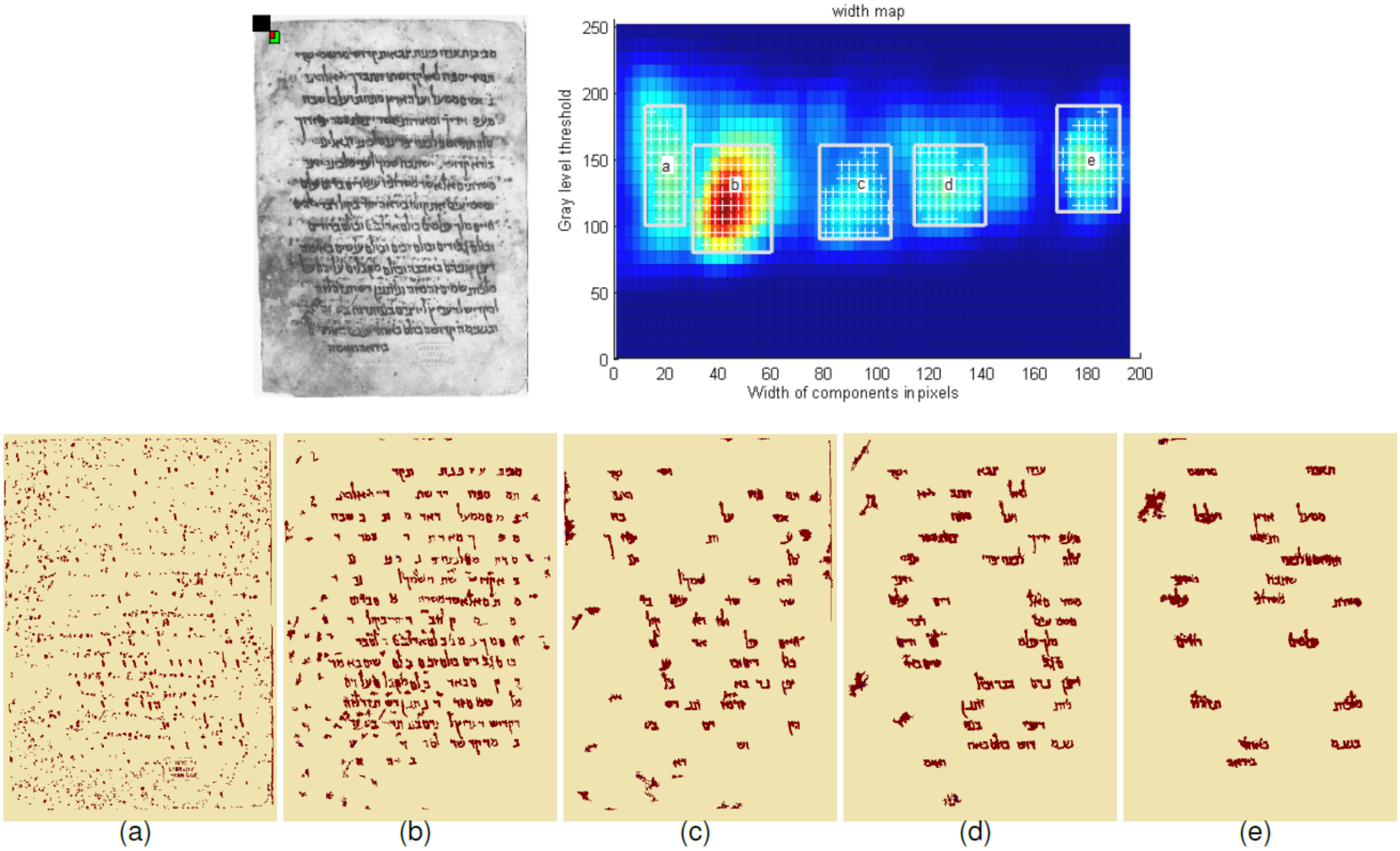


Figure 5 A document image with its components according to the width evolution map. The images (A, B, C, D, E), display the components represented by each of the blobs, marked on the map by (A, B, C, D, E).

The blobs c, d, and e, are at high width values and represent objects of two, three, and four connected letters, respectively.

To extract the information found in the CEM, we look for the main blobs in the map, and model each of them by an anisotropic Gaussian. First, to determine the top blob centroids, we use a sweeping plane that moves downward along the *z*-axis. As the plane descends, it encounters the peaks of the blobs, one by one. Neighboring blobs expand until they touch each other, or reach a low value below a predefined threshold. We use the data of the peaks and their immediate neighborhood to create an anisotropic Gaussian model for each blob. We fit a quadratic surface to the log of the data from the map using the least squares method. We obtain the Gaussian characteristics using the coefficients of the fitted quadratic surface. Given a blob on the map and it's Gaussian modeling, we can obtain an estimation for the distribution of the property values for the elements represented by the blob. In the width example, we receive distribution of the width of elements represented by each blob. In order to receive some hard limits on the values of the property values for the elements in the group, we should apply some threshold on the given distribution (e.g., 3 times the standard deviation).

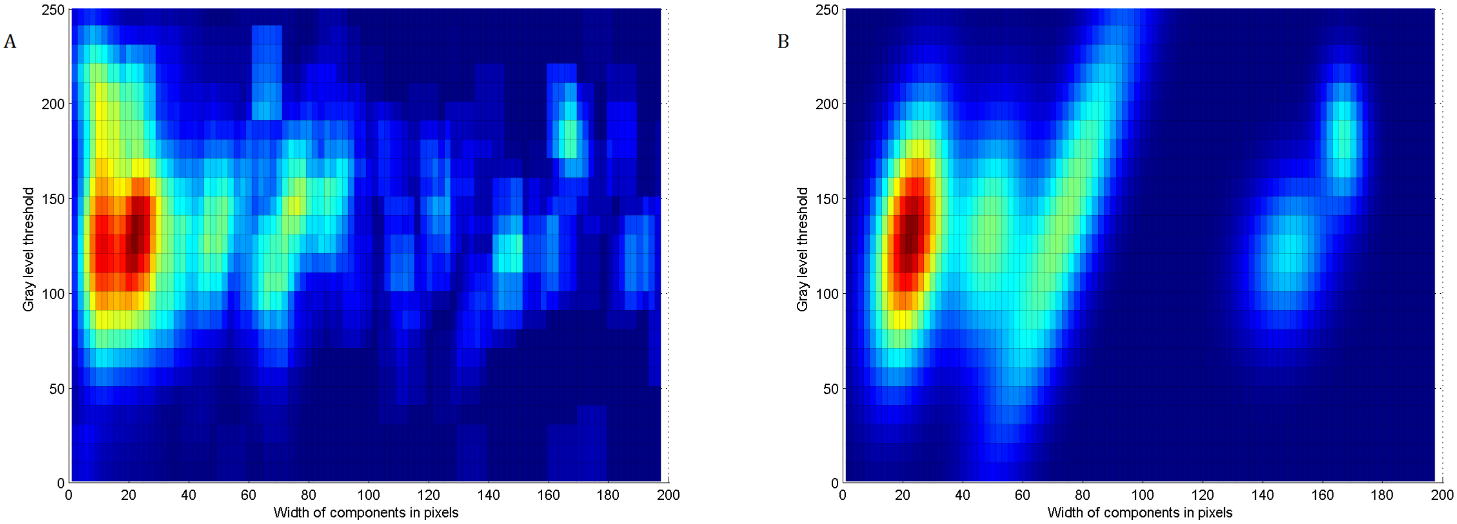


Figure 6 The original width CEM (A), and its modeling by anisotropic Gaussians (B).

Gaussian modeling was validated as suitable for the CEM blobs by examining the data and verifying that the blob distribution is very close to normal distribution. Using Gaussians to model the blobs provides a concise and simplified representation, and handles blob intersection appropriately. It also provides the probability that an element belongs to the set represented by a certain blob. Using Gaussian mixture model was examined as an alternative option, but was found this to be less effective since the mixture model, using expectation maximization, tends to capture small components, while we prefer to ignore small components and model the leading salient blobs more accurately. Figure 6 shows a width CEM and its simplified modeling using the topmost Gaussians.

APPLICATIONS

In this section we demonstrate some applications of CEMs for several document analysis tasks. We robustly estimate the character dimensions and stroke width on degraded documents. This information is then used by many document analysis algorithms to improve their performance and reduce the need for manual adjustments of predefined parameters. We demonstrate utilization of this information to improve state of the art binarization method and simplify line segmentation of degraded documents. Finally, we show how the CEMs is used for exploring feature behavior in text documents. In order to evaluate the performance of these tasks we had to manually generate ground truth for each task (character dimensions, line segmentation, and stroke width). For each task, we randomly selected a subset of the documents from the collection and generated the ground truth relevant for the task.

Estimation of character dimensions

One of the salient elements in a CEM of a text document is the blob which represents the documents' characters. To determine the blob that represents the document's characters, we consider the top blobs and grade them according to the following features. We define the *components of a blob B* as as the set of connected components corresponding to the points (g,w) of B, and the *cardinality of the blob B* as the *number* of connected components represented by B.

Let p be the percentage of the image area occupied by all the components in the blob B, and let n denote the total number of components in B. Equation (1) formulates the score of of the blob B.

$$\text{score} = (a \cdot p) \times \frac{1}{1 + \exp(-c1 \cdot (n - c2))}. \tag{1}$$

The exponential term in the denominator, parametrized by the given constants $c1$ and $c2$, suppresses the score of blobs with small number of components. The blobs with the highest score from both width and height CEMs represent the width and height of the letters in the input image. We also take into account the agreement level between the selected blobs from the width and the height CEMs– the two blobs should have a similar mean and standard deviation on the y-axis (intensity) projection.

The same experiment was conducted also for other properties, such as the component's diagonal and the projection of the component on its main principal axis. These maps provide correct estimates of the character dimensions in most cases, but are less accurate than the estimate obtained by width and height evolution maps.

We applied our method on a collection of digitized historical documents from the Cairo Genizah. We ran our algorithm on a set of hundred documents (selected randomly) with various degradation levels, resolutions and character dimensions. We created width and height CEMs for each document, and estimated the width and height ranges of the characters in each document. For these documents, we have created ground truth (using *Biller et al., 2013*) which includes bounding boxes of the letters. To measure the accuracy of our results, we calculated the precision, recall, and f-measure of our estimate with respect to the ground truth. Figure 8 shows the width CEM for a specific document. Displayed in black over it is a histogram of the ground truth widths. Our estimate of character width is marked by a white rectangle. We calculated the recall as the percentage of letters from the ground truth which fit in our width estimate. The precision is calculated as the ratio between the overlap of our width estimate with the ground truth and the range of our width estimate. Our method achieves accuracy of 87.8% (f-measure) with respect to the provided ground truth.

Improving binarization methods

In this section, we demonstrate improving the results of a binarization algorithm by using CEM to estimate character dimension. We use the winning algorithm in the H-DIBCO-2010 binarization contest (*Pratikakis, Gatos & Ntirogiannis, 2010*) proposed by *Bar-Yosef et al. (2007)*. This binarization algorithm is based on the sliding window approach. The window size is configurable, and the authors recommend it to be around one and a half times the letter size. First we ran the binarization algorithm with a default window size (Fig. 7A). Then, we ran the same binarization algorithm using a window size of one and a half times the maximum of the width and height of the characters in the

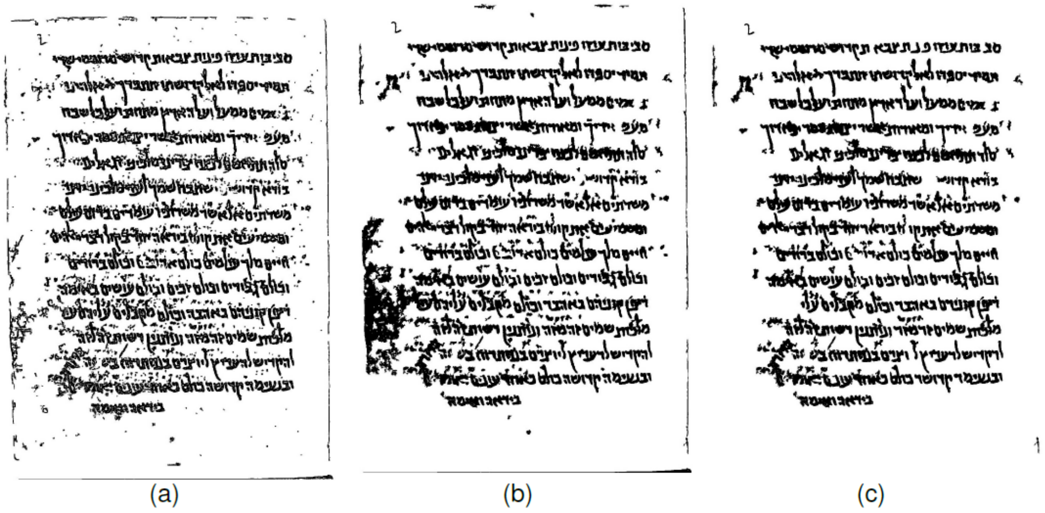


Figure 7 Utilizing our estimation to improve binarization algorithm. (A) Binarization using default window size. (B) Binarization with optimized window size using the output of the CEM. (C) Result after filtering out-of-range components, using the estimate of character dimension, given by the CEM.

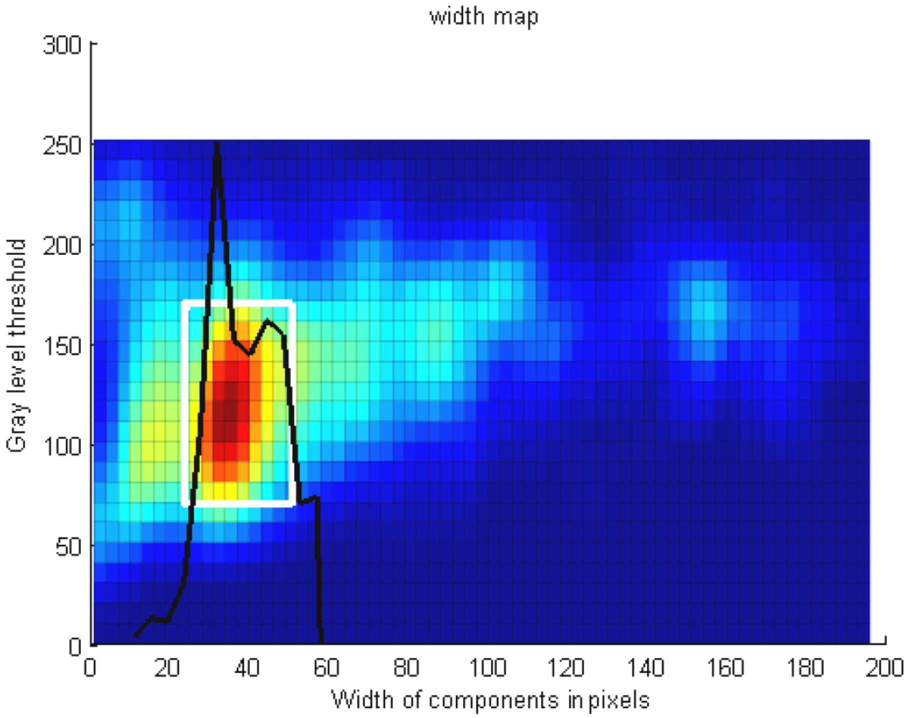


Figure 8 The width CEM of a document's image. Displayed in black on the CEM is the histogram of the ground truth of widths for the letters (the height of the black line represents the count). The white rectangle shows the estimate of the width range of the characters, made by our algorithm.

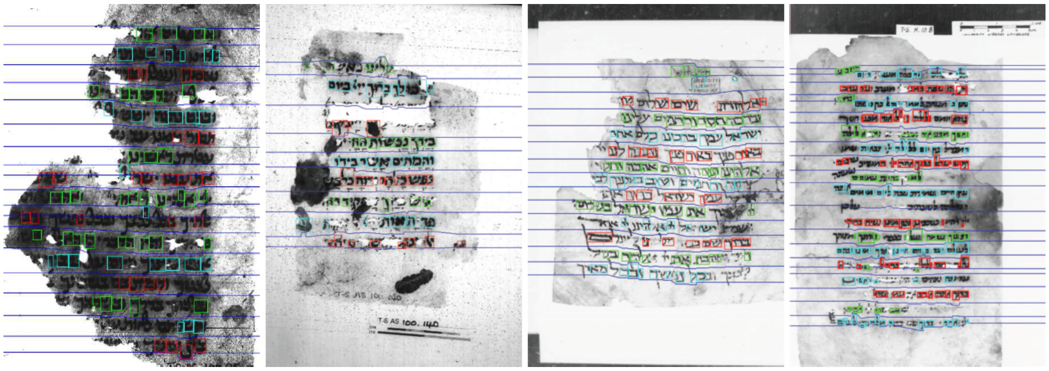


Figure 9 Line detection results of the algorithm. The elements that belong to the same line have the same color. The segmentation borders are shown as blue curves.

document obtained by CEM (Fig. 7B). As an additional improvement of the binarization, we filter out all components of the binarized image which substantially exceed the range of character dimensions (Fig. 7C). We applied our method on seventy degraded documents from the Genizah collection, and observed a substantial improvement in the quality of the results compared with the Bar-Yosef algorithm with default parameters. To quantify the improvement we randomly selected five documents, manually generated a ground truth of binarization for each, and measured the performance of the two binarization methods against the ground truth. The proposed method achieved an average F-Measure of 79.9% (with average percision of 73.1% and average recall of 88.6%) while the original algorithm achived 70.3% (with average percision of 60.1% and average recall of 85.2%).

Line segmentation of degraded documents

In this section, we demonstrate utilizing CEM for text line detection in gray scale images of highly degraded historical documents. Using the information obtained from the CEM, we identify components which are likely to be characters of the documents. We take into account their width, height, and range of thresholds in which they are received. The set of identified elements does not have to include all the characters present in the document. Nevertheless, the identified elements represent most of the characters in the document, which allows detecting text lines in the degraded document, as described below.

The detected potential characters are accumulated into lines using a sweeping line approach. A vertical sweeping line moves across the image in the direction of writing. When the sweeping 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. Full details of the algorithm can be found in *Rabaev et al. (2013)*.

The final result of the text line detection is illustrated in Fig. 9. Elements that belong to the same line have the same color. The segmentation borders are shown in blue.

To evaluate the performance of the algorithm, we employed the evaluation strategy used in the ICDAR handwriting segmentation contest (*Gatos, Stamatopoulos & Louloudis, 2011*). 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:

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

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 measure that combines detection rate and recognition accuracy is the performance metric FM:

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

The algorithm was applied to 58 extremely degraded document from Cairo Genizah collection. The results averaged over these documents 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. In addition, the presented method does not require any preprocessing step, e.g., noise reduction or text zone location.

To test the applicability of the proposed approach to non Hebrew documents, we used Saint Gall and Parzival databases. The Saint Gall database (presented in *Fischer et al. (2011)* and *Garz et al. (2012)*) contains 60 pages of a Latin manuscript from the 9th century written by single writer. The Parzival database, described in *Fischer et al. (2012)*, contains 47 pages of a German manuscript from the 13th century written by three writers. The results of applying our algorithms are DR = 0.9784, RA = 0.8633, and FM = 0.9142 on Saint Gall database, and DR = 0.8106, RA = 0.8652 , and FM = 0.8363 on Parzival database. (*Garz et al. (2012)* used slightly dierent evaluation criteria. Without getting into details, our result for the Saint Gall dataset using their evaluation criteria is line accuracy 0.9784, while the result of Garz et al. is 0.9797. As can be seen, the results are very similar). In addition, we have applied the algorithm of *Asi, Saabni & El-Sana (2011)* on the Cairo Genizah dataset. While the algorithm in *Asi, Saabni & El-Sana (2011)* had been reported to achieve excellent results for documents of reasonable quality, it gave meaningless results on our extremely degraded dataset.

Stroke width estimation

Another application for utilizing evolution maps is evaluation of the range of stroke width in degraded documents. Many binarization methods use the stroke width as part of the binarization process (e.g., *Su, Lu & Tan, 2013*; *Liu & Srihari, 1997*; *Rivest-Hénault, Moghaddam & Cheriet, 2012*; *Badekas and Papamarkos, 2007*; *Ntirogiannis, Gatos & Pratikakis, 2009*). In extremely degraded documents, calculating the stroke width is hard and is highly influenced by noise and stains. The stroke width evolution map provides a statistical and comprehensive overview of the range of stroke widths revealed by the range of intensity thresholds.

To estimate the stroke width for a component, we first compute the component's distance transform, starting from the boundary of the component. We compute the

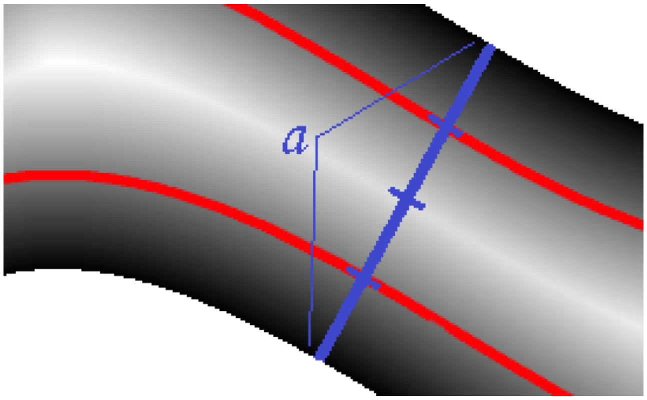


Figure 10 **Let *a* be the stroke width.** The maximal values the distance transform of the component achieves are around $a/2$. The average therefore can be used as an estimate for $a/4$.

average of the distance transform values inside the component and multiply it by four to receive an estimate for the average stroke width (see illustration in Fig. 10). A stroke width is consistent if it does not change much within the component. We compute a consistency factor for the stroke within of the component, using the standard deviation of the histogram of the component's distance transform. For each component we multiply its area by the stroke width consistency factor, so components with consistent stroke width will have a higher weight. In the CEM, we depict for each stroke width and intensity the sum of the weighted component areas.

We select the most salient blob on the map using techniques described earlier, and use the Gaussian fit of that blob as an estimate of the stroke width. In Fig. 11 we show an example document with its stroke width evolution map. In the map, the selected blob is marked with a white boundary. The estimated range of stroke widths is illustrated on the top left of the document using red lines. The left and rightmost lines are the range boundaries and the middle line shows the average stroke width detected.

To evaluate this method, we took fifteen degraded historical documents from the Cairo Genizah and sampled the stroke width manually in several random characters in the document. Over 92.5% of the samples across the documents where within the range of the estimate.

Analysis of feature distribution in documents

In many applications of document analysis, one of the first steps is looking for suitable features. The evolution maps provide a convenient tool for exploring and visualizing the behavior of different features in a given document. Using the maps, one can explore each feature, the distribution of its values, and view the elements of the document corresponding to different value ranges of that feature.

Based on the CEM, we developed an interactive tool which enables flexible generation and exploration of evolution maps per property, which we call the evolution map explorer. The user can interactively examine the maps, adjust the maps' display by setting properties for each map, and explore the maps. The user defines an area on the CEM with the mouse

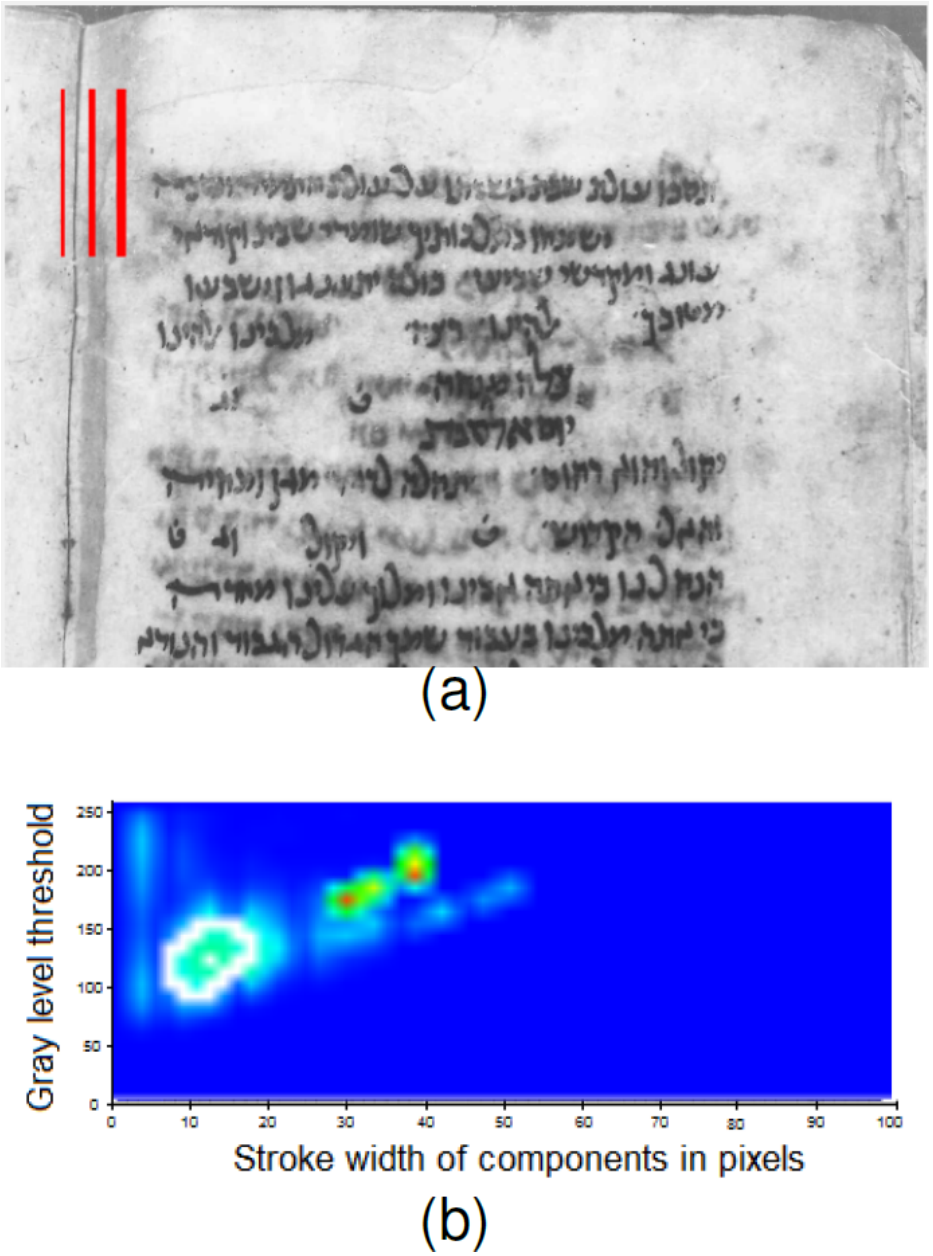
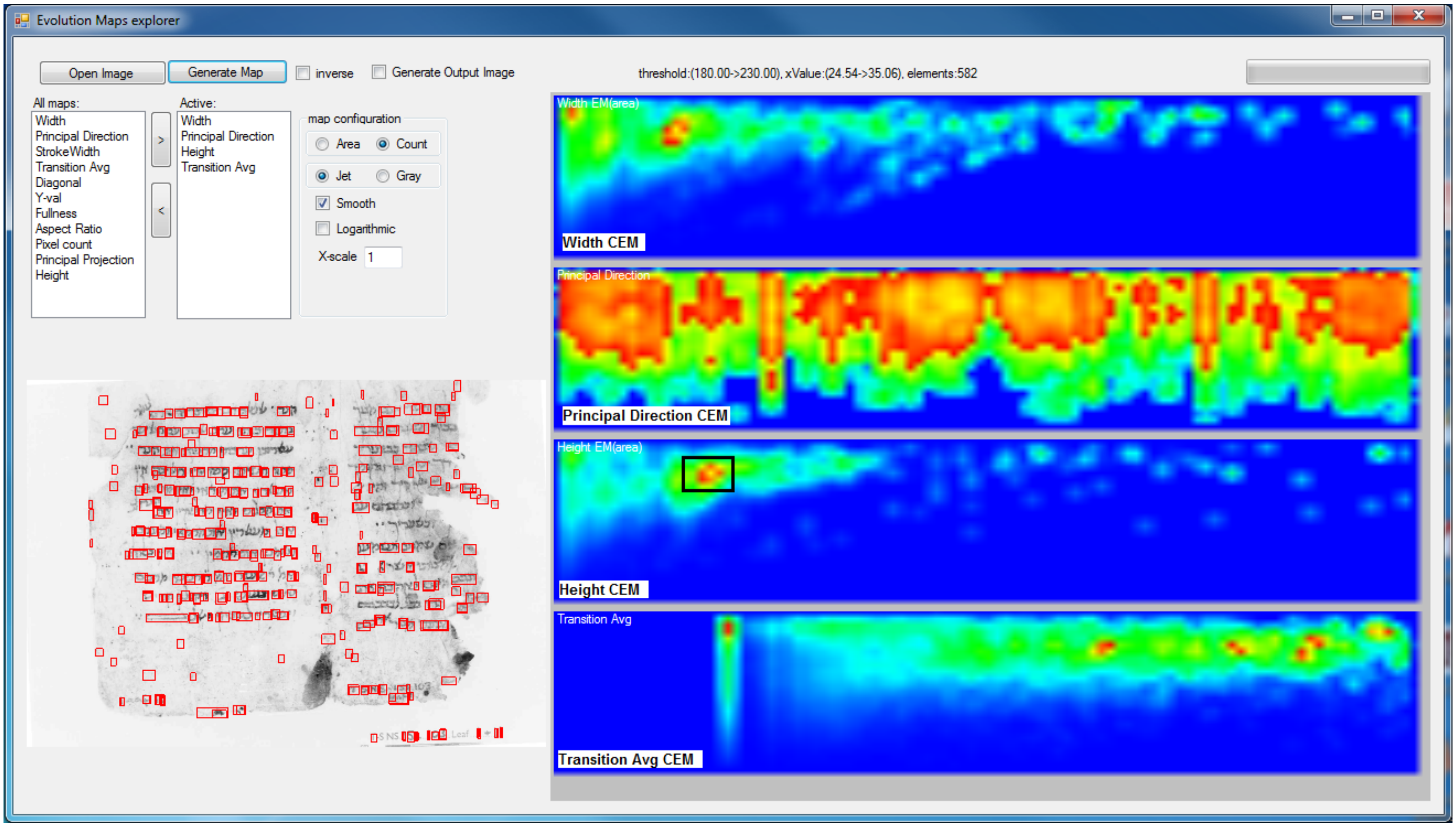


Figure 11 **(A) A document and (B) its stroke width evolution map.** The selected blob on the map (marked with a white border) is used to estimate the stroke width of letters in the document. The estimate is marked by red vertical lines over the document image. The leftmost and the rightmost lines are the range boundaries and the middle line shows the average stroke width detected.

(see, e.g., the black rectangle in the CEM in Fig. 12), and the system marks on the image of the document the elements corresponding to the rectangle, and displays information about the selection, e.g., the range of intensities, range of property values, and the count of elements relevant for this selection.

For example we picked a document and the feature "transition average." Our tool created the evolution map, shown in Fig. 13. In this evolution map, the *x* axis represents the


Evolution Maps explorer

□ inverse
□ Generate Output Image

threshold:(180.00->230.00), xValue:(24.54->35.06), elements:582

All maps:
Width
Principal Direction
StrokeWidth
Transition Avg
Diagonal
Y-val
Fullness
Aspect Ratio
Pixel count
Principal Projection
Height

Active:
Width
Principal Direction
Height
Transition Avg

map configuration
○ Area ● Count
● Jet ○ Gray
☑ Smooth
□ Logarithmic
X-scale 1

Figure 12 A general overview of the application. The four presented evolution maps are the width, principal direction, height and transition number average. The rectangle in the height CEM corresponds to the letters in the document on the left of the figure, whose heights and intensities correspond to the values in the rectangle.

average number of transitions from background to foreground for a connected component (averaging over the rows and columns of the component). The figure shows, for each selected rectangle on the CEM, the document and the elements corresponding to this rectangle. The leftmost rectangle represents noise elements which have average transition values around 1. The middle rectangle represents sets of characters with average transition values ranging from 2.9 to 4.4. The rightmost rectangle represents elements including more than one connected character with average transition values from 6.9 to 7.8. One can observe on the CEM (Fig. 13) a number of peaks which distinguish between characters with high average transition count and low one. By selecting different rectangles within this area, the user can explore the distribution of the different letters corresponding to the examined feature, and estimate the feature's ability to discriminate between different letters.

SUMMARY

In this work we introduced component evolution map which maps the evolution of a property of the connected components along the change in gray scale intensity level. We have demonstrated the contribution and potential of CEM in several tasks: estimating letter dimensions, improving a state of the art binarization algorithm, performing

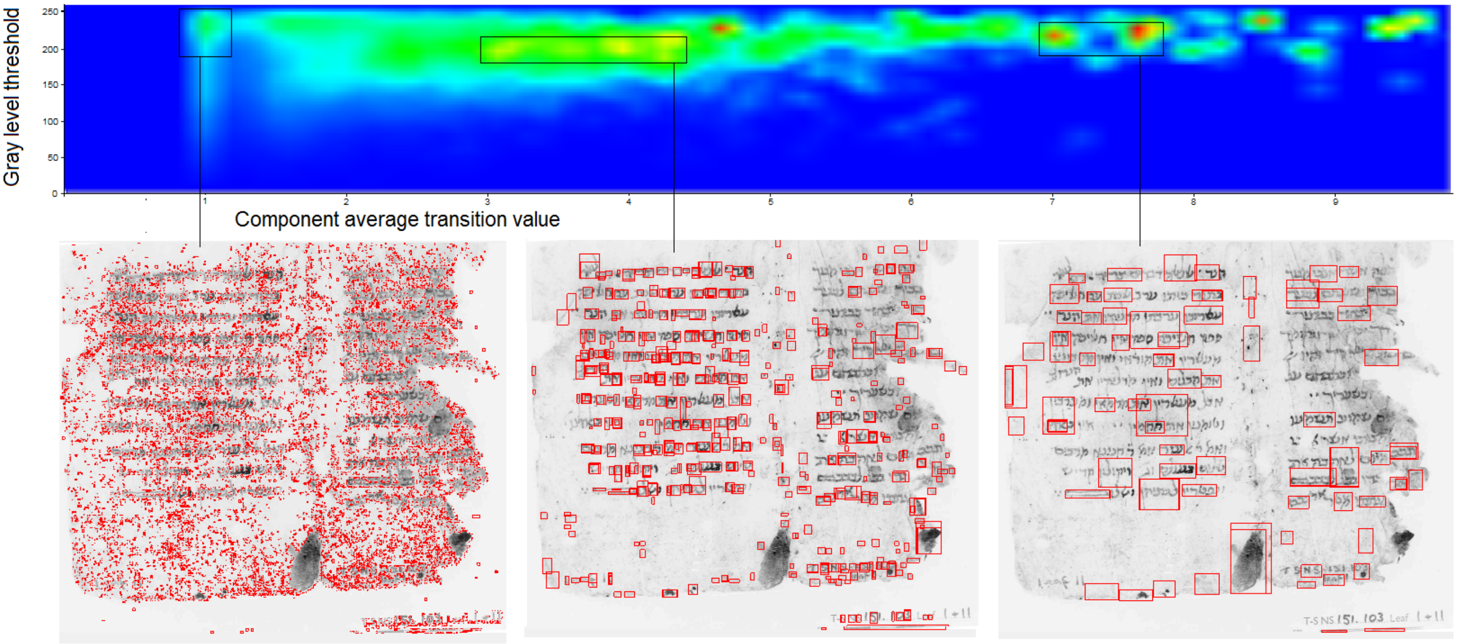


Figure 13 The evolution map of transition average for a document. Three selections of peek ranges (marked by black rectangles) in the evolution map, for each selection the elements corresponding to the selection are marked over the document. The leftmost range include transition values around 1, the middle selection contains transition values from 2.9 to 4.4, and the rightmost values between 6.9 to 7.8.

line segmentation in degraded documents, stroke width estimation, and analysis of feature distribution in text documents. This method is applicable for a wide range of text documents, and is especially capable of dealing with highly degraded and noisy documents.

We see in the CEM method potential for contribution in different directions in the document analysis field. We plan to continue looking for different ways to exploit the information gained by the CEMS, and also applying evolution maps on additional features. Among the uses we plan to examine in the usage of the maps as descriptor of a document for classification by writer or by origin manuscript (finding fragments of the manuscript). Furthermore, we plan to deal with limitations we took in this work such as regarding more than one writing style in a page. Another interesting direction we plan to investigate is using the maps for extracting automatically the degradation level of the document.

The documents used in our experiments are accessible via the Genizah project site (http://www.genizah.org/). To receive the ground truth data, please contact the authors by email.

ACKNOWLEDGEMENTS

This research was supported in part by the DFG-Trilateral Grant no. 8716. We thank Prof. Uri Ehrlich and Uri Safrai from the Goldstein-Goren Department of Jewish Thought, Ben-Gurion University of the Negev, for their assistance in generating the ground truth.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This project was funded by the German Research Foundation under contract FI 1494/3-2. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
German Research Foundation: FI 1494/3-2.

Competing Interests

Klara Kedem is an Academic Editor for PeerJ Computer Science.

Author Contributions

- Ofer Biller conceived and designed the experiments, performed the experiments, analyzed the data, contributed reagents/materials/analysis tools, wrote the paper, prepared figures and/or tables, performed the computation work.
- Irina Rabaev conceived and designed the experiments, performed the experiments, performed the computation work.
- Klara Kedem conceived and designed the experiments, analyzed the data, wrote the paper, reviewed drafts of the paper.
- Its'hak Dinstein and Jihad J. El-Sana conceived and designed the experiments, analyzed the data, reviewed drafts of the paper.

Data Availability

The following information was supplied regarding data availability:

The ground truth transcription for a set of document from the Genizah collection:

Named by catalog number, formated by WebGT, and detailed at http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6628633.

The data can be found at:

http://www.cs.bgu.ac.il/∼billero/GenizahDocumentsAnnotationData.zip.

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj-cs.39#supplemental-information.

REFERENCES

Asi A, Saabni R, El-Sana J. 2011. Text line segmentation for gray scale historical document images. In: *Proceedings of the 2011 workshop on historical document imaging and processing*. New York: ACM, 120–126. *Available at http://dl.acm.org/citation.cfm?id=2037362*.

Badekas E, Papamarkos N. 2007. Optimal combination of document binarization techniques using a self-organizing map neural network. *Engineering Applications of Artificial Intelligence* **20(1)**:11–24 DOI 10.1016/j.engappai.2006.04.003.

Bar-Yosef I, Beckman I, Kedem K, Dinstein I. 2007. Binarization, character extraction, and writer identification of historical hebrew calligraphy documents. *International Journal on Document Analysis and Recognition* **9(2)**:89–99 DOI 10.1007/s10032-007-0041-5.

Biller O, Asi A, Kedem K, El-Sana J. 2013. WebGT: an interactive web-based system for historical document ground truth generation. Technical Report 13–03. Be'er Sheva: Computer Science Department, Ben-Gurion University of the Negev, Israel.

Bukhari SS, Azawi MIAA, Shafait F, Breuel TM. 2010. Document image segmentation using discriminative learning over connected components. In: Doermann DS, Govindaraju V, Lopresti DP, Natarajan P, eds. *The Ninth IAPR international workshop on document analysis systems, DAS 2010*. New York: ACM, 183–190.

De Carvalho MAG, Da Costa AL, Ferreira ACB, Marcondes César Júnior R. 2010. Image segmentation using component tree and normalized cut. In: *SIBGRAPI*. Piscataway: IEEE Computer Society, 317–322.

Fischer A, Frinken V, Fornes A, Bunke H. 2011. Transcription alignment of latin manuscripts using hidden markov models. In: *1st international workshop on historical document imaging and processing (HIP)*. New York: ACM, 29–36.

Fischer A, Keller A, Frinken V, Bunke H. 2012. Lexicon-free handwritten word spotting using character hmms. *Pattern Recognition Letters* **33**:934–942 DOI 10.1016/j.patrec.2011.09.009.

Gatos B, Stamatopoulos N, Louloudis G. 2011. ICDAR2009 handwriting segmentation contest. *International Journal on Document Analysis and Recognition* **14(1)**:25–33 DOI 10.1007/s10032-010-0122-8.

Garz A, Fischer A, Sablatnig R, Bunke H. 2012. Binarization-free text line segmentation for historical documents based on interest point clustering. In: *Document Analysis Systems (DAS), 2012 10th IAPR international workshop*. Piscataway: IEEE, 95–99.

Jain AK, Zhong Y. 1996. Page segmentation using texture analysis. *Pattern Recognition* **29(5)**:743–770 DOI 10.1016/0031-3203(95)00131-X.

Liu Y, Srihari SN. 1997. Document image binarization based on texture features. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19(5)**:540–544 DOI 10.1109/34.589217.

Mosorov V, Kowalski TM. 2002. The development of component tree for grayscale image segmentation. In: *Proceedings of the international conference on moderns problems of radio engineering, telecommunications and computer science TCSET*. Piscataway: IEEE, 252–253.

Naegel B, Wendling L. 2010. A document binarization method based on connected operators. *Pattern Recognition Letters* **31(11)**:1251–1259 DOI 10.1016/j.patrec.2010.04.003.

New B, Ferrand L, Pallier C, Brysbaert M. 2006. Reexamining the word length effect in visual word recognition: new evidence from the English lexicon project. *Psychonomic Bulletin and Review* **13(1)**:45–5 DOI 10.3758/BF03193811.

Ntirogiannis K, Gatos B, Pratikakis I. 2009. A modified adaptive logical level binarization technique for historical document images. In: *10th international conference on document analysis and recognition*. Piscataway: IEEE, 1171–1175.

Pajdla T, Urban M, Chum O, Matas J. 2002. Robust wide baseline stereo from maximally stable extremal regions. In: *Proceedings of the British machine vision conference*. p. 3D and Video. *Available at http://cmp.felk.cvut.cz/~matas/papers/matas-bmvc02.pdf*.

Pikaz A, Averbuch A. 1996. Digital image thresholding, based on topological stable-state. *Pattern Recognition* **29(5)**:829–843 DOI 10.1016/0031-3203(95)00126-3.

Pratikakis I, Gatos B, Ntirogiannis K. 2010. H-DIBCO 2010—handwritten document image binarization competition. In: *ICFHR*. Piscataway: IEEE Computer Society, 727–732.

Rabaev I, Biller O, El-Sana J, Kedem K, Dinstein I. 2013. Text line detection in corrupted and damaged historical manuscripts. In: *12th international conference on document analysis and recognition (ICDAR'13)*. Piscataway: IEEE.

Raju SS, Pati PB, Ramakrishnan AG. 2004a. Gabor filter based block energy analysis for text extraction from digital document images. In: *Proceedings of the first international workshop on document image analysis for libraries (DIAL'04)*. Piscataway: IEEE Computer Society, 233. *Available at http://dl.acm.org/citation.cfm?id=968882.969393*.

Raju SS, Pati PB, Ramakrishnan AG. 2004b. Gabor filter based block energy analysis for text extraction from digital document images. In: *Document image analysis for libraries*. Piscataway: IEEE, 233–243.

Rivest-Hénault D, Moghaddam RF, Cheriet M. 2012. A local linear level set method for the binarization of degraded historical document images. *International Journal on Document Analysis and Recognition* **15(2)**:101–124 DOI 10.1007/s10032-011-0157-5.

Roy P, Pal U, Llados J, Delalandre M. 2009. Multi-oriented and multi-sized touching character segmentation using dynamic programming. In: *Proceedings of the 2009 10th international conference on document analysis and recognition, ICDAR '09*. Piscataway: IEEE Computer Society, 11–15.

Su B, Lu S, Tan CL. 2013. Robust document image binarization technique for degraded document images. *IEEE Transactions on Image Processing* **22(4)**:1408–1417 DOI 10.1109/TIP.2012.2231089.

Wen D, Ding X. 2004. A general framework for multicharacter segmentation and its application in recognizing multilingual Asian documents. In: Smith EHB, Hu J, Allan J, eds. *Proceedings of the SPIE conference on document recognition and retrieval XI*, vol. 5296, Bellingham: SPIE, 147–154. *Available at https://www.msu.edu/~wendi/publication/DRRXI_Manuscript.pdf*.

Zagoris K, Papamarko N. 2010. Text extraction using document structure features and support vector machines. In: *Proceedings of the 11th IASTED international conference computer graphics and imaging (CGIM 2010)*. Calgary: IASTED, 88–91.