Evolution maps for connected components in text documents

Ofer Biller, Klara Kedem, Itshak Dinstein, Jihad El-Sana
Ben-Gurion University, Beer-Sheva, Israel
billero,klara,el-sana@cs.bgu.ac.il, dinstein@ee.bgu.ac.il

Abstract

For highly degraded text documents, common tasks such as binarization and line extraction, remain difficult tasks. Equipped with a reliable information regarding the distribution of character dimensions in the document, one can improve results of these algorithms significantly. We introduce a novel perspective of the image data which maps the evolution of connected components along the change in gray scale threshold. We use these maps to provide a robust algorithm for extracting information about character dimensions in degraded documents, and demonstrate improvement in binarization results using this information. We analyze statistically the characteristics of the evolution maps for text documents, and compare our results with ground truth data.

1 Introduction

Various algorithms for historical document processing, such as binarization, segmentation, word spotting, and recognition, often require preliminary information about the input document, such as character dimensions, stroke width, and noise types. When dealing with severely degraded documents, this information is usually more crucial than for less degraded documents, but unfortunately more difficult to obtain automatically. Parameters of stroke width and character size are often obtained relatively accurately using a well binarized version of the document [11, 12, 9]. Information regarding character dimensions is often determined in an ad-hoc manner or expected to be provided by the user [6, 1].

The binarization of very degraded documents is not very reliable as it may introduce noise, falsely merged components, and other artifacts. To overcome this limitation we analyze gray level text documents in order to automatically provide fundamental parameters. We supply information such as character sizes and their gray level span as a preprocessing step to improve and fully automate image processing algorithms for historical documents.

In our work we estimate character dimensions and character intensity distribution by analyzing the distribution of the character's properties, e.g., width and height of connected components, for each possible gray-scale threshold. We generate a histogram for each property of the connected components, so that we can observe the evolution in the distribution of this property along the change in the intensity threshold. To simplify the discussion, from now on we will explain only the width evolution maps. For example, in the text document in Figure 1, we display 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 amount (density) of the components for each width in the given threshold (warm – high value, cold – low). Since the map represents a text document, we expect to see high density in the range of character width and height, and the range of intensity thresholds which discriminate the characters from the background. The histogram in Figure 1 shows a blob centered around 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 depicts a high count of narrow connected components in a wide range of gray scale thresholds.

The presented method supplies required parameters for various applications, such as binarization, line segmentation, word spotting, and recognition, in documents with high degradation and non-uniform background. The method is robust in presence of noise, local distortions, and various types of document transformations, therefore, it can be applied automatically on large collections of documents of high variety.

In the next section we describe the background for this work. In section 3 we describe in details the component evolution maps, analyze them, and extract automatically the dimension of characters. In section 4

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.201

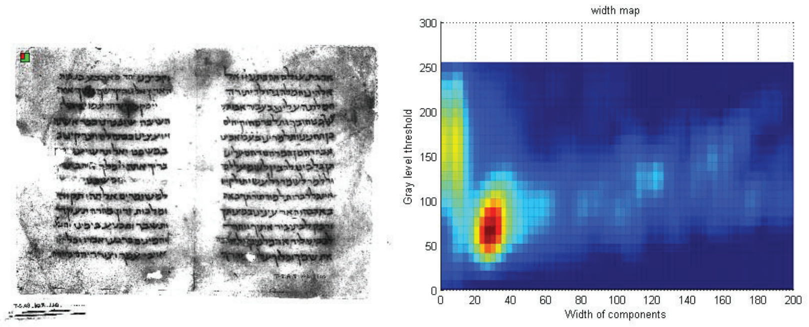


Figure 1. A document image, and the distribution of connected components along their width property (x-coordinate), for each possible gray scale threshold (y-coordinate).

we describe our experiments of detecting character size in highly degraded documents, demonstrate utilization and evaluation of our results. Finally, in section 5 we draw conclusions and outline future work.

2 Related Work

In this research we map the evolution of connected components in degraded gray scale text documents in order to provide information regarding letter dimensions for higher level algorithms. An intensive usage of connected components analysis was performed in work on document layout segmentation and text separation for binarized document images (e.g. [10], [13], [3]).

The component tree is a graph image representation computed from the cross-section decomposition of the image gray levels [4]. As in our work it uses connected components of gray level cross-sections of the image to assemble a different perspective of the image data. While our method provides a general perspective about the image and analyzes character attributes and noise, the component tree applies operators on the single component level for image segmentation [2] and for document binarization [5].

Pikaz and Averbuch [7] select 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 remain stable. This method may be viewed as a special case of our approach.

3 Our Approach

Historical documents are usually highly degraded, due to aging and storing conditions, which complicates common tasks such as binarization, text line extraction and pattern recognition. Reliable preliminary information concerning character dimension is necessary for these processing tasks. In this work we present an elegant approach to compute the range of character dimensions for degraded historical documents. We offer a novel perspective of the image data which reflects the evolution of connected components along the change in gray scale threshold. We refer this perspective as *Component Evolution Maps* (CEM). Next we discuss in detail the construction of these evolution maps followed by analysis and applications of the maps.

3.1 Evolution map for connected components

The *evolution map* is a function of the intensity and a property of the image to 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, when thresholding the image at intensity g. To simplify the discussion, we refer to width CEM in a gray scale image. CEM provides an intuitive visualization tool to analyze the distribution of an image property. Figure 1 represents the width CEM for a 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 comonents' widths in the image binarized using g as a threshold. Figure 2 shows three cross sections at different thresholds, and the generated corresponding binary images. Figure 3 shows the cross section as histogram of component width for a specific threshold, and the corresponding components of the document for two ranges of width (see the squares on the graph).

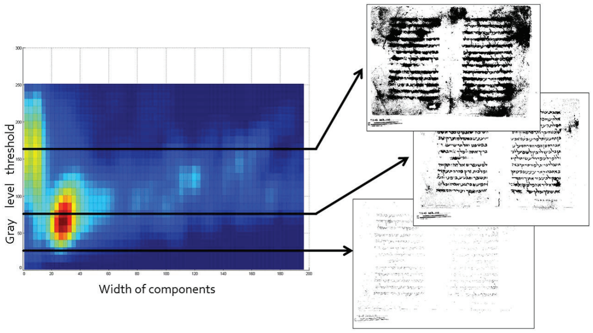


Figure 2. Horizontal cross section of the component width evolution map and the related binary images.

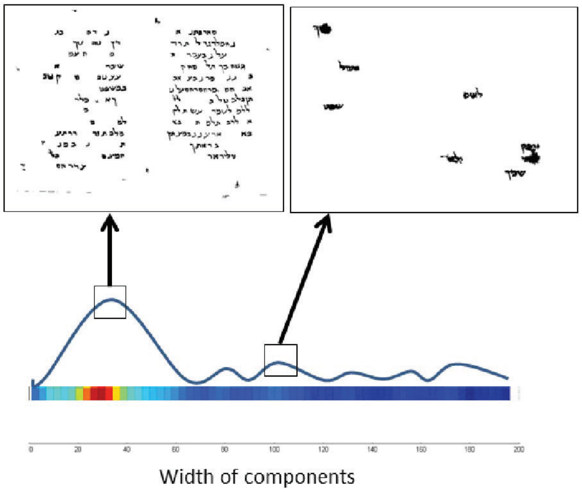


Figure 3. A horizontal cut of the width CEM at one gray level value. The squares on the graph depict the most popular widths corresponding to the component images above.

3.2 Construction of the evolution map

In our current implementation, we build an evolution map in a straightforward manner by thresholding the image document over gray levels, and counting the number of resulting components for each width in the binary image. Since noise usually produces a vast number of components and skews the histogram, we accumulate instead of component count the *relative total area* of the components. The relative total area for components of width w at threshold g is the sum of their areas divided by the area of the whole image. This value is less sensitive to noise than component count, as shown in Figure 4. As seen, the red blob in the left illustration emphasizes the noise, and the right highlights the sought properties. We smooth the CEM using a Gaussian kernel to reduce the influence of local irregularities.

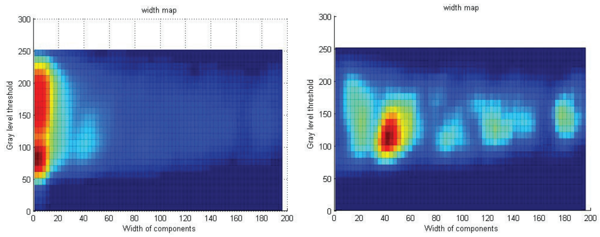


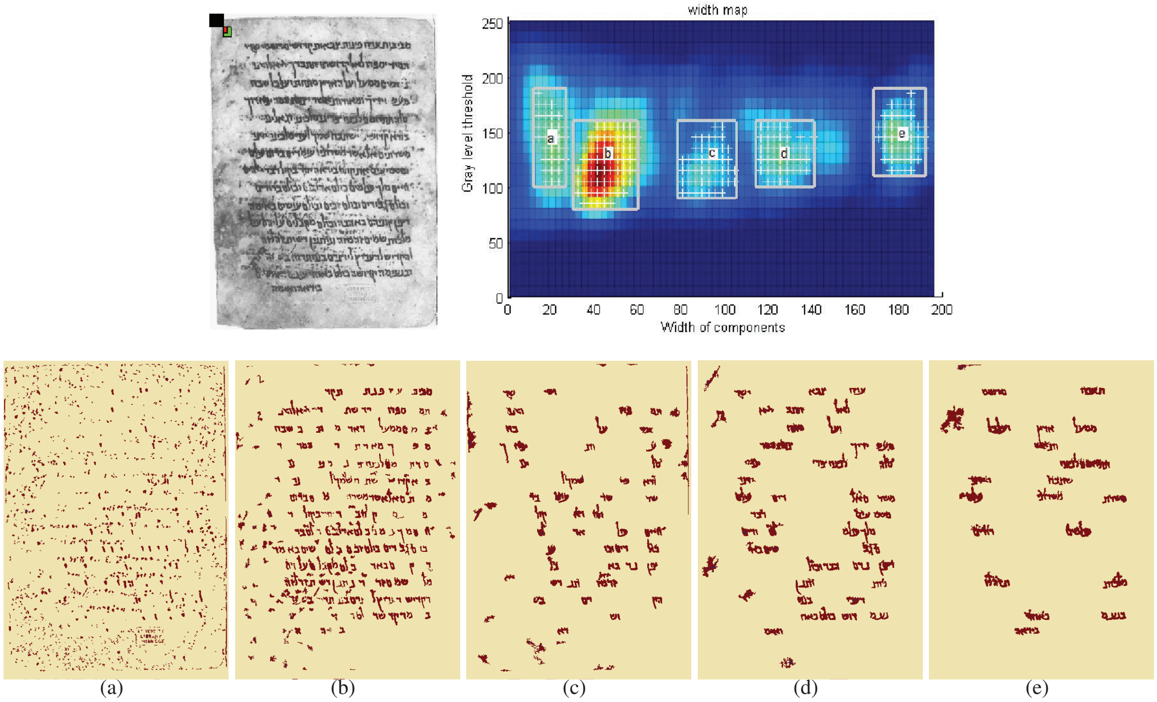
Figure 4. Left map displays the count of components per threshold and width, and the right map shows the total relative area of components per threshold and width.

3.3 Analysis of the CEM

Considering the CEM as a height map, a local maximum indicates the existence of objects (noise, character, connected characters, etc.) in the image. The characteristics of the local maxima and their neighborhoods, can indicate attributes of the object sets represented by the neighborhoods. Figure 5 shows a document with its width CEM. The images a-e show the set of elements on the original image corresponding to the blobs a-e in the width CEM. Blob a represents a set of noise stains in the document, which are received 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. The blobs c, d, and e, received at high width values and represent objects of two, three, four connected letters, respectively.

3.4 Extracting character dimensions

As seen, the CEM brings to the surface the underlying information about the text components in the document image. One of the salient elements in a CEM of a text document is the blob which represents the documents' characters. To determine the blobs we use a sweeping plane that moves downward along the z-axis. As the plane descends, it encounters the peeks of the blobs. Neighboring blobs expand until they touch each other, or reach a low value beneath a predefined threshold. Next we extract the following characterizing features for each blob: The percentage of the image area,



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

Figure 5. A document image with its components 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).

p, occupied by the components corresponding to all the points (g,w) in the blob. The total number of connected components represented by the blob, n. Average components' density factors, d, for all components represented by the blob. Density factor is defined as the percent of the area which the component occupies in its bounding box. This factor tends to have high values for random noise elements.

We use the features above to generate a score for each blob, by linear combination of p and d, multiplied by a logistic function of n:

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

The contribution of the logistic function suppresses the score for blobs with small number of components. Currently we set both the coefficients and the general form of the formula empirically. In the future we plan to use a learning algorithm to automate the scoring process. We select the blob with the highest score from each width and height CEMs to indicate the width and height of letter respectively. An additional parameter we take into account at this selection is the agreement in gray levels of the chosen width and the height blobs. Both blobs should have a similar mean and standard deviation on the y-axis projection.

The final refinement of the boundaries of the blob uses the evolution map with count of elements instead of the total relative area. While the total relative area map gives a much better overview of the elements in the document it has a displacement artifact due to the area covered by the components.

The mean intensity of the selected blobs is a good threshold candidate for global binarization. Although global binarization is obviously not an acceptable solution for degraded documents, in many cases it is used as an initial step in more sophisticated algorithms.

4 Experimental Results

We applied our method on a collection of digitized historical documents from the Cairo Genizah. We ran our algorithm on a set of seventy documents with various degradation conditions, resolutions and character dimensions. We created width and height CEMs for each of these documents, and estimated the width and height ranges of the characters in each document. For ten of these documents we generated ground truth data, which specifies the bounding box of each letter (over all, more than 4000 letters).

4.1 Applying our results for document binarization

We demonstrate that the information provided by our method improves the results of binarization algorithm. The algorithm we chose was the state of the art binarization method developed by Bar-Yosef at al [1] which won the binarization contest H-DIBCO-2010 [8]. This binarization algorithm is based on the sliding window approach. The window size is configurable, and recommended by the author to be the size which is relative to the dimensions of letters in the document. First we ran the binarization algorithm with a default window size (Figure 6(a)). Then we ran the same binarization algorithm using window size of 1.5 times the maximum of the width and height of character in the document given by our algorithm (Figure 6(b)). As an additional improvement of the binarization, we filter out all components of the binarized image which substantially exceed the range of character dimensions as computed by our algorithm (Figure 6(c)). This experiment was run on seventy documents, of different dimensions, resolution and degradation state, all from the collection of the Cairo Genizah. A substantial improvement in binarization results were shown over the entire tested documents.

4.2 Evaluation

To measure the accuracy of our results we calculated the precision, recall, and f-measure of our estimation with respect to the ground truth. Figure 7 shows the histogram of the widths of the characters for a specific document (by our ground truth) displayed in black over our width CEM. Our estimation of character width is marked by a white rectangle. We calculate the recall as the ratio between the number of letters in the ground truth which are inside our estimation's width range, and the total number of ground truth letters. The precision is calculated as the ratio between the overlap of our width estimation with the ground truth, and the range of our width estimation. For all the documents in the test set our method selected the correct blobs in the CEM to present the characters, and gave an estimation with an average accuracy (in f-measure) of 84.9% with respect to the exact boundaries provided by the ground truth.

To evaluate the robustness of our method for a larger set of documents (70 documents), we selected manually, for each document, a central point in the blobs corresponding to the correct width and height CEM in the document. For 94% of the documents our algorithm found the correct blobs.

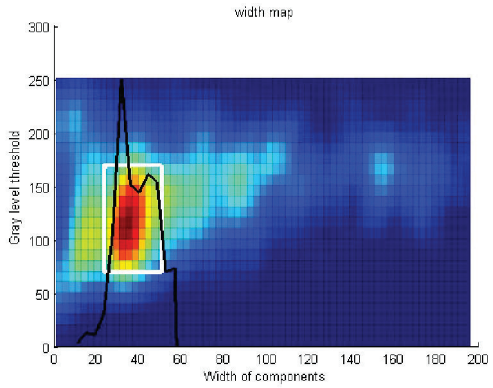


Figure 7. A histogram (the black curve) of the ground truth characters widths, displayed on the width CEM. Our character width estimation is marked by a white rectangle.

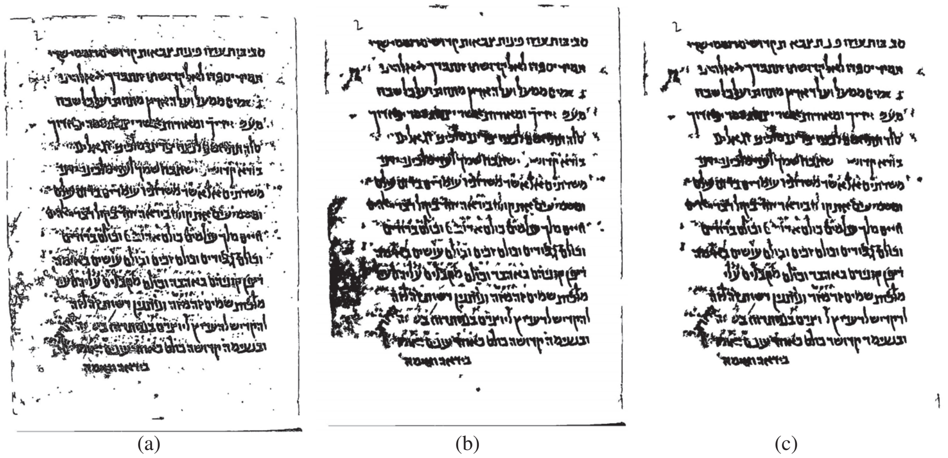
5 Conclusions and Future work

In this work we map the evolution of the dimensions of connected components along the change in gray scale threshold level. We use the CEMs to extract fundamental data on documents of high degradation. We provide valuable information to higher level algorithms, which improves their results. This method is applicable for a wide range of degraded and noisy documents. We have demonstrated the contribution of estimating letter dimensions for improving a state of the art binarization algorithm.

We believe that CEMs hold great potential and can be developed in several directions, we will explore in the future: (i) Analysis of noise types and amount in the document image. (ii) Analysis of palimpsest document and documents with bleed through. (iii) Using CEMs for measuring documents' similarity. (iv) Using learning algorithms to automate score generation for the blobs' score.

Acknowledgment

This research was supported in part by the Israel Science Foundation grant no. 1266/09, DFG-Trilateral Grant no. 8716, the Lynn and William Frankel Center for Computer Sciences, and the Paul Ivanier Center for Robotics and Production Management at Ben-Gurion University, Israel. We also 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.



(a) (b) (c)

Figure 6. Utilizing our estimation to improve binarization algorithm. (a) Binarization using default window size. (b) Binarization with optimized window size using the output of our method. (c) Result after filtering out-of-range components, using the estimation of character dimension, given by our method.

References

[1] I. Bar-Yosef, I. Beckman, K. Kedem, and I. Dinstein. Binarization, character extraction, and writer identification of historical hebrew calligraphy documents. *International Journal on Document Analysis and Recognition*, 9(2):89–99, 2007.

[2] M. A. G. de Carvalho, A. L. da Costa, A. C. B. Ferreira, and R. M. C. Júnior. Image segmentation using component tree and normalized cut. In *SIBGRAPI*, pages 317–322. IEEE Computer Society, 2010.

[3] A. K. Jain and Y. Zhong. Page segmentation using texture analysis. *Pattern Recognition*, 29(5):743–770, May 1996.

[4] V. Mosorov and T. M. Kowalski. The development of component tree for grayscale image segmentation. In *Proc. of International Conference on Moderns Problems of Radio Engineering, Telecommunications and Computer Science TECSET, Slavsko Ukraine*, pages 252–253. TECSE, 2002.

[5] B. Naegel and L. Wendling. A document binarization method based on connected operators. *Pattern Recognition Letters*, Aug. 2010.

[6] B. New, L. Ferrand, C. Pallier, and M. Brysbaert. Reexamining the word length effect in visual word recognition: New evidence from the english lexicon project. *Psychonomic Bulletin and Review*, 13(1):45–5, 2006.

[7] A. Pikaz and A. Averbuch. Digital image thresholding, based on topological stable-state. *Pattern Recognition*, 29(5):829–843, May 1996.

[8] I. Pratikakis, B. Gatos, and K. Ntirogiannis. H-DIBCO 2010 - handwritten document image binarization competition. In *ICFHR*, pages 727–732. IEEE Computer Society, 2010.

[9] S. S. Raju, P. B. Pati, and A. G. Ramakrishnan. 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)*, DIAL '04, pages 233–, Washington, DC, USA, 2004. IEEE Computer Society.

[10] S. S. Raju, P. B. Pati, and A. G. Ramakrishnan. Gabor filter based block energy analysis for text extraction from digital document images. In *Document Image Analysis for Libraries*, pages 233–243, 2004.

[11] P. Roy, , U. Pal, J. Llados, and M. Delalandre. Multioriented and multi-sized touching character segmentation using dynamic programming. In *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, ICDAR '09, pages 11–15, Washington, DC, USA, 2009. IEEE Computer Society.

[12] D. Wen and X. Ding. A general framework for multicharacter segmentation and its application in recognizing multilingual asian documents. In E. H. B. Smith, J. Hu, and J. Allan, editors, *DRR*, volume 5296 of *SPIE Proceedings*, pages 147–154. SPIE, 2004.

[13] K. Zagoris and N. Papamarko. Text extraction using document structure features and support vector machines. In *Proceedings of the 11th IASTED International Conference Computer Graphics and Imaging(CGIM 2010)*, pages 88–91, 2010.