Open access • Proceedings Article • DOI:10.1145/2501115.2501117

Robust text and drawing segmentation algorithm for historical documents — Source link

Rafi Cohen, Abedelkadir Asi, Klara Kedem, Jihad El-Sana ...+1 more authors

Institutions: Ben-Gurion University of the Negev

Published on: 24 Aug 2013

Topics: Scale-space segmentation, Segmentation-based object categorization, Image segmentation, Segmentation and Historical document

Robust Text and Drawing Segmentation Algorithm for Historical Documents

Rafi Cohen*, Abedelkadir Asi*, Klara Kedem, Jihad El-Sana
Computer Science Department
Ben-Gurion University of the Negev
{rafico,abedas,klara,el-sana@cs.bgu.ac.il}

Itshak Dinstein
Department of Electrical and Computer Engineering
Ben-Gurion University of the Negev
dinstein@ee.bgu.ac.il

ABSTRACT

We present a method to segment historical document images into regions of different content. First, we segment text elements from non-text elements using a binarized version of the document. Then, we refine the segmentation of the non-text regions into drawings, background and noise. At this stage, spatial and color features are exploited to guarantee coherent regions in the final segmentation. Experiments show that the suggested approach achieves better segmentation quality with respect to other methods. We examine the segmentation quality on 252 pages of a historical manuscript, for which the suggested method achieves about 92% and 90% segmentation accuracy of drawings and text elements, respectively.

Categories and Subject Descriptors

I.7.5 [**Document Capture**]: Document analysis

General Terms

Algorithms, Performance

Keywords

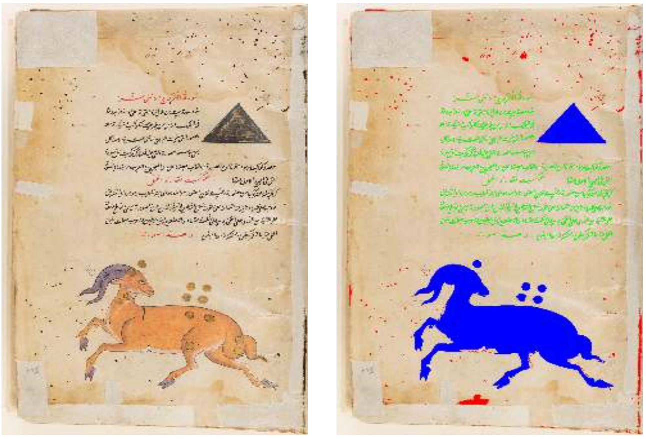
Segmentation, layout, historical documents, superpixel, CRF.

1. INTRODUCTION

Grouping documents into regions of different content plays a powerful role in human visual perception [29]. A human would invest minor efforts to perceive global aspects of the image and consequently segmenting it into regions, such as text and images. For computers, on the other hand, reliable and efficient content-based segmentation remains a great challenge.

Page layout analysis is a fundamental step of any document image understanding system. The analysis process consists of two main steps, page segmentation and block classification. In the first step a document image is segmented into homogeneous regions. The classification step attempts to distinguish among the segmented regions whether they are text, image, drawing, etc. Each region is fed into an appropriate algorithm, according to the type of the region, for further processing.

Historical documents may contain drawings in addition to text and ornamentation as appears in Figure 1(a). Separating text from these documents significantly contributes to word-spotting techniques [2, 21]. On the other hand, localizing drawings and classifying them, as shown in Figure 1(b), can expedite their automatic retrieval; as a result, the manuscript authentication process [20] can be facilitated.



(a) (b)

Figure 1: (a) Arabic historical document image with text and drawings. The manuscript dates back to the 17^{th} century and it is from the IHP collection [13], (b) segmentation result of text (in green), drawings (in blue) and noise (in red).

Using pixel grid to compute features from individual pixels can be computationally expensive. To overcome this limitation, researchers suggest compact representations of images [1, 16], which are adapted to the local structure of the image. Superpixels are a good example of such a representation which we are using in our algorithm. This representation is obtained by a conservative over-segmentation of the image into compact and highly uniform regions. Superpix-

*Corresponding authors

HIP '13 August 24 2013, Washington, DC, USA
Copyright 2013 ACM 978-1-4503-2115-0/13/08 ...$15.00
http://dx.doi.org/10.1145/2501115.2501117.

els are the elementary units from which local image features can be computed. They have proved increasingly useful in computer vision tasks as they greatly reduce the complexity of subsequent image processing steps [19, 14].

In this paper we present a method for separation between text and drawings in historical documents. In the first stage a classifier is used to separate text from non-text elements in a binarized version of the document. In the second stage we remove the text from the document and use another classifier to separate drawings from background and noise. We exploit both spatial and color features of superpixels to extract drawings. Both stages use *Conditional Random Fields* (CRFs) to enforce spatial coherence in the final segmentation.

2. RELATED WORK

In this section we briefly review existing image segmentation techniques in general, and document image segmentation methods in particular. Approaches for document image segmentation can be divided into two main categories: top-down and bottom-up. In the top-down category, the image is coarsely segmented and a subsequent refining process is applied as a second step. In the bottom-up category, elementary units in the image (such as pixels, connected-components, superpixels, etc.) are aggregated to form larger regions which define different image classes. The aggregation process usually optimizes a cost function to meet a specific criterion.

2.1 Bottom-up Approaches

Graph-based algorithms constitute the large portion of this category. In these algorithms, each elementary unit is treated as a node in a graph, and edge weight between two nodes is determined according to a similarity measure. Shi and Malik [26] addressed image segmentation as a graph partitioning problem using the Normalized Cuts global criterion. The minimization of this criterion can be formulated as a generalized eigenvalue problem. Following a recursive scheme, the eigenvectors were used to generate good partition of the image. Felzenszwalb and Huttenlocher [11] presented a greedy algorithm that clusters pixel nodes in the graph into segments. Each segment is the minimum spanning tree of the constituent pixels. Recently, Kim *et al.* [16] proposed an approach that utilizes a modified version of the correlation clustering algorithm for image segmentation. They applied it on a pairwise superpixel graph to merge superpixels into homogeneous regions. A higher-order correlation clustering was applied on the induced hypergraph as well. Structured support vector machine (S-SVM) [27] was used for training of parameters in correlation clustering.

Bukhari *et al.* [5] proposed a method for text and non-text segmentation based on connected components. They classified connected components according to a set of simple and representative features. A multi-layer perceptron classifier was exploited. In [6] the authors extended the former approach to segment sidenotes in page margins from the main-body text in historical documents.

Dan Bloomberg introduced a simple and effective approach for page segmentation based on multi-resolution morphology [3]. This method aims at separating halftone figures from text. He managed to segment halftones from text by using a combination of dilations and erosions. Since it is expensive to use large structuring elements at high image resolution to complete the missing parts of the halftone, he introduced the interesting concept of multi-resolution morphology. However, the considered algorithm cannot segment non-text elements such as drawings, graphs, maps, etc. from text components. Bukhari *et al.* [7] improved the former method so that it can cope with non-text components including halftones, drawings, graphs, maps, etc. The improvement relies heavily on combinations of basic morphological operations. These combinations lead to two new modifications: hole-filling and reconstruction of broken drawing lines. These modifications generalize Bloomberg's work so that it can segment more non-text components from documents.

2.2 Top-down Approaches

Mean-Shift [8], a feature-space analysis technique, was used in the domain of computer vision to generate homogeneous clusters. It locates peaks in high-dimensional density functions without computing the complete function explicitly. It generates image segments by grouping pixels in the feature space that climb to the same peak in the density function. Wang *et al.* [28] used an optimized decision tree classifier to classify components of different target classes. They introduced a block classification system in which a block is represented by a 25 dimensional feature vector. Later, Keysers *et al.* [15] showed that a document block classification system can be constructed by merely using run-length histogram feature vectors. They managed to segment several classes such as math, logo, text, table, drawing, halftone, ruling and speckles. However, block-based approaches are relatively limited due to their sensitivity to the accuracy of page segmentation into homogeneous blocks.

3. THE PROPOSED METHOD

Our method is based on a two stage bottom-up approach. First, we segment the text from a binarized version of the document. We utilize shape features extracted from the connected components, and use CRFs to enforce spatial coherence. Second, drawings are extracted by exploiting features from superpixels such as the spatial location and the CIE-Lab color distribution [8]. We define drawings as a group of adjacent pixels that significantly differ in color from the background and occupy a relatively large portion of the document. We use the former definition to assign appropriate probabilities to superpixels. Spatial coherence is guaranteed by applying CRF on the considered superpixels.

3.1 Text Segmentation

The main features that characterize text with respect to drawing in a document are its - relatively constant - stroke width [10], size and texture. Text elements form virtual horizontal lines, which can characterize text as well. Candidate horizontal text lines are extracted using multi-scale texture analysis and used to learn the shape and location of text elements. These candidates guide the extraction of text within the document. The former process is repeated, guided by the text lines extracted at each iteration, until no more text lines are found.

The estimation of text lines is based upon a multi-scale analysis of the document, where each scale corresponds to an estimated text line height. The local orientation of the pixels in a given scale is extracted in the following way:

1. We convolve the image with different filters, 6 of them are anisotropic Laplacian of Gaussian (LoG) at different orientations, and the 7th is an isotropic Gaussian. All of the filters are in the same scale.
2. The orientation of the filter with the strongest response determines the orientation of each pixel. This produces the orientation map (as shown in Figure 2(a)).

It has been demonstrated that LoGs robustly detect line elements within images [12]. We use them to extract horizontal lines, as they are reliable candidates of text lines. The candidate text lines of a given scale are extracted from the obtained orientation map.

For each scale, the connected components (CCs) which constitute of pixels with horizontal orientation, represent text lines, and other elements that we regard as noise (as shown in Figure 2(b)). Separation between text lines and noise is done based on the aspect ratio between the width and height for each line. Text lines are chosen as the elongated lines, using K-means (as shown in Figure 2(c)). After this initial separation between text lines and noise, we select the text lines from the scale which produces the most *regular* text lines. The irregularity for text lines is defined as the sum of the variances of the *mean stroke width*, the *maximal vertical run-length* and the Euclidean distance between each pair of neighbouring text lines. The stroke width and maximal vertical stroke width features are explained below.

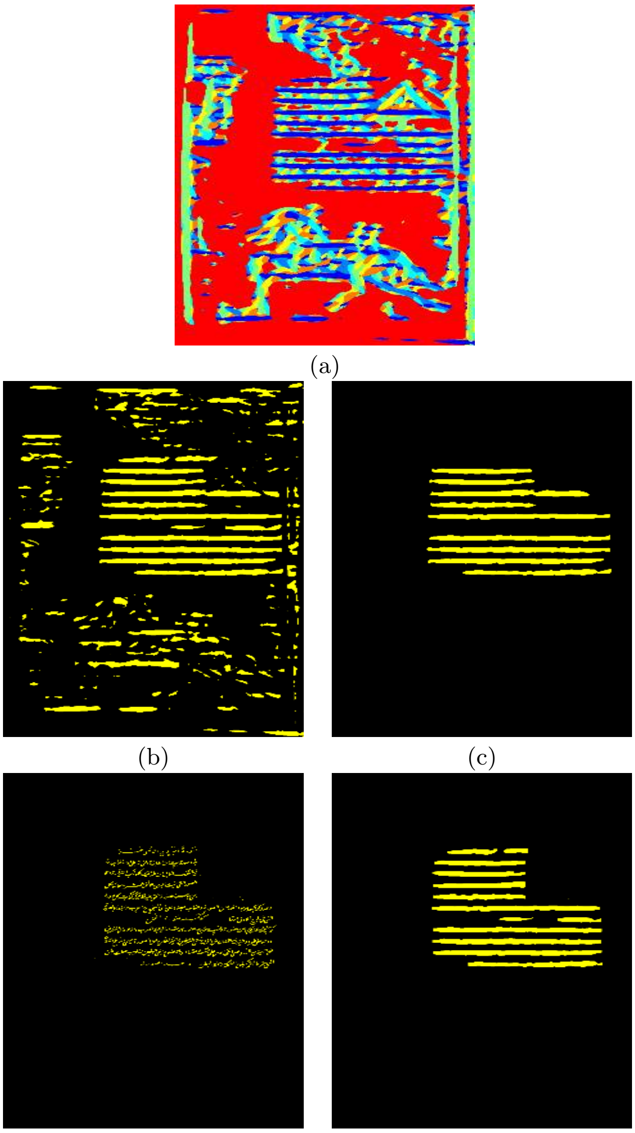
The estimated text lines are used to guide the extraction of text from a binarized image. After binarizing the document using a standard text segmentation technique [22], the CCs (henceforth elements) which overlap with the estimated text lines are being used to characterize the shape and location of text elements. For each element that overlaps with an estimated text line a shape related feature vector is extracted. The feature vectors are modeled using a multivariate normal distribution $\mathcal{N}(\vec{\mu}, \Sigma)$ in order to estimate the probability for all elements within the document.

We use the following features to generate feature vectors for each element:

1. **Bounding Box Features:** the height and width of the bounding box.
2. **Area:** The number of pixels contained in the element.
3. **Stroke Width Features:** The mean stroke width is obtained by

   $$2 * \frac{|S|}{|D|},$$

   where $|S|$ is the number of foreground pixels, and $|D|$ is the number of pixels on the boundary [18]. In addition, we measure stroke width by counting the number of consecutive foreground pixels in a given direction, i.e., run-length. A histogram of run-length measurements indicates the number of runs of each length [9].



(a)

(b)

(c)

(d)

(e)

Figure 2: (a) The orientation map for 10 pixels scale; each color represents a different orientation; the blue color represents pixels with horizontal orientation. (b) and (c) depict the initial set of horizontal lines and the set after K-Means, respectively. (d) depicts the corresponding text. (e) depicts the estimated text lines after one iteration.

   We generate horizontal and vertical run-length histograms, and from each histogram we extract the mode, mean, variance and maximal run-length.

4. **Distance to estimated text lines:** the distance of an element (e) to the text lines (ℓ) is defined in Eq. (1), where v_i and v_j are pixels in e and ℓ, respectively.

   $$d(e, \ell) = \max_{\forall v_i \in e} \{ \min_{\forall v_j \in \ell} ||v_i - v_j||_2 \} \qquad (1)$$

We formulate the problem of labeling each element as text or non-text as an energy minimization problem [4], where the energy function consists of data and smoothness costs.

The data cost $D_p(f_p)$ measures how well an element p fits to the text model. The data cost of assigning the text label for p is defined in Eq. (2), while $D_p(f_p = \text{non-text}) = const.$ This constant value acts as a tolerance threshold. The lower the value, the more confidence we require from the normal distribution in order to assign a text label for an element. The value n in Eq. (2), is the length of the feature vector, and $\vec{p}$, is the feature vector of element p.

$$D_p(f_p = \text{text}) = \ln(\sqrt{(2\pi)^n|\Sigma|}) + \frac{1}{2}(\vec{p} - \vec{\mu})^T\Sigma^{-1}(\vec{p} - \vec{\mu}) \quad (2)$$

The smoothness term $V_{pq}(f_p, f_q)$ measures the spatial correlation of neighboring elements. Elements with a smaller distance have a higher probability to belong to the same label than those distant ones. This is defined in our energy minimization scheme as the smoothness term. Kubovy and Van den Berg [17] showed that proximity grouping strength decays exponentially with Euclidean distance. This reflects in the smoothness energy term which is defined in Eq. (3). The term $dis(p, q)$ is the Euclidean distance between p and q, and the constant α is defined as $(2 \langle dis(p, q)\rangle)^{-1}$, where $\langle\cdot\rangle$ denotes expectation over all pairs of adjacent elements [23].

$$V_{pq} \quad = \quad \exp(-\alpha \cdot dis(p, q)) \qquad (3)$$

Now that the energy model is fully defined, the segmentation can be estimated as a global minimum of Eq. (4), where CCs is the set of all elements, N a set of adjacent elements, and f_p the label assigned to element p. We define the neighborhood of an element as consisting of its 8 closest elements. Finding a solution to this labeling problem is optimized using a standard graph-cut algorithm as proposed by [4] (see results in Figure 2(d)).

$$E(f) = \sum_{p \in CCs} D_p(f_p) + \sum_{\{p,q\} \in N} V_{pq}(f_p, f_q) \qquad (4)$$

The segmented text elements are reused in an iterative way to refine the results. Each horizontal line ℓ (as shown in Figure 2(b)) that overlaps with a text element is added to the text lines after refinement. Let us define $\widehat{\ell}$, as the intersection between ℓ and elements classified as text, and let us define $\min_x(\widehat{\ell})$, and $\max_x(\widehat{\ell})$ as the column of the leftmost and rightmost foreground pixel in $\widehat{\ell}$. The refined version of ℓ is defined in Eq. (5), where $\min_y(\widehat{\ell})$, and $\max_y(\widehat{\ell})$ are defined for rows. See Figure 2(c) for the intitial set of estimated text lines and Figure 2(e) for the estimated text lines after one iteration.

$$\begin{aligned} \widetilde{\ell} = \{(x, y) \in \ell | \min_x(\widehat{\ell}) \leq x \leq \max_x(\widehat{\ell}) \text{ and} \\ \min_y(\widehat{\ell}) \leq y \leq \max_y(\widehat{\ell})\} \end{aligned} \qquad (5)$$

After refining the estimated text lines, text elements are extracted again using the described above algorithm, and the process repeats iteratively, until no more new text lines are added.

3.2 Drawing Extraction

As mentioned in the beginning of this section, we seek a group of adjacent pixels that greatly differ in color from the background and occupy a large portion of the document. To ease the computations we use superpixels [8].

We use the color distribution of the largest superpixel (in terms of its pixels number) as an approximation to the color distribution of the background. For each superpixel we define its distance from the background in the CIE-Lab color space by using the Earth Mover's Distance metric (EMD) [24]. The EMD is an image similarity metric that was popularized by Rubner *et al.* [24] for content based image retrieval. For each image (superpixel in our case) a signature is extracted and the distance between the two signatures is posed as a linear programming (LP) problem.

Superpixels that belong to page margins and text have a high probability of not being part of a drawing. So as a preprocessing step, we remove text and page margin superpixels from the image in the following way: a morphological close operation is applied on the text lines discovered in the previous step, and each superpixel that is fully contained within the generated mask is marked as text superpixel. (A generated mask is illustrated in Figure 3(a)).

We use the previously obtained orientation map (as shown in Figure 2(a)) to extract horizontal and vertical lines from page margins. One can notice that horizontal margin lines are horizontal CCs which are located within M_t marginal image rows, and for which $\frac{width(CC)}{height(CC)} > E_t$. Vertical margin lines are defined symmetrically. The constants E_t and M_t are thresholds on the elongation of margin lines and page margins respectively. Superpixels that overlap with margin lines are classified as margin superpixels (as shown in Figure 3(b)).

The data cost $D_p(f_p)$ measures how far in color the superpixel is from the background (bg). The data cost of assigning the drawing label for a text or margin superpixel p is $D_p(f_p = \text{drawing}) = const$, while for any other superpixel p it is defined as:

$$D_p(f_p = \text{drawing}) = \beta e^{-\gamma \cdot EMD(p,bg)}.$$

The data cost of non-drawing label is:

$$D_p(f_p = \text{non-drawing}) \quad = \quad \begin{cases} 0, & p = \text{text/margin}, \\ const, & p = \text{otherwise}, \end{cases}$$

where the constants were tuned for drawing extraction, and are defined in Section 4. See Figure 3(c) for the value of $EMD(p, bg)$, and Figure 3(d) for the value of $D_p(f_p = \text{drawing})$.

The smoothness term $V_{pq}(f_p, f_q)$ measures the color correlation of neighboring superpixels. Superpixels with a similar color distribution and similar size have a higher probability to belong to the same label than the non similar ones. This is defined in our energy minimization scheme as the smoothness term. Between two neighboring superpixels p and q, the smoothness energy term is defined in Eq. (6), where $|p|$ is the number of pixels in p, and δ is defined in a similar way to the constant α. Now that the energy model is fully defined, the segmentation can be estimated as a global minimum, in a similar way to Eq. (4).

$$V_{pq} = \frac{2\min(|p|,|q|)}{(|p|+|q|)} \cdot \exp(-\delta \cdot EMD(p,q)) \qquad (6)$$

The resulting segmentation is a group of connected components (CCs) which consists of drawings and noise (as shown in Figure 3(e)). A post-processing is applied to remove noise. First, margin lines are removed, then we compute for each CC its *Area* feature. We take as seeds (CC_0) all the CCs which are larger in area than $mean(Area) + 0.5std(Area)$, where $mean(Area)$, and $std(Area)$, are the mean and standard deviation of the *Area* feature, respectively (as shown in Figure 3(f)). At the next step we consider as part of a drawing each $c \in CCs$, which is close spatially and in color to the seeds. More formally, we add each $c \in CCs$ which satisfies:

$$\begin{aligned} \exists c_0 \in CC_0, \text{ s.t. } dist(c_0, c) &< D_t, \\ EMD(c_0, c) &< C_t, \end{aligned}$$

where D_t and C_t are thresholds on the proximity in the spatial and color space (defined in Section 4). This produces the final result as shown in Figure 4(b).

4. EXPERIMENTS

We evaluate our algorithm on a historical manuscript from the Islamic Heritage project (IHP) collection [13]. The manuscript dates back to the 17th century and consists of 252 pages that contain text, drawings and a noisy background. PixLabeler [25], a tool for labeling document images, has been used to generate a pixel-level groundtruth, for text and drawings, and hence evaluating the performance of the suggested technique. Each groundtruth image contains two class labels, i.e., drawings and text. The segmentation accuracy of our algorithm has been determined by examining the overlapping percentage between results of the algorithm and the groundtruth. This induces the precision and recall measures which are combined in a single representative scalar, i.e., the F-measure[1].

	Bukhari *et al.* [7]		Ours	
	with noise	w/o noise	with noise	w/o noise
Drawings	85.6%	88%	88.9%	91.7%
Text	89.8%	94.9%	93.8%	94.2%
Average	87.7%	91.4%	91.4%	**92.95%**

Table 1: Segmentation accuracy of our method with respect to Bukhari *et al.*method [7] on both text and drawings. The results are averaged on the IHP dataset using the F-measure.

In all our experiments we have used the thresholds: $D_p(f_p =$ non-text$) = 25$, $D_p(f_p =$ non-drawing$) = 70$, $\beta = 4000$, $\gamma = 0.7$, $M_t = 7\%$ of page margins, $E_t = 4$, $C_t = 16$, and $D_t = 70$ pixels. For text extraction, we experimented all scales between 8 and 20 pixels.

Bukhari *et al.* [7] suggested a method to segment text and non-text entities in binarized document images. We compare the performance of our algorithm with this work on the same dataset. Due to binarization, the documents contain noise artifacts. In contrast to our work, this noise was not considered as a different class in [7]. It is important to note that detecting noise as a separate class in historical documents leverages the segmentation accuracy of other classes, i.e, drawings and text. Table 1 shows a compre-

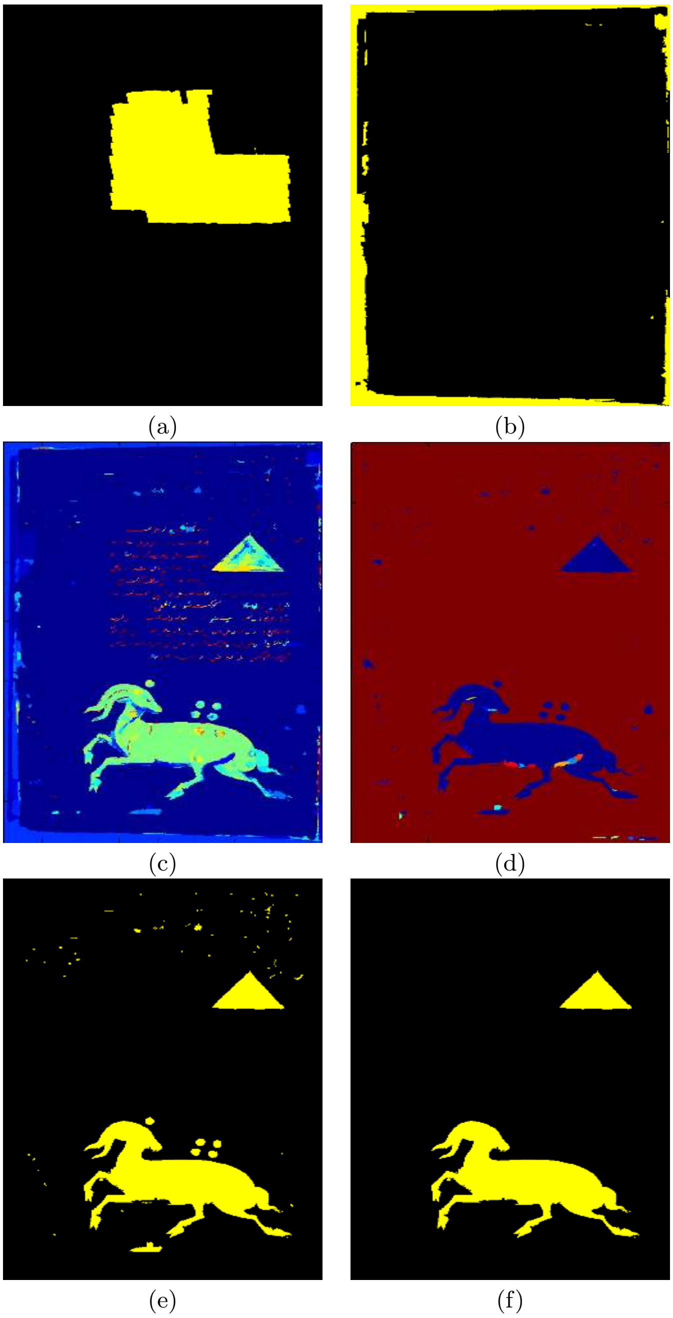
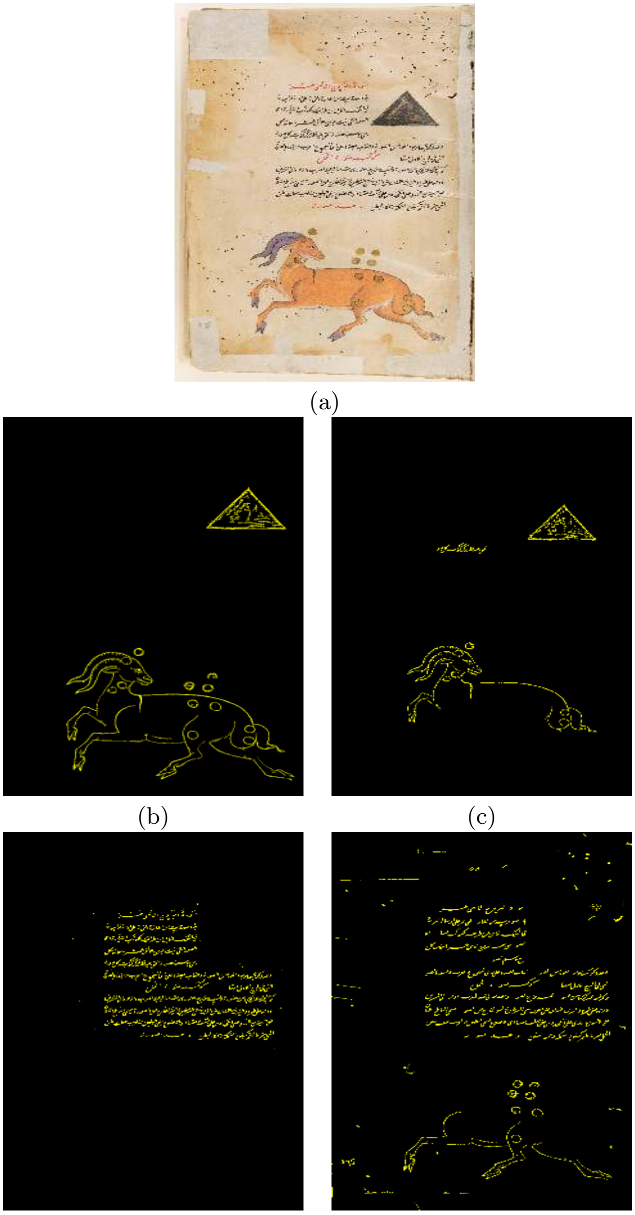


Figure 3: (a) Result of the close operation on the text lines (b) marginal superpixels (c) a heat map for $EMD(p, bg)$, blue color represent low values, whereas red color represents high values; (d) the heat map for $D_p(f_p =$ drawing$)$ (e) a mixture of drawings and noise after applying the CRF for drawings (f) seed for drawing extraction.

[1] The dataset, the results, and the groundtruth are available upon request from the corresponding authors.

hensive performance comparison between the two methods with and without taking noise into consideration during the evaluation process. Figure 4 depicts the segmentation result for the two methods on a random page from the considered manuscript.



(a)

(b)

(c)

(d)

(e)

Figure 4: (a) Original image. Results of our algorithm for drawing segmentation (b) and text segmentation (d). Corresponding results for Bukhari *et al.*method [7] are in (c) and (e) respectively.

Let us concentrate on the case where noise is not part of the evaluation scheme. One can notice that the suggested method achieves better segmentation of drawings with respect to [7]. This fact becomes clearer once we observe that on average 17% of the pixels classified as drawing pixels by Bukhari *et al.* [7] are actually non-drawing pixels (false-positive). On the other hand, the suggested method misses 9% of drawing pixels in terms of false-positive. This observation explains the differences in the precision measure in Table 2. Both methods classify about 5% of drawing pixels as non-drawing pixels (false-negative). Table 2 reports almost equal recall percentage. Exploiting features from both binary and CIE-Lab color versions of the image induces the mentioned results and the advantage of our method on drawing segmentation with respect to Bukhari *et al.* [7].

	Precision	Recall
Bukhari *et al.* [7]	82.5%	94.4%
Ours	**89.6%**	**94.9%**

Table 2: Accuracy of drawing segmentation in terms of precision and recall. Note that these results refer to the case where noise is ignored in the evaluation scheme.

Considering text segmentation, both methods have very low percentage of drawing pixels mistakenly classified as non-drawing pixels (false-positive). One can notice that this positively affects the precision measure in Table 3. Our method classifies 10% of text pixels as non-text pixels (false-negative), while 7% of text pixels are classified as non-text pixels by Bukhari *et al.* [7] approach. Table 3 summarizes the precision and recall measures in the context of text segmentation. It is important to emphasize that the suggested method provides better segmentation accuracy of text with respect to [7] once noise is taken into account.

	Precision	Recall
Bukhari *et al.* [7]	97.5%	**92.9%**
Ours	**99.8%**	89.6%

Table 3: Accuracy of text segmentation in terms of precision and recall. Note that these results refer to the case where noise is ignored in the evaluation scheme.

Figure 5 depicts some sample images and their corresponding results generated by the considered methods. One can notice the importance of exploiting color in the second sample, where Bukhari et al. [7] method misses all the red worms. However, our method perfectly extracts all the worms. Classifying noise elements as text can minimize the accuracy of any text recognition system. The results in Figure 5 reflect the robustness of our method, especially when separating text from noise is considered.

5. CONCLUSION

We present a two stage bottom-up segmentation approach. The suggested method separates text from drawings in historical documents. In the first stage we utilize a binarized version of the document to detect and extract text. In the second stage we remove the text from the document and use a classifier to separate drawings from other classes, e.g., background and noise. A compact representation of the image, i.e., superpixels, is used to optimize the performance of the method. An optimization framework is used to solve the energy equation which defines cost and smoothness terms. Experiments show that the suggested approach achieves better segmentation quality with respect to other existing methods.

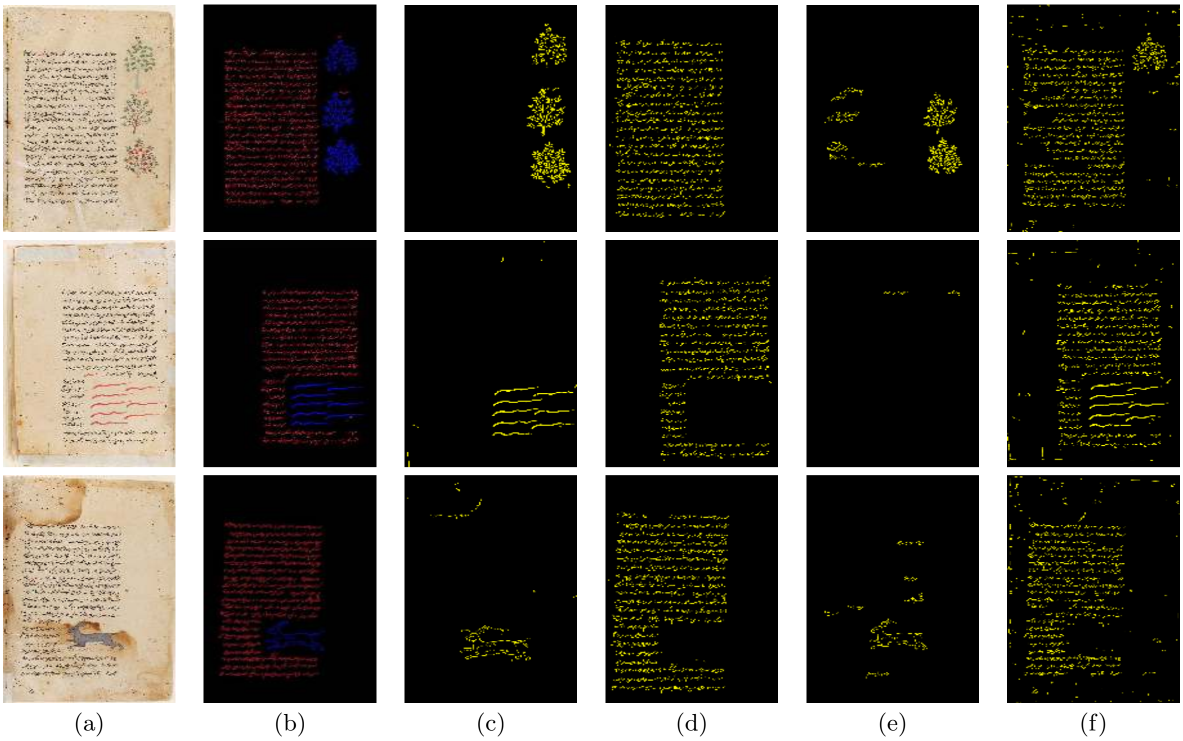


Figure 5: (a) Original image and its (b) Ground truth. Results of our algorithm for (c) drawing segmentation, and (d) text segmentation. Results of Bukhari *et al.* [7] method appear in (e) for drawing segmentation, and (f) for text segmentation.

6. FUTURE WORK

We plan to focus on improving the noise classification step. Mainly because noise stains may have a salient color which is different from the color of the background. In this case our method might classify the stains as drawings. The influence of the binariziation step on the generation of seeds will be examined as well. Moreover, the scope of future work includes improving the method so that it can process more types of document images such as newspapers. Newspapers may contain big titles and various types of drawings which requires additional efforts.

7. ACKNOWLEDGMENTS

This research was supported in part by the DFG-Trilateral grant no. FI 1494/3-2, the Ministry of Science and Technology of Israel, the Council of Higher Education of Israel, the Lynn and William Frankel Center for Computer Sciences and by the Paul Ivanier Center for Robotics and Production Management at Ben-Gurion University, Israel.

8. REFERENCES

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC Superpixels Compared to State-of-the-art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274 – 2282, 2012.

[2] A. Asi, I. Rabaev, K. Kedem, and J. El-Sana. User-assisted alignment of arabic historical manuscripts. In *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*, HIP '11, pages 22–28, 2011.

[3] D. S. Bloomberg. Multiresolution morphological approach to document image analysis. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 963–971, 1991.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[5] S. Bukhari, M. A. Azawi, F. Shafait, and T. Breuel. Document image segmentation using discriminative learning over connected components. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 183–190. ACM, 2010.

[6] S. S. Bukhari, A. Asi, T. M. Breuel, and J. El-Sana. Layout analysis for arabic historical document images using machine learning. In *International Conference on Frontiers in Handwriting Recognition*, pages 639–644, 2012.

[7] S. S. Bukhari, F. Shafait, and T. M. Breuel. Improved document image segmentation algorithm using multiresolution morphology. In *Proceedings of SPIE*. International Society for Optics and Photonics, 2011.

[8] D. Comaniciu, P. Meer, and S. Member. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.

[9] I. Dinstein and Y. Shapira. Ancient hebraic handwriting identification with run-length histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 12(3):405–409, 1982.

[10] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2963–2970, 2010.

[11] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59(2):167–181, Sept. 2004.

[12] J.-M. Geusebroek, A. W. Smeulders, and J. Van De Weijer. Fast anisotropic gauss filtering. *IEEE Transactions on Image Processing*, 12(8):938–943, 2003.

[13] I. H. P. Harvard University. Manuscript of the Ajaib Al-makhluqat (wonders of creation) of Qazwini, 17-Century. http://ocp.hul.harvard.edu/ihp/.

[14] X. He, R. S. Zemel, and D. Ray. Learning and incorporating top-down cues in image segmentation. In *In European Conference on Computer Vision*, pages 338–351. Springer, 2006.

[15] D. Keysers, F. Shafait, and T. M. Breuel. Document image zone classification - a simple high-performance approach. In *2nd International Conference on Computer Vision Theory and Applications*, pages 44–51, 2007.

[16] S. Kim, S. Nowozin, P. Kohli, and C. D. D. Yoo. Higher-order correlation clustering for image segmentation. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1530–1538. 2011.

[17] M. Kubovy and M. van den Berg. The whole is equal to the sum of its parts: A probabilistic model of grouping by proximity and similarity in regular patterns. *Psychological review*, 115(1):131–154, 2008.

[18] M. Lettner and R. Sablatnig. Spatial and spectral based segmentation of text in multispectral images of ancient documents. In *10th International Conference on Document Analysis and Recognition*, pages 813–817, 2009.

[19] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3):303–308, Aug. 2004.

[20] L. Likforman-Sulem, A. Zahour, and B. Taconet. Text line segmentation of historical documents: a survey. *International Journal on Document Analysis and Recognition*, 9:123–138, 2007.

[21] R. Manmatha, C. Han, and E. M. Riseman. Word spotting: New approach to indexing handwriting. In *Proceeding of Computer Vision and Pattern Recognition*, pages 631–637, 1996.

[22] N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[23] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 309–314. ACM, 2004.

[24] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[25] E. Saund, J. Lin, and P. Sarkar. Pixlabeler: User interface for pixel-level labeling of elements in document images.

[26] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 1997.

[27] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

[28] Y. Wang, I. T. Phillips, and R. M. Haralick. Document zone content classification and its performance evaluation. *Pattern Recogn.*, 39(1):57–73, Jan. 2006.

[29] M. Wertheimer. *Laws of organization in perceptual forms*, pages 71–88. Routledge and Kegan Paul, 1938.