

VML-MOC: Segmenting a multiply oriented and curved handwritten text line dataset

Berat Kurar Barakat, Rafi Cohen, Jihad El-Sana
Department of Computer Science
Ben-Gurion University
berat, rafico, el-sana@post.bgu.ac.il

Irina Rabaev
Software Engineering Department
Shamoon College of Engineering
irinar@ac.sce.ac.il

Abstract—This paper publishes a natural and very complicated dataset of handwritten documents with multiply oriented and curved text lines, namely VML-MOC dataset. These text lines were written as remarks on the page margins by different writers over the years. They appear at different locations within the orientations that range between 0° and 180° or as curvilinear forms. We evaluate a multi-oriented Gaussian based method to segment these handwritten text lines that are skewed or curved in any orientation. It achieves a mean pixel Intersection over Union score of 80.96% on the test documents. The results are compared with the results of a single-oriented Gaussian based text line segmentation method.

I. INTRODUCTION

Handwritten document image recognition has several processing phases, including text line segmentation. Output of text line segmentation phase is commonly used for word and character recognition in turn. Therefore, a robust text line segmentation is crucial for successful handwriting recognition. Driven by this importance text line segmentation is extensively studied in the recent years. Most existing methods [1]–[5] are designed with the assumption of horizontal or near-horizontal text lines. Consequently, there is still a large gap when segmentation applied to text lines with arbitrary orientations.

The challenge with the segmentation of skewed and curved text lines comes with the lack of a natural benchmark dataset for testing and comparing algorithms in realistic scenarios. There are works on slightly skewed handwritten text lines [6], [7] and on curved printed text lines [8]–[11]. However, their dataset is either synthetic or slightly skewed or not available publicly.

We present a natural handwritten benchmark dataset, VML-MOC, for heavily skewed and curved text lines (Figure 1). These text lines are side notes added by scholars over the years on the page margins, each time with a different orientation and sometimes in an extremely curvy form due to space constraints. The dataset consists of 30 document images which are taken from multiple manuscripts and contains various kinds of skewed and curved text lines. We evaluate a multi-oriented Gaussian based method and a single-oriented Gaussian based method. Multi-oriented Gaussian based method was previously proposed as a part of a whole framework for simplifying reading of historical manuscripts [12]. This paper investigates details of this method in further, reports its results on the

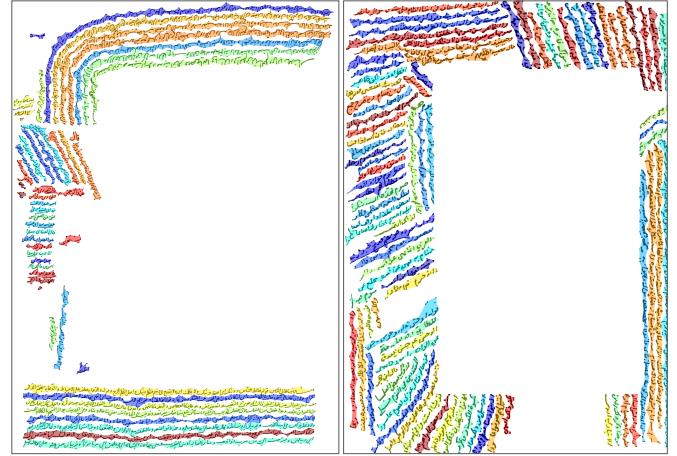


Fig. 1. Sample images from the VML-MOC dataset with colored visualization of their bounding polygon labels.

proposed benchmark dataset and compares with the results of a single-oriented Gaussian based method [1].

II. RELATED WORK

Text line segmentation is a prior step for various algorithms, such as indexing, word spotting and OCR. The vast majority of procedures for text line extraction are designed to process horizontal or straight lines. Hence, they are unsuitable for scenarios where text exhibit multi-skewed, multi-directed and highly curled lines. Few methods address text line segmentation of warped and multi-skewed lines, which we can divide into two broad categories: whether a method requires a learning process or not.

Learning-free approaches are mostly based on projection profiles, Gabor Transform, active contours, and features specific to application.

Basu *et al.* [6] assumed hypothetical flows of water, from both left and right sides of the image boundary, which face obstruction from characters of the text line. The stripes of areas left dry at the end of the process represent text lines. Roy *et al.* [11] utilized water reservoir-based background information to identify text lines in printed documents. Reservoirs are based on connected component cavities, and are used to estimate skews of line-parts. Bukhari *et al.* [8], [13]–[15] presented active contour models for segmenting warped textlines from camera-captured printed documents.

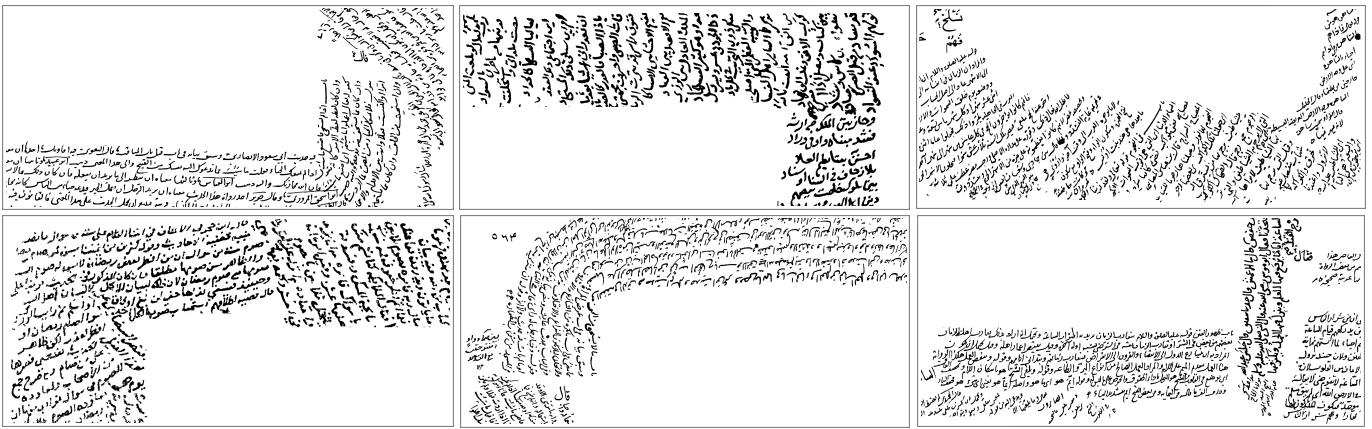


Fig. 2. Sample patches from document images of the VML-MOC dataset. There are text lines with a skew range of $[0^\circ, 180^\circ]$ and with various arc shapes.

The snakes are initialized over each connected component and, following the deformation, joined together to result in textline. In [10], [16] active contour model was adapted to handwritten documents. Ouwayed and Belaïd [17] adopted active contour approach to estimate mesh side over the document image, where each mesh is designed to contain parts of few lines. Then, skew in each mesh is detected using projection profiles. Morillot *et al.* [18] used a sliding window to estimate the lower baseline position for each image column, followed by a vertical shift correction. Boubaker *et al.* [19] approximated baselines with piece-wise linear curves, based on a language specific features. Herzog *et al.* [20] and Asi *et al.* [21] adopted Gabor Transform for identifying multi-oriented text blocks in handwritten documents, where lines of each text block share homogeneous orientations.

Recently methods inspired by machine learning have proven to be efficient for textline extraction. Zhang *et al.* [22] presented a framework for multi-oriented text detection in natural images, where they integrated semantic labeling by Fully Convolutional Networks. The method assumes that characters from each text line are in the arrangement of straight or near-straight line. Bukhari *et al.* [23] used machine learning to identify text areas of different orientations in Arabic manuscripts. A number of features are extracted from connected components, and fed into a multi-layer perceptron classifier.

Although textline segmentation has been extensively studied during the last decades, it remains a challenging problem for documents with complex layout such as those present in Figure 1.

III. VML-MOC DATASET

VML-MOC (Visual Media Lab - Multiply Oriented and Curved) dataset is a collection of 30 pages selected from several manuscripts. Some of these manuscripts are from a private library in the old city of Jerusalem, and others are from the Islamic manuscript digitization project of Leipzig University Library. We consider that prospective algorithms can be learning based and provide an official dataset split to evaluate different models under the same conditions. Ran-

domly, 20 pages were selected for train set and 10 pages for test set. The images with their corresponding ground truth files are publicly available¹.

VML-MOC dataset document images purely contain side notes, which are binarized using the algorithm from [24]. Hence, the researchers can focus only on text line extraction of multiply oriented and curved text lines, devoid of dealing with the challenges of page segmentation, heterogeneity of side text and main text areas and binarization defects. Variance is mostly at the orientation and curvature of the text lines. The dataset contains text lines with a skew range of $[0^\circ, 180^\circ]$ and with all possible arc shapes. Figure 2 shows sample patches from document images of the VML-MOC dataset.

For annotation we used Aletheia [25], a semi-automated ground truthing system. The ground truth is provided in three forms: raw pixel labeling, DIVA pixel labeling and PAGE [26] xml file.

A. Raw pixel labeling

Raw pixel labeling classifies each pixel as a part of a text line or background. It is a matrix of non-negative integers of the same size as the document image. Background pixels are labeled as 0. Pixels of the first text line are labeled as 1, pixels of the second text line are labeled as 2 and so on. Figure 3 shows colored visualization of raw pixel labeling.



Fig. 3. Colored visualization of raw pixel labeling. All the pixels of a text line are assigned the same non-negative integer.

¹https://www.cs.bgu.ac.il/~berat/data/moc_dataset.zip

B. PAGE xml file

PAGE xml file contains a bounding polygon for every text line in a document image. Bounding polygons were extracted using the raw pixel labeling as follows: Pixels of a text line were considered as a set of points, and a concave hull that envelops these points was computed and regarded as the bounding polygon of this text line. Figure 1 shows colored visualization of bounding polygons in PAGE xml file.

C. DIVA pixel labeling

DIVA pixel labeling is provided to be used with the ICDAR2017 competition line segmentation evaluator [27]. It is a matrix of non-negative integers of the same size as the document image, and assigns a label for every pixel in the document image. It distinguishes text line pixels, background pixels and boundary pixels. Boundary pixels are the pixel inside a bounding polygon of a text line that do not belong to the foreground (Figure 4).

To prepare DIVA pixel labeling, we first overlaid the binarized document image with the polygons given by the PAGE xml file. The foreground pixels within the polygons are encoded as text line using the color code (0, 0, 1) in RGB. The background pixels within polygons are encoded as boundary using the color code (128, 0, 0) in RGB. All the pixels out of polygons are encoded as background using the color code (0, 0, 0) in RGB. Figure 4 illustrates these encodings and corresponding color codes.

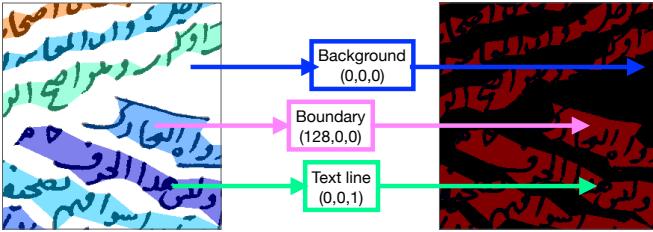


Fig. 4. DIVA pixel labeling encodes the pixels out of polygons into (0, 0, 0), the background pixels within the polygons into (0, 0, 1), and the foreground pixels within the polygons into (128, 0, 0).

IV. METHOD

The method starts with text line enhancement by convolving the image with second derivative of multi-oriented and multi-scaled Gaussians. The enhanced image is then binarized by Niblack algorithm. Binarization output contains blob lines hovering over the places of text lines. It may also contain some false ligature blobs caused by the multiple orientations of Gaussian. Therefore, the blob lines are classified as valid or invalid, based on how well a blob line can be approximated by piecewise linear approximation. The invalid blob lines are then morphologically skeletonized and decomposed at bifurcation points of the skeleton. After the decomposition, energy minimization removes false ligature blobs. Removal of false ligatures leaves some broken blob lines. These broken blob lines are merged using Minimum Spanning Tree (MST).

In the final stage, the connected components of text lines are assigned to blob lines using energy minimization.

In the rest of this section we further study each of the above steps.

A. Text line enhancement and binarization

The pixels in an image can be regarded as two dimensional random variables generated by an unknown probability distribution function. Usually pixels of text lines have smaller intensity values than those in the rest of the image. Therefore, convolution of a text line with second derivative of an anisotropic Gaussian elongated along the text line direction generates ridges over the text line areas. Here arise two issues with VML-MOC dataset: 1) Text line height varies due to ascenders and descenders, 2) Text line direction varies due to multiple orientations or curvatures.

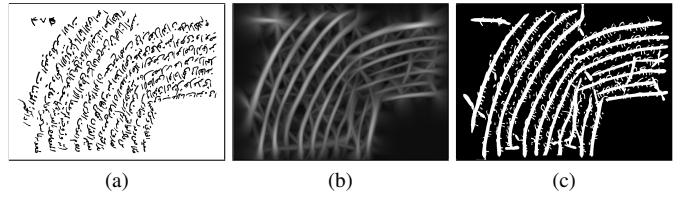


Fig. 5. (a) An input patch from VML-MOC. (b) Enhanced text lines resulted from applying the multi-oriented and multi-scaled Gaussian filter bank. (c) Final blob lines after Niblack binarization of enhanced text lines.

To deal with these issues, we generated a filter bank using second derivative of anisotropic Gaussian. This bank contains all possible combinations of orientations within the range of $[0^\circ, 180^\circ]$ and scales within the range of $[\mu/2, (\mu + \sigma)/2]$, where μ and σ are the average and standard deviation of the heights of connected components in the image. We applied this filter bank and considered the optimal scale and the optimal orientation for each pixel. Therefore each pixel returns the maximum possible response (Figure 5b) for it, using this filter bank. Enhanced text lines were then binarized by Niblack algorithm, to get the final blob lines that hovers over the text lines (Figure 5c).

B. False ligature removal

Blob lines resulted from the binarization phase might contain false ligature blobs (Figure 5c). To remove these ligatures we first classified blob lines as valid or invalid. For each blob line, principal component analysis was applied to its pixels. Then the blob line was horizontally aligned via the rotation transformation matrix in Equation (1).

$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (1)$$

where θ is the reference angle of the first principal component.

We fitted least square linear splines using 20 knots, to the horizontally aligned set of points (Figure 6a). For each spline on a blob line, fitting score is the 1-norm between the linear fit and the blob line points in that spline. Finally, we considered a blob line as valid, if its maximum fitting score is less than

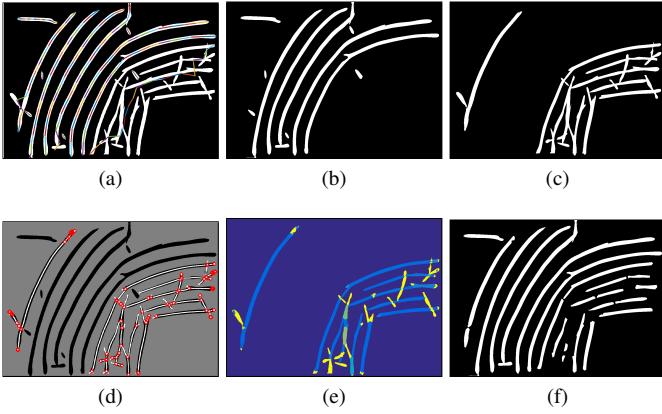


Fig. 6. (a) Least square linear splines fitted to the blob lines. Each spline in a blob line is represented by a color. Splines were fitted to horizontally aligned points, this figure shows re-rotated data for visualization purposes. (b) Valid blob lines. (c) Invalid blob lines. Notice the false ligatures on invalid blob lines. (d) Invalid blob lines are decomposed at their bifurcation points. (e) Each of the decomposed invalid blob lines is assigned a label cost. (f) False ligature blobs with high label cost are removed using energy minimization.

the 80% percent of maximum filter scale (Figure 6b). Invalid blob lines (Figure 6c) were skeletonized and decomposed at their bifurcation points (Figure 6d).

False ligatures in the set of decomposed blob lines were removed by energy minimization. To do this, every blob line in the decomposed blob lines set was assigned a label cost (Figure 6e) that describes how much its orientation deviates from the dominant orientation in a small radius around it. This radius is equal to 18 times the ratio of total areas of blob lines to the total perimeters of blob lines in the image. Dominant orientation (θ_{hist}) within this radius is the peak value in histogram of orientation angles of the filter bank that gave the highest response with the pixels within this radius. Blob line orientation (θ_{pca}) is the slope of first principal component of this blob line's pixels. For each blob line the label cost is computed by using Equation (2)

$$h_\ell = \exp(\gamma \cdot (1 - |\theta_{hist} - \cos(\theta_{PCA})|)) \quad (2)$$

where γ is a constant which was set to 50. Finally, these label costs are fed into energy minimization function to remove false ligature blobs (Figure 6f).

C. Merging broken blob lines

We merged the broken blob lines using a minimum spanning tree (MST) on an undirected weighted graph, $G = (V, E)$. The set of vertices, V , is composed of all the end-points of the blob lines, in addition to a root vertex that is connected to all the end-points. The set of edges is $E = E_1 \cup E_2 \cup E_3$:

- 1) E_1 is the set of edges that connects two vertices of a blob line, their weight is set to 0.
- 2) E_2 is the set of edges between the root and all the vertices, their weight is the normalized number of foreground pixels overlapping with the blob line.
- 3) E_3 consists of edges between end-points that belong to different lines. Their weight is a local linearity measure defined in the following. In set E_3 , for every end-point

we considered only the edges with the most successful linearity measure.

Local linearity measure: Local linearity measure shows how linear the connection is between two end-points, u and v , of two blob lines. For each end-point, u and v , it chooses a nearby point on its blob line. These near-by points are s and t , respectively (Figure 7a). Then local linearity measure is computed as in Equation (3):

$$w(u, v) = \exp\left(\gamma \left(\frac{\|s - u\| + \|u - v\| + \|v - t\|}{\|s - t\|} - 1\right)\right) \quad (3)$$

where γ is a constant and $\|u - v\|$ is the Euclidean distance between u and v .

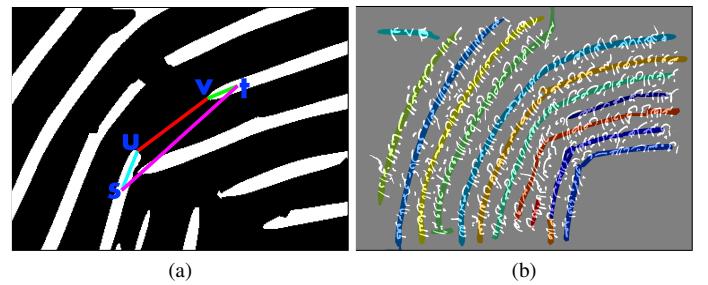


Fig. 7. (a) Local linearity measure. (b) Merged blob lines resulted from applying MST.

D. Labeling connected components

This phase uses energy minimization [28] for assigning connected components to text lines. Let \mathcal{L} be the set of blob lines, and \mathcal{C} be the set of connected components in the image. Energy minimization finds a labeling f that assigns each component $c \in \mathcal{C}$ a label $l_c \in \mathcal{L}$, where $E(f)$ has the minimum.

$$E(f) = \sum_{c \in \mathcal{C}} D(c, l_c) + \sum_{\{c, c'\} \in \mathcal{N}} d(c, c') \cdot \delta(l_c \neq l_{c'}) + \sum_{\ell \in \mathcal{L}} h_\ell \quad (4)$$

Energy function has three terms: data cost, smoothness cost, and label cost.

- 1) Data cost: For every $c \in \mathcal{C}$, $D(c, l_c)$ is defined as the Euclidean distance between the centroid of c and the blob line l_c .
- 2) Smoothness cost: Let N be the nearest component pairs. For every $\{c, c'\} \in N$, $d(c, c') = \exp(-\alpha \cdot d_e(c, c'))$ where $d_e(c, c')$ is the Euclidean distance between the centroids of the components, c and c' . $\delta(l_c \neq l_{c'})$ is 1 if the condition inside the parentheses holds and 0 otherwise.
- 3) Label cost: For every blob line $\ell \in \mathcal{L}$, h_ℓ is defined as $\exp(\beta \cdot r_\ell)$ where r_ℓ is the normalized number of foreground pixels overlapping with blob line ℓ .

V. EVALUATION

We used ICDAR2017 line segmentation evaluator tool [27] for the evaluation. This tool is freely available as open source².

²https://github.com/DIVA-DIA/DIVA_Line_Segmentation_Evaluator



Fig. 8. The resultant pixel labels for text line segmentation of the sample patch in Figure 5a

An open source tool has a reviewed source code that minimizes the risk of erroneous implementations. Besides, it enables fair comparison of methods in the same way as publicly published datasets.

Evaluation of text line segmentation is based on Intersection over Union (IU). First, an IU score is computed for each possible pair of Ground Truth (GT) polygons and Prediction (P) polygons according to the Equation 5:

$$IU = \frac{IP}{UP} \quad (5)$$

where IP denotes number of intersecting foreground pixels among the pair of polygons and UP denotes number of foreground pixels in the union of foreground pixels of the pair of polygons. Then, the pairs with maximum IU score are selected as the matching pairs of GT polygons and P polygons. Pixel IU and Line IU are calculated among these matching pairs.

A. Pixel IU

Pixel IU is measured at pixel level. First, for each matching pair, line TP, line FP and line FN is computed.

- Line TP is the number of foreground pixels that are correctly detected.
- Line FP is the number of background pixels that are falsely detected as foreground.
- Line FN is the number of foreground pixels that are not detected by the method.

Then pixel IU is calculated according to the following Equation 6:

$$\text{Pixel } IU = \frac{TP}{TP + FP + FN} \quad (6)$$

where TP is the global sum of line TPs, FP is the global sum of line FPs, and FN is the global sum of line FNs.

B. Line IU

Line IU is measured at line level. First, for each matching pair, line precision and line recall is computed according to

the Equations 7 and 8:

$$\text{Line precision} = \frac{\text{line } TP}{\text{line } TP + \text{line } FP} \quad (7)$$

$$\text{Line recall} = \frac{\text{line } TP}{\text{line } TP + \text{line } FN} \quad (8)$$

Then, line IU is calculated according to the Equation 9:

$$\text{Line } IU = \frac{\text{CL}}{\text{CL} + \text{ML} + \text{EL}} \quad (9)$$

where CL is the number of correct lines, ML is the number of missed lines, and EL is the number of extra lines. For each matching pair and threshold value of 0.75:

- A line is correct if both, the line precision and the line recall are above the threshold value.
- A line is missed if the line recall is below the threshold value.
- A line is extra if the line precision is below the threshold value.

C. Mean pixel IU and mean line IU

For each page, first the pixel IU and the line IU are computed. Then, mean pixel IU is obtained by averaging the pixel IU of all pages of the test set and mean line IU is obtained by averaging the line IU of all pages of the test set.

VI. EXPERIMENTS

We run experiments on VML-MOC test set using our method based on multi-oriented Gaussian, and a single-oriented Gaussian based method [1]. We report the mean pixel IU and the mean line IU over the entire test set. The scores achieved by the two methods are presented in Table I, and some qualitative results are shown in Figure 9. The results

TABLE I
PERFORMANCE PERCENTAGE OF TWO METHODS ON THE VML-MOC TEST SET.

	Mean pixel IU	Mean line IU
Single-oriented Gaussian [1]	37.88	12.89
Multi-oriented Gaussian	80.96	60.99

reveal that multi-oriented Gaussian based method can extract multi-oriented and curved text lines in opposite to the single-oriented Gaussian based method. However, these are baseline results and there is still room for improvement.

VII. CONCLUSION AND FUTURE WORK

This paper presents a multiply oriented and curved handwritten text line dataset, namely VML-MOC dataset. To the best of the authors' knowledge, VML-MOC dataset is the first publicly available dataset that introduces the problem of segmenting multiply oriented and curved handwritten text lines. Furthermore, we evaluated and compared two line extraction methods, a single-oriented Gaussian based method and a multi-oriented Gaussian based method, on this dataset. Results have shown that ordinary text line segmentation methods are not

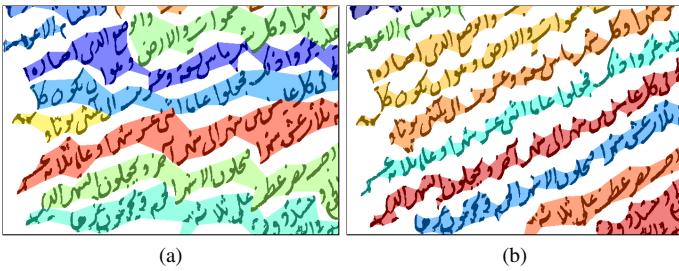


Fig. 9. (a) Result from single-oriented Gaussian based method on an example patch. (b) Result from multi-oriented Gaussian based method on the same patch.

successful on VML-MOC dataset, and text line segmentation methods without horizontal/straight line assumption has to be developed. An important direction for future work would be the evaluation of deep learning based methods.

ACKNOWLEDGMENT

Authors would like to thank Hamza Barakat for his helps in preparing the dataset. This research was supported in part by Frankel Center for Computer Science at Ben-Gurion University of the Negev.

REFERENCES

- [1] R. Cohen, I. Dinstein, J. El-Sana, and K. Kedem, “Using scale-space anisotropic smoothing for text line extraction in historical documents,” in *International Conference Image Analysis and Recognition*. Springer, 2014, pp. 349–358.
- [2] B. Moysset, C. Kermorvant, C. Wolf, and J. Louradour, “Paragraph text segmentation into lines with recurrent neural networks,” in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2015, pp. 456–460.
- [3] B. Moysset, C. Kermorvant, and C. Wolf, “Full-page text recognition: Learning where to start and when to stop,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 871–876.
- [4] G. Renton, C. Chatelain, S. Adam, C. Kermorvant, and T. Paquet, “Handwritten text line segmentation using fully convolutional network,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 5. IEEE, 2017, pp. 5–9.
- [5] G. Renton, Y. Soullard, C. Chatelain, S. Adam, C. Kermorvant, and T. Paquet, “Fully convolutional network with dilated convolutions for handwritten text line segmentation,” *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 21, no. 3, pp. 177–186, 2018.
- [6] S. Basu, C. Chaudhuri, M. Kundu, M. Nasipuri, and D. K. Basu, “Text line extraction from multi-skewed handwritten documents,” *Pattern Recognition*, vol. 40, no. 6, pp. 1825–1839, 2007.
- [7] N. Ouwayed, A. Belaid, and A. Belaïd, “A general approach for multi-oriented text line extraction of handwritten document A general approach for multi-oriented text line extraction of handwritten document,” *International Journal on Document Analysis and Recognition*, vol. 14, no. 4, p. 10, 2011. [Online]. Available: <https://hal.inria.fr/inria-00635363>
- [8] S. Bukhari, F. Shafait, and T. M. Breuel, “Segmentation of curled textlines using active contours,” in *2008 The Eighth IAPR International Workshop on Document Analysis Systems*. IEEE, 2008, pp. 270–277.
- [9] S. Bukhari, F. Shafait, and T. Breuel, “Ridges based curled textline region detection from grayscale camera-captured document images,” in *International Conference on Computer Analysis of Images and Patterns*. Springer, 2009, pp. 173–180.
- [10] S. Bukhari, F. Shafait, and T. M. Breuel, “Text-line extraction using a convolution of isotropic gaussian filter with a set of line filters,” in *2011 International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 579–583.
- [11] P. P. Roy, U. Pal, and J. Lladós, “Text line extraction in graphical documents using background and foreground information,” *International Journal on Document Analysis and Recognition*, vol. 15, no. 3, pp. 227–241, 2012.
- [12] A. Asi, R. Cohen, K. Kedem, and J. El-Sana, “Simplifying the reading of historical manuscripts,” in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2015, pp. 826–830.
- [13] S. Bukhari, T. Breuel, and F. Shafait, “Textline information extraction from grayscale camera-captured document images,” in *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2009, pp. 2013–2016.
- [14] S. Bukhari, F. Shafait, and T. Breuel, “Performance evaluation of curled textline segmentation algorithms on cbdar 2007 dewarping contest dataset,” in *2010 IEEE International Conference on Image Processing*. IEEE, 2010, pp. 2161–2164.
- [15] S. Bukhari, F. Shafait, and T. Breuel, “Coupled snakelets for curled textline segmentation from warped document images,” *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 16, no. 1, pp. 33–53, 2013.
- [16] S. Bukhari, F. Shafait, and T. M. Breuel, “Script-independent handwritten textlines segmentation using active contours,” in *2009 10th International Conference on Document Analysis and Recognition*. IEEE, 2009, pp. 446–450.
- [17] N. Ouwayed and A. Belaïd, “A general approach for multi-oriented text line extraction of handwritten documents,” *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 15, no. 4, pp. 297–314, 2012.
- [18] A. Kölsch, A. Mishra, S. Varshneya, M. Z. Afzal, and M. Liwicki, “Recognizing challenging handwritten annotations with fully convolutional networks,” in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2018, pp. 25–31.
- [19] H. Boubaker, M. Kherallah, and A. M. Alimi, “New algorithm of straight or curved baseline detection for short arabic handwritten writing,” in *2009 10th International Conference on Document Analysis and Recognition*. IEEE, 2009, pp. 778–782.
- [20] R. Herzog, A. Solth, and B. Neumann, “Text block recognition in multi-oriented handwritten documents,” 2014. [Online]. Available: <http://edoc.sub.uni-hamburg.de/informatik/volltexte/2014/207/>
- [21] A. Asi, R. Cohen, K. Kedem, J. El-Sana, and I. Dinstein, “A coarse-to-fine approach for layout analysis of ancient manuscripts,” in *2014 14th International Conference on Frontiers in Handwriting Recognition*. IEEE, 2014, pp. 140–145.
- [22] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, “Multi-oriented text detection with fully convolutional networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4159–4167.
- [23] S. Bukhari, T. Breuel, A. Asi, and J. El-Sana, “Layout analysis for arabic historical document images using machine learning,” in *2012 International Conference on Frontiers in Handwriting Recognition*. IEEE, 2012, pp. 639–644.
- [24] O. Biller, I. Rabaev, K. Kedem, J. J. El-Sana *et al.*, “Evolution maps and applications,” *PeerJ Computer Science*, vol. 2, p. e39, 2016.
- [25] C. Clausner, S. Pletschacher, and A. Antonacopoulos, “Aletheia-an advanced document layout and text ground-truthing system for production environments,” in *2011 International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 48–52.
- [26] S. Pletschacher and A. Antonacopoulos, “The page (page analysis and ground-truth elements) format framework,” in *2010 20th International Conference on Pattern Recognition*. IEEE, 2010, pp. 257–260.
- [27] F. Simistira, M. Bouillon, M. Seuret, M. Würsch, M. Alberti, R. Ingold, and M. Liwicki, “Icdar2017 competition on layout analysis for challenging medieval manuscripts,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 1361–1370.
- [28] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, “Fast approximate energy minimization with label costs,” *International journal of computer vision*, vol. 96, no. 1, pp. 1–27, 2012.