Segmentation of Arabic Handwritten Documents into Text Lines using Watershed Transform

A. Souhar[1]*, Y. Boulid[2], ElB. Ameur[1] and Mly. M. Ouagague[2]

[1]University Ibn Tofail, Faculty of science, 14000, Kenitra (Morocco)
[2]Research and Development Unit Maxware Technology, 14000, Kenitra (Morocco)

Received 1 June 2017 | Accepted 13 July 2017 | Published 21 August 2017

Abstract

A crucial task in character recognition systems is the segmentation of the document into text lines and especially if it is handwritten. When dealing with non-Latin document such as Arabic, the challenge becomes greater since in addition to the variability of writing, the presence of diacritical points and the high number of ascender and descender characters complicates more the process of the segmentation. To remedy with this complexity and even to make this difficulty an advantage since the focus is on the Arabic language which is semi-cursive in nature, a method based on the Watershed Transform technique is proposed. Tested on «Handwritten Arabic Proximity Datasets» [21] a segmentation rate of 93% for a 95% of matching score is achieved.

Keywords

Text Line Segmentation, Arabic Script, Handwritten Character Recognition, Connected Component Analysis, Projection Profile, Watershed Transform.

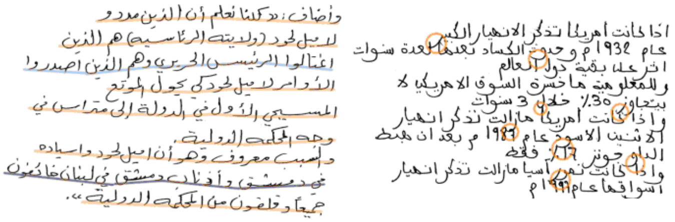
DOI: 10.9781/ijimai.2017.08.002

I. Introduction

In order to recognize texts in a document, often character recognition systems go through four stages: the pre-processing stage, which concerns the preparation of the document in terms of normalization and suppression of noise. The segmentation stage that allows the detection of lines, words and also the segmentation of those words into characters. The third stage concerns the feature extraction from the character, this feature allows to minimize the intra-class variance while maximizing the inter-class variance. The fourth stage involves learning and testing to recognize new letters or new words based on machine learning algorithms. To these stages is added a post-processing stage to verify recognized words using a lexical, syntax and semantic analysis.

All these phases are crucial since if an error is made in one it will strongly influence the subsequent ones [23]. In this paper we focus on the text line segmentation which could be defined according to [1] in the process of assigning the same label to units that are partially aligned. We are interested in Arabic handwritten documents which are more challenging than Latin documents and that is mainly due to the semi-cursive nature of the Arabic script which is characterized by its calligraphy and the presence of ascending and descending character, the overlapping between piece of Arabic words and also the diacritical points located either above or below characters [24].

In addition to these characteristics, the different writing styles and the inclination within the same line make the process of text line extraction from Arabic handwriting challenging (Fig. 1).



وأضاف: «كلنا نعلم أن الذين هددوا
لاميل لحود (ولايته الرئاسية) هم الذين
اغتالوا الرئيس الحريري وهم الذين اصدروا
الأوامر لاميل لحود كي يحول الموقع
المسيحي الأول في الدولة الى متراس في
وجه المحكمة الدولية.
والسبب معروف وهو أن اميل لحود واسياده
في دمشق وأذناب دمشق في لبنان خائفون
جميعاً وخائفون من المحكمة الدولية».

اذا كانت امريكا تتذكر الانهيار الكبير
عام 1932 م وحدوث الكساد بعدها لعدة سنوات
اثرت على بقية دول العالم
والمعلومة ماخسرة السوق الامريكية لا
يتجاوز 30% خلال 3 سنوات
واذا كانت امريكا مازالت تذكر انهيار
الاثنين الاسود عام ١٩٨٧ م بعد ان هبط
الداو جونز ٢٢٪ فقط
واذا كانت نمور آسيا مازالت تذكر انهيار
اسواقها عام ١٩٩٧ م

Fig. 1. Some problems of Arabic handwritten text lines extraction.

To cope with this complexity several techniques have been proposed in the literature. Generally we can categorize them into three approaches (Fig. 2).

The first approach is based on the analysis of the arrangement of the connected components in order to construct successively the lines. A multi-agent system to detect and group connected components that belong to the same line is proposed in [2]. The first agent estimates global parameters and extracts the lines, the second one searches and detects adjacent components and the third one segments the touching lines. A similar technique is called Smearing method where the goal is to apply some transformation on the text bloc in order to group together homogeneous blocks constituting the lines [3, 4, 5, 6].

The second approach tries to search between interlines in order to separate the adjacent lines. The algorithm in [7] computes the distance transform directly from the gray scale images and generates two types of seams: the medial seams determine the text lines and the separating seams define the upper and lower boundaries of the text lines. Similarly, the work in [8] splits the document into vertical slices and applies a matching method on the result of projection profile in order to estimate the medial seam, then a modified version of seam carving procedure is used to compute the separating seam. The final seams will go through the regions between text lines.

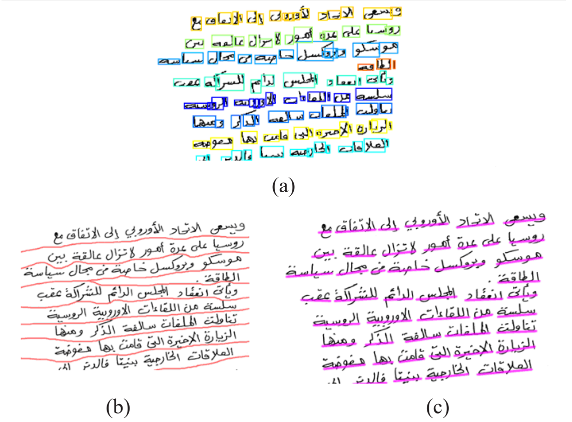
* Corresponding author.

E-mail address: houssouhar@gmail.com

A method was proposed in [9] that uses ALCM algorithm which gives a mask corresponding to text line locations. Then, the text lines are extracted by superimposing the components with text line pattern mask. Another method is based on convolution neural network with watershed transform, which is proposed in [10] to estimate the text area between the baseline and the corpus line.

The third approach uses the baseline of words and tries to connect those that participate to the same lines. Other methods represent the problem as a graph and search the orientations of text components and uses Breadth First Search algorithm and affinity propagation clustering method to assign the components to text lines [11, 12]. The work in [13] adapts the problem of text line extraction from binary Arabic handwritten documents as a Markov Decision Processes using knowledge about the features and arrangement of the components belonging to the same line.

The papers in [1, 14] give a complete study of several text lines segmentation methods of handwritten document.


(a)

(b)

(c)

Fig. 2. The main approaches for handwritten text lines extraction.

To respond to the specificity of the Arabic script, we investigate the use of the Watershed Transform which is a well known image segmentation approach. The idea is that when considering the text components as markers, the catchment basin (regions) of watershed occupy a space that adapts with the shape and size of the component with respect to its neighbors. This effect could be exploited to estimate the locations of text lines.

To respond to the specificity of the Arabic script, we found interesting to investigate the approach based on the analysis of text component using watershed transform as preprocessing step to get the regions of the text component in order to localize the neighboring components within the same line. We found that the work in [15] first estimates the baseline using adaptive head-tail connection in order to link the component with each other and then uses watershed on the result of the baseline detection in order to extract the lines.

We would like also to investigate the use watershed transform as a post processing step in order to localize the lines estimated by a matching horizontal projection profile technique based on the work in [8].

The goal here is to compare if it is interesting to use watershed transform as preprocessing or post processing steps. This paper is an extension of our work in [22], and it is organized as follows: section 3 describes the proposed approaches. Section 4 presents and discusses the results and finally section 5 concludes the paper.

II. Proposed Approach

According to a local vision, a text line can be perceived as a set of aligned words. Thus we focus on the detection of Piece of Arabic words that constitute this line. From this point of view, the adequate approach will be the one that is based on the analysis of the arrangement of the connected components (Fig. 2.a)

According to a global vision, a text line can only be defined by means of the following line. Here we are not interested to the constituent of the line, but to its neighbors. From this point of view the adequate approach will be the one that approximates the baseline (Fig. 2.b and 2.c).

Therefore we investigate the use of watershed transform technique from two visions, local and global.

A. Watershed Transform

In topography watershed means the ridge that divides areas drained by different river systems. Here we refer to image segmentation technique in the field of mathematical morphology. The idea is to treat the image as a topographic surface in which the dark pixels are considered low elevations and lighter ones are considered high elevations and when the flooding starts from the low elevations and the merging of the waters from different locations is prevented, the resulted image will be portioned into catchment basins and watershed lines as illustrated in Fig. 3.

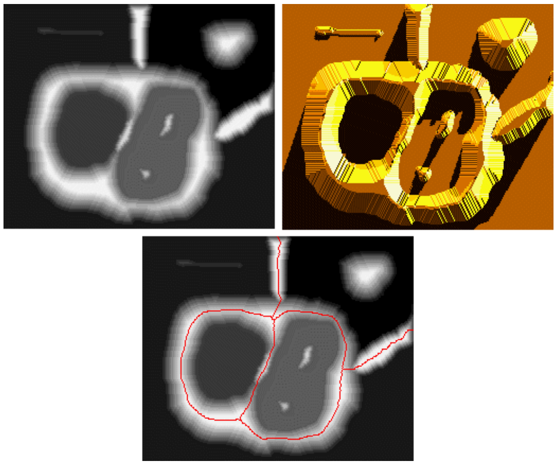


Fig. 3. Illustration of watershed segmentation: (top-left) gray scale image, (top-right) topographic surface, (bottom) watersheds in red.

Fernand Meyer [16] introduced an algorithm based on a set of defined markers in order to overcome the problem of over segmentation which can be summarized as follows:

1. Each of the markers is given a different label.
2. The pixels around each marked area are inserted into a priority queue with priority level that corresponds to the gradient magnitude of the pixel.
3. The pixel having the lowest priority is extracted from the queue and if all its neighbors that have already label have the same label, then it will be labeled with their label. The non-marked neighbors that are not in the priority queue are put in it.
4. Redo step 3 until the queue is empty.

The resulted image will contain non-labeled pixels which correspond to watershed lines.

B. Local Vision

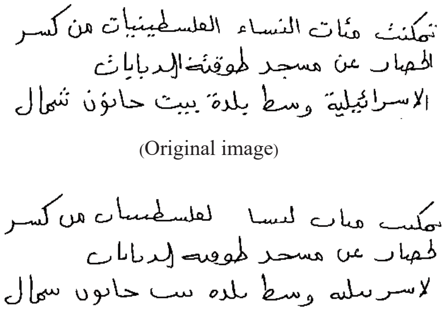
As stated in [22], the idea of using watershed transform comes from the observation of its effect on binary images. If the text components are set as marker, the resulted regions of watershed allow them to occupy a region which adapts according to their size and shape. These regions can help in localizing the neighboring components that may belong to the same line.

Since the approach is based on the analysis of the connected component in order to extract the lines, diacritical points may influence the results of the analysis and therefore should be removed before applying watershed transform. Local linear regions are detected by analyzing the spatial relationship of each resulted region with its neighbors and if one of the neighboring text components is located in its field of view then it will be linked to it by a line thus resulting in a new image where all neighboring text components are linked together. Finally watershed transform is applied again on this new image which results in detecting regions of the text lines.

1. Diacritical Points Removal

The proposed approach is sensitive to the presence of diacritical points since it is based on the analysis of the arrangement of the text component. So as a preprocessing step to remove these points, we use the technique in [2] which is based on the estimation of the stroke width of text component. Fig. 4 illustrates the document without diacritics.

After locating the lines, these points will be assigned to their corresponding words.


تمكنت مئات النساء الفلسطينيات من كسر
الحصار عن مسجد طوقته الدبابات
الاسرائيلية وسط بلدة بيت حانون شمال

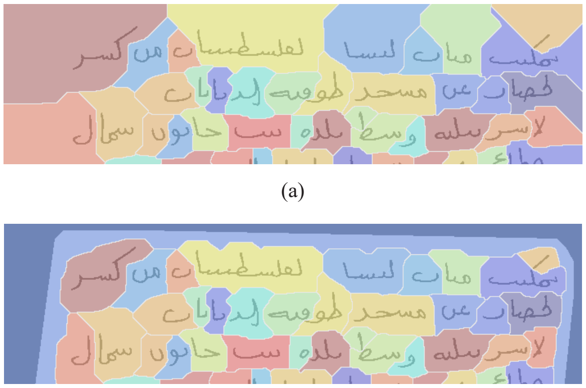
(Original image)

(Resulted image)

Fig. 4. Result of diacritical points removal.

2. Finding Local Linear Regions

After removing the diacritical points, we set the markers as text components and apply Watershed Transform and in order to prevent the flooding of the edge regions in the corners of the image (Fig. 5.a), we set a new flooding area represented by the inverse of the convex hull of all the text components. The result of this process is shown in Fig. 5.b.


(a)

(b)

Fig. 5. Result of watershed transform on binary document.

One of the characteristic of the watershed transform is that it keeps the alignment of the text lines. For each region, the analysis of the spatial arrangement of text components that may participate to the same line is restricted only to the neighboring regions which correspond to those intersecting the contour of the region in question after dilating its contour with a disk of two pixels. Fig. 6 illustrates an example of neighboring regions detection.

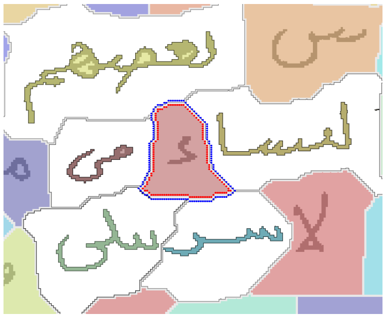


Fig. 6. Example of localizing the neighboring regions of the text component in the middle.

For any linear writing, words and characters in a local region are more or less aligned, and this remains valid even for documents with curved lines. Taking into consideration this fact, the idea is to search, for each text component, among its neighbors those that intersect with its field of view. As for the example in Fig. 7 the blue rectangle corresponds to the current components and the green rectangles correspond to text components that intersect the field of view and therefore are probably located in the same line as the current component.

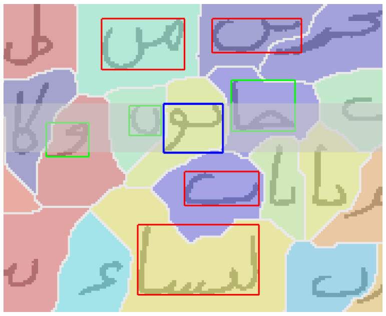


Fig. 7. Example of detection of the neighboring text components included in the field of view of the text component in the middle.

Applying this criterion as it is on the whole components, gives errors in the case of narrow gaps between adjacent lines where the bounding box of the neighboring component in the adjacent lines may intersect with the field of view of the component in question. This is illustrated in Fig. 8.

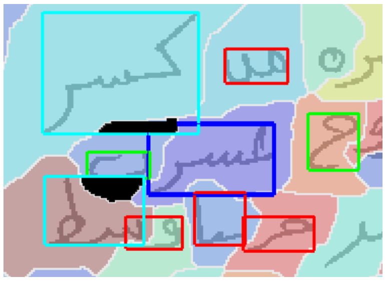


Fig. 8. Example of removing text component in the adjacent lines (black regions correspond to intersection of the components in cyan with others in the neighborhoods).

In order to resolve this issue, the idea is to remove text component in the neighboring regions with bounding box that contains portion of other text component region that intersect the field of view. In the case where tow text components participating in the field of view and intersecting each other we keep them both.

As shown in Fig. 9, other problems occur where text parts may be located in a place which makes it participate at the same time in more than on field of view thus allowing to link component in adjacent lines which result in merging the two lines together. Thus to resolve this, the component will be linked to the component in which it participate the most based on the percentage of participation described in [13].

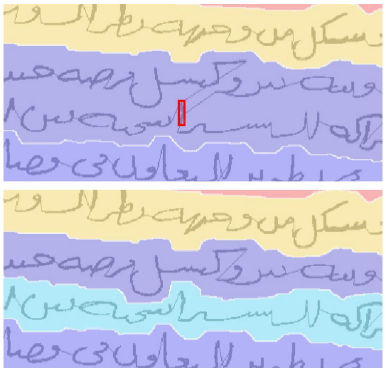


Fig. 9. Example of detection of text part (red rectangle) that participate in two adjacent lines.

3. Extraction of the Lines

As a result of finding local linear regions, we get a vector that contains for each text component the local ones that participate to the same line. A recursive function is used on this vector in order to find all components that are related to each other that will be linked together from their centroids by a line. Fig. 10 illustrates this process. Finally the watershed transform is again applied to the new image in order to estimate the location of the lines.

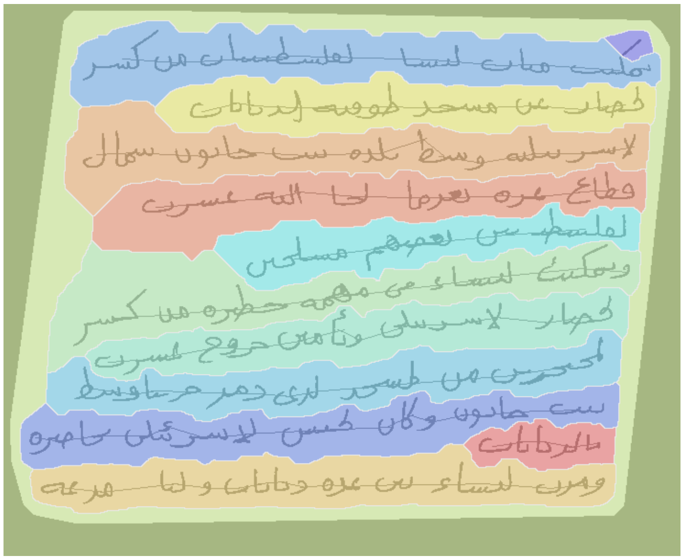


Fig. 10. Example of text lines detection on binary documents.

In some cases parts of characters or big diacritics may be located in interline and therefore remain isolated (not linked to any other component). Based on their width and height which is often smaller than the average words width and the average line height, these isolated components are detected and reassigned to the nearest line (Fig. 11).

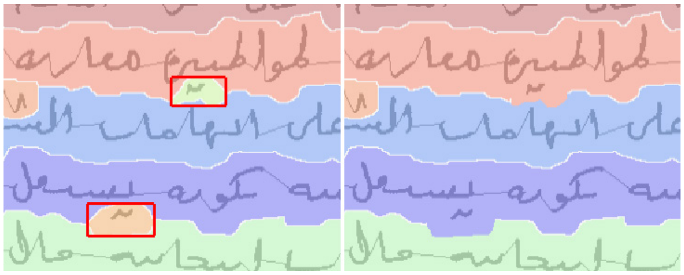


Fig. 11. Example of detection of diacritical points and broken characters (red rectangles).

C. Global Vision

We found interesting to investigate the use of horizontal projection profile as a global technique for the approximation of the base lines and then applying watershed transform on the extracted paths in order to extract the lines.

1. Projection Profile

One of the most used methods in text segmentation is called Projection Profile. Horizontal projection profile produces an histogram that represents for each line in the image: the number of black pixels [17, 18], the number of transitions black to white [19]. Locations of the maxima and minima values are detected, then the space between two consecutive minimums correspond to the location of the text line. This technique is well adapted for printed documents that contain straight lines. But for handwritten documents that present curved and short lines, the peaks do not reflect the location of the lines, also the presence of diacritical points in the Arabic document makes this technique too sensitive to this kind of documents (Fig. 12).

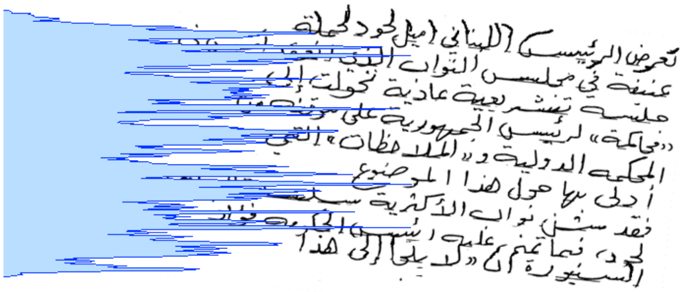


Fig. 12. Example of the histogram (in blue) of the horizontal projection profile.

In order to cope with the problem of curved lines, the idea is to divide the document into vertical slices and then analyzing each slice separately [20, 8].

2. Estimation of the Baseline

The first stage in [8] detects the medial seams (approximation of the orientation of each text line) using a projection profile matching approach. After splitting the page into slices a smoothed horizontal projection profiles is computed for each slice independently. Local maxima in two consecutive slices are matched in both directions, if a matching is found between two peaks a line is drawn between them, thus creating for each line a curve that goes through its peaks.

Due to the presence of diacritical points and ascenders and descenders, the histogram of the horizontal projection profile may contain lot of peaks, so in order to have a histogram that reflects the number and position of the lines, as in [8], we apply a cubic spline that smooths the histogram. Example of this process in shown in Fig. 13.

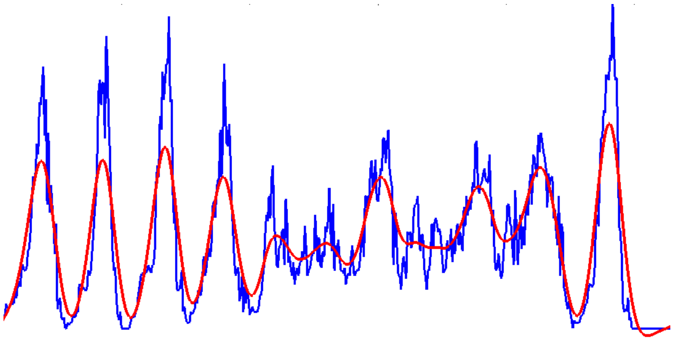


Fig. 13. Example of the histogram of the horizontal projection profile (in blue) and its smoothed version (in red).

We adapt the method in [8] by changing the direction of the matching between peaks of the slices in order to take into consideration the writing direction of the Arabic script which is from right to left, also we ignore the step of removing lines that start from intermediate column since in Arabic documents we may face this kind of situation. The step of extending the small lines is modified by simply taking the last coordinate of each line and creating a straight line between it and the corresponding coordinate in the last column of the image.

After extracting the diacritical points, the horizontal projection profile is applied as shown in Fig. 14.

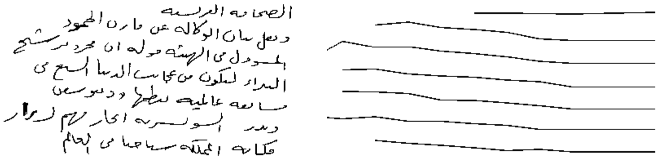


Fig. 14. Example of lines detection using matching horizontal projection profile.

3. Extraction of the Lines

The curves of the projection profile are dilated vertically and marked as locations for the watershed transform where the flood must start. Finally, the text lines correspond to the regions of the watershed transform. The result of this process is illustrated in Fig. 15.

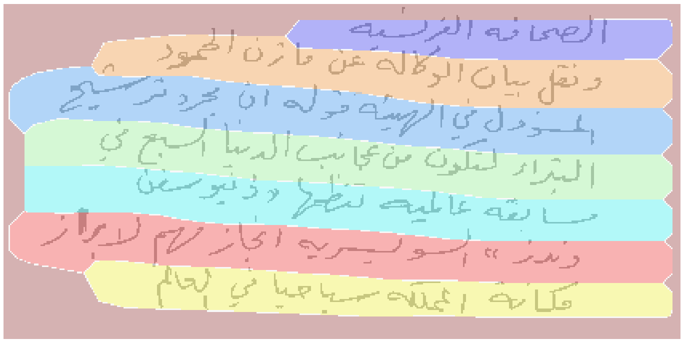
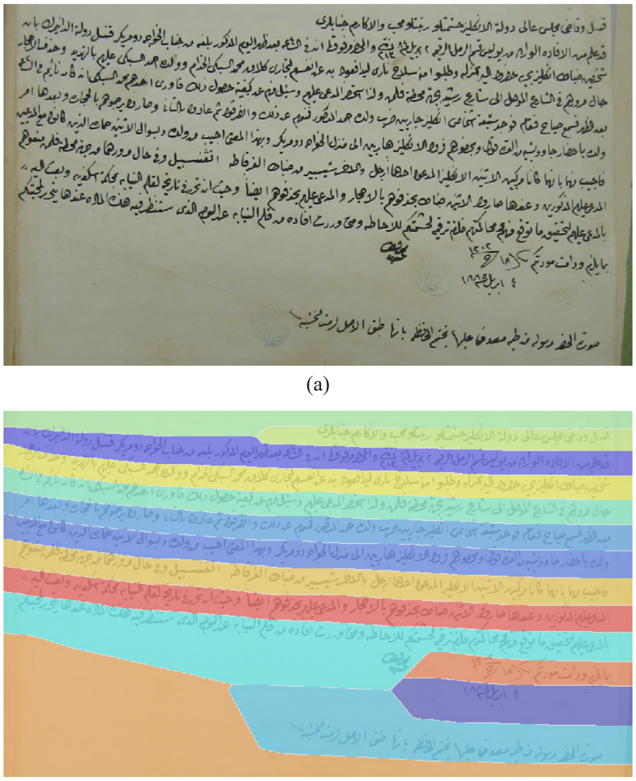


Fig. 15. Result of watershed transform on the projection profile lines superposed on the original image.

One of the advantages of this approach is its ability to work on grayscale documents. As illustrated in Fig. 16, the method detects efficiently the line event in the presence of noise.



(a)

(b)

Fig. 16. Example of text lines detection on gray level documents.

III. Comparative Analysis

In order to assess the accuracy of the proposed approach we use a subset of the publicly available data set "Handwritten Arabic Proximity Datasets" [21]. The F1-mesure score explained in [13] is used here too. Table I compares the score of the proposed method with previously proposed methods tested on the same samples.

TABLE I. Results of the Segmentation Rate using F1-score

Methods	F1-mesure
The method in [13]	90.5%
The method in [2]	94.3%
Watershed as pre-processing	89.4%
Watershed as post-processing	93.3%

As can be observed from the result it is clear that using watershed transform as post-processing step to the horizontal projection profile gives better result than using it as pre-processing step to the analysis of the linearity of connected component. This could be justified by the fact that the projection profile treats the document from a global vision by analyzing the peaks of text lines as if it searches the useful part of words which result in an approximation of the baseline. While by analyzing the connected components, the document is treated from a local vision and thus when constructing the lines we may deviate and this is mainly caused by the presence of touching lines where a given component may be linked to other, which is located in adjacent lines.

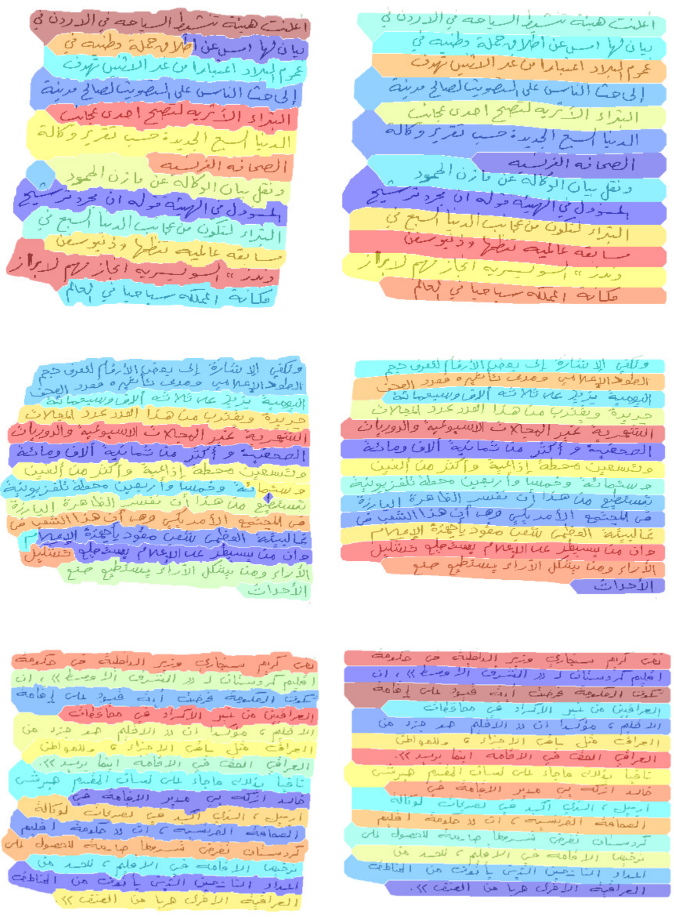
Another advantage of this approach is to treat an also gray level document which is benefic in the case of the presence of complicated noise that could not be removed in the binarized version of the document. Meanwhile, the drawback is that the approximation of the line is global; therefore, some parts of words may be broken in the extracted lines.

Table II recaps the main differences of the two approaches.

TABLE II. Differences between using Watershed Transform

	As pre-processing	As post-processing
Gary scale or Binary	Binary	Both
Sensitive to noise	Yes	No
Sensitive to touching lines	Yes	No
Broken characters	No	Yes
Multi-languages	Yes	Yes
Simple and effective	No	Yes

Fig. 17 illustrates some results of the two approaches for text lines detection from Arabic handwritten documents.



(a)

(b)

Fig. 17. Examples of handwritten text lines detection using the two proposed approaches using watershed transform as preprocessing (a) and as pos-processing (b).

IV. Conclusion

This paper investigates the use of watershed transform in the context of text lines segmentation from handwritten Arabic documents. The first approach treats the document from a local perspective by analyzing the arrangement of the connected component locally; while the second approach treats it from a global perspective by estimating the baseline. Tested on the same samples, the second approach even if it is coarse achieved better results and showed its less sensitivity to the presence of noise and touching lines.

As future work, we are willing to enhance the approach in order to prevent the broking of the characters and also to be able to treat historical and complex documents.

References

[1] Likforman-Sulem, L., Zahour, A., & Taconet, B. (2007). Text line segmentation of historical documents: a survey. International Journal on Document Analysis and Recognition, 9(2), 123-138.

[2] Boulid, Y., Souhar, A., & Elkettani, M. Y. (2016) Segmentation approach of Arabic manuscripts text lines based on multi agent systems. International Journal of. Computer Information Systems and Industrial Management, 8, 173–183.

[3] Khayyat, M., Lam, L., Suen, C. Y., Yin, F., & Liu, C. L. (2012, March). Arabic handwritten text line extraction by applying an adaptive mask to morphological dilation. In Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on (pp. 100-104). IEEE.

[4] Nikolaou, N., Makridis, M., Gatos, B., Stamatopoulos, N., & Papamarkos, N. (2010). Segmentation of historical machine-printed documents using Adaptive Run Length Smoothing and skeleton segmentation paths. Image and Vision Computing, 28(4), 590-604.

[5] Shi, Z., & Govindaraju, V. (2004). Line separation for complex document images using fuzzy runlength. In Document Image Analysis for Libraries, 2004. Proceedings. First International Workshop on (pp. 306-312). IEEE.

[6] Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., & Alaei, A. (2013, August). ICDAR 2013 handwriting segmentation contest. In Document Analysis and Recognition (ICDAR), 2013 12th International Conference on (pp. 1402-1406). IEEE.

[7] Saabni, R., Asi, A., & El-Sana, J. (2014). Text line extraction for historical document images. Pattern Recognition Letters, 35, 23-33.

[8] Arvanitopoulos, N., & Süsstrunk, S. (2014, September). Seam carving for text line extraction on color and grayscale historical manuscripts. In Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on (pp. 726-731). IEEE.

[9] Shi, Z., Setlur, S., & Govindaraju, V. (2009, July). A steerable directional local profile technique for extraction of handwritten arabic text lines. In Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on (pp. 176-180). IEEE.

[10] Pastor-Pellicer, J., Afzal, M. Z., Liwicki, M., & Castro-Bleda, M. J. (2016, April). Complete System for Text Line Extraction Using Convolutional Neural Networks and Watershed Transform. In Document Analysis Systems (DAS), 2016 12th IAPR Workshop on (pp. 30-35). IEEE.

[11] Kumar, J., Abd-Almageed, W., Kang, L., & Doermann, D. (2010, June). Handwritten Arabic text line segmentation using affinity propagation. In Proceedings of the 9th IAPR International Workshop on Document Analysis Systems (pp. 135-142). ACM.

[12] Kumar, J., Kang, L., Doermann, D., & Abd-Almageed, W. (2011, September). Segmentation of handwritten textlines in presence of touching components. In Document Analysis and Recognition (ICDAR), 2011 International Conference on (pp. 109-113). IEEE.

[13] Boulid, Y., Souhar, A., & El Kettani, M. E. Y. (2016). Detection of Text Lines of Handwritten Arabic Manuscripts using Markov Decision Processes. International Journal of Interactive Multimedia and Artificial Intelligence, 4(1), 31-36.

[14] Razak, Z., Zulkiflee, K., Idris, M. Y. I., Tamil, E. M., Noor, M. N. M., Salleh, R., ... & Yaacob, M. (2008). Off-line handwriting text line segmentation: A review. International Journal of Computer Science and Network Security, 8(7), 12-20.

[15] Oh, K., Kim, S., Na, I., & Kim, G. (2014). Text Line Segmentation using AHTC and Watershed Algorithm for Handwritten Document Images. International Journal of Contents, 10(3), 35-40.

[16] Meyer, F. (1994). Topographic distance and watershed lines. Signal processing, 38(1), 113-125.

[17] Bennasri, A., Zahour, A., & Taconet, B. (1999). Extraction des lignes d'un texte manuscrit arabe. In Vision Interface (Vol. 99, pp. 42-48).

[18] Nicolaou, A., & Gatos, B. (2009, July). Handwritten text line segmentation by shredding text into its lines. In Document Analysis and Recognition,

2009. ICDAR'09. 10th International Conference on (pp. 626-630). IEEE.

[19] Marti, U. V., & Bunke, H. (2001). On the influence of vocabulary size and language models in unconstrained handwritten text recognition. In Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on (pp. 260-265). IEEE.

[20] Zahour, A., Likforman-Sulem, L., Boussellaa, W., & Taconet, B. (2007, September). Text line segmentation of historical Arabic documents. In Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on (Vol. 1, pp. 138-142). IEEE.

[21] Handwritten Arabic Proximity Datasets. Language and Media Processing Laboratory. http://lampsrv02.umiacs.umd.edu/projdb/project.php?id=65.

[22] Boulid, Y., Souhar, A., Ameur, Elb. & Ouagague, Mly. M. (2017) Watershed transform for text lines extraction on binary Arabic handwritten documents. In Proceedings of BDCA'17, Tetouan, Morocco, March 29-30, 2017, 6 pages. https://doi.org/10.1145/3090354.3090444

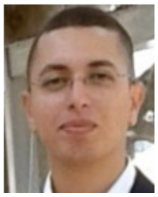
[23] Boulid, Y., Souhar, A., & Elkettani, M. E. Multi-agent Systems for Arabic Handwriting Recognition. International Journal of Interactive Multimedia and Artificial Intelligence, (2017, In Press), http://dx.doi.org/10.9781/ijimai.2017.03.012.

[24] Boulid, Y., Souhar, A., & Elkettani, M. E. (2017). Handwritten Character Recognition Based on the Specificity and the Singularity of the Arabic Language. International Journal of Interactive Multimedia and Artificial Intelligence,4(4), 45-53.

Abdelghani Souhar

Abdelghani Souhar is a full Professor of computer science at the University of Ibn Tofail, Faculty of science Kenitra Morocco. He received the M.S. degree in applied Mathematics in 1992, PhD degree in computer science in 1997 from the University of Mohammed 5 in Rabat-Morocco. His habilitation thesis concerned Modeling complex systems As Arabic handwritten recognition and an intelligent system for generating mesh.

His research interests include automatic processing of Arabic language, Modeling complex systems, CAE/CAD, and Artificial Intelligence.



Youssef Boulid

Youssef Boulid received his M.S. degree in Decision Support Systems and Project Management in 2012 and PhD degree in Computer Science in 2016 from University Ibn Tofail, Faculty of science, Kenitra-Morocco. Now he works as a researcher at Maxware Technology Kénitra – Morocco.

His research interests include image processing, handwritten document analysis, Arabic handwritten recognition and artificial intelligence.

El Bachir Ameur

El Bachir Ameur is a full Professor of computer science at the University of Ibn Tofail, Faculty of science Kenitra (Morocco). In 2002 he received the Ph. D. degree in numerical analysis and computer science from the University of Mohamed I Oujda (Morocco).

His Ph. D. concerned approximation of curves and surfaces by spline and wavelet functions.

His research interest concerns approximation and reconstruction of 2D/3D surfaces by spline and wavelet, signal and image processing, watermarking and steganography.

Mly Moustafa Ouagague

Mly Moustafa Ouagague received his M.S. degree in Decision Support Systems in 2001 from Blaise Pascal University. Now he works as a researcher at Maxware Technology Kénitra - Morocco.

His research interests include artificial intelligence, complex systems, image processing and serious games.