

Handwritten Arabic Documents Segmentation into Text Lines using Seam Carving

M. Daldali, A. Souhar *

LaRIT laboratory, Faculty of science, University Ibn Tofail, Kénitra (Morocco)

Received 22 December 2017 | Accepted 3 May 2018 | Published 8 June 2018



ABSTRACT

Inspired from human perception and common text documents characteristics based on readability constraints, an Arabic text line segmentation approach is proposed using seam carving. Taking the gray scale of the image as input data, this technique offers better results at extracting handwritten text lines without the need for the binary representation of the document image. In addition to its fast processing time, its versatility permits to process a multitude of document types, especially documents presenting low text-to-background contrast such as degraded historical manuscripts or complex writing styles like cursive handwriting. Even if our focus in this paper was on Arabic text segmentation, this method is language independent. Tests on a public database of 123 handwritten Arabic documents showed a line detection rate of 97.5% for a matching score of 90%.

KEYWORDS

Text Line Segmentation, Arabic Documents, Handwritten Character Recognition, Projection Profile, Seam Carving.

DOI: 10.9781/ijimai.2018.06.002

I. INTRODUCTION

WITH the advent of digital means to share information, people are slowly abandoning paper as a medium and use digital devices and technologies instead. With the volume of processed information growing every day, businesses, organizations and public services adopt new digital technologies instead of paper documents. This transition brings a real need for scalable optical character recognition (OCR) systems capable of converting paper documents - handwritten or printed - into digital formats.

In addition to modern handwritten texts, there is a huge amount of historical documents in libraries and archives that have not been exploited yet. Systems capable of processing such manuscripts will be beneficial for building indexed databases and computer systems helping to preserve the cultural heritage that these documents represent, as well as easing the research process in historical libraries, which will be beneficial for researchers across disciplines.

OCR systems use many steps to process document images. In general, these steps can be summarized in four main steps, text blocks identifying, text lines segmentation, word segmentation, and finally character recognition. This is making the text lines segmentation a very important phase in the recognition process since the recognition rate depends strongly on it, as any error occurring at this stage will highly affect the recognition efficiency.

Most of the proposed solutions in the literature are based on Connected Components (CC) analysis. This type of approaches has shown some struggles dealing with low text-to-background contrast images such as historical manuscripts or damaged documents because of the binarization pre-processing as shown in [1], where binarization

may ensue critical information loss. An approach based on less filtered data would offer better results on such documents. The CC analysis may also be an overkill when applied to well-structured text lines, like in the case of printed documents.

Human readers can identify text lines without knowing exactly their content or the language which they are written in. This can be noticed when the document is presented to the reader from a considerable distance. As a result, we can safely assume that the text line segmentation is based on identifying the interline space where each line is defined by their neighbors as previously stated in the watershed approach [2], where the authors used CC analysis in order to define the locations of high and low information density, recognizing the text areas from the empty interlines. In a similar way, the seam carving is an image processing tool based on the notion of identifying high and low information areas on an image. Relying on dynamic programming, this can offer better computation efficiency and can show new opportunities for developing more complex and efficient approaches. Additionally, the seam carving offers an additional layer of versatility as only the gray scale of the document image is used, which can make an OCR system capable of processing many types of documents.

The next chapters will discuss the seam carving technique, its text line segmentation implementation, and practical tests results. Firstly, an overview on related works will be conducted in the second chapter. Followed by a presentation of the seam carving technique and its use for text line segmentation, then in the fourth chapter, practical test and accuracy results will be conducted. Finally, the robustness of this seam carving approach is discussed with practical illustrations leading to conclusions and perspectives for future works.

II. RELATED WORKS

In general, there are two types of text lines segmentation systems: either by searching in a structural way for related physical units such as Connected Components, or through analyzing the documents in search for global image features.

* Corresponding author.

E-mail addresses: mehdi.daldali@outlook.com (M. Daldali), houssouhar@gmail.com (A. Souhar).

In the article [2], a watershed approach was used to detect text components locations then trying to find the best arranged parts to form text lines, following a local vision or as more commonly known as “bottom-up” analysis. The segmentation process begins by identifying the multiple text components present on the image, then the use of the watershed technique to split the document image into well-defined areas containing each component. These areas are then analyzed in order to clustering them into well-aligned text lines. The same can be said for the other [3] approach where a multi-agent system was used to process each step of the line segmentation from de-noising to dealing with touching components. In case of [4], the problem was modeled into a decision making situation. Starting from a text component, the system tries to find the best next text component to add to a collection until a text line is finally defined using a Markov Decision Process (MDP). The work presented in [5] was also based on connected components analysis where text components are categorized into subsets according to their importance in defining potential text lines that are later analyzed using the Hough Transform. While the last works are based on text components detection, the approach used in [6] rely heavily on binarization of the document image in order to differentiate between text ink and blank spaces, these spaces are then used to define potential text blocks and text lines across the document.

In contrast, the methods using “top-down” analysis do not rely on rough estimations of physical components to detect text lines which require a binary representation of the document image. By using less filtered data avoiding binarization, these methods can extract more global features resulting in a better accuracy especially on handwritten documents using cursive styles like Arabic manuscripts or degraded documents showing a low text-to-background contrast as shown in [1] where the seam carving technique was used combined with a base line approximation technique in order to guide the seam carving process through the inter-line space. Unfortunately, the approach used was inefficient in case of informal handwritten documents where paragraphs and lines show a high amount of discord. [7] was a prior work to [1], and used the seam carving technique in a different context. The document image is first converted to binary representation, then a distance transform was used to measure distance separating the text ink (pixels), then by using this distance as energy, the seam carving technique was used to find the least cost seams passing through the text components. These approximations were then used to group the components into aligned text lines.

Interestingly, in [2] the authors used an hybrid approach to improve the system accuracy. Even if the analysis relies heavily on CC analysis, a projection profile analysis is used as an extension in order to help deciding which components are more likely to be from the same text line following a global vision of the document. This approach is based on watershed technique which relies on connected components analysis to create a similar image partitioning to the Voronoi diagram, where these partitions are then analyzed to merge them into text lines.

The practical tests showed that the watershed approach is efficiently accurate in most of the cases, but relying on CC analysis makes it highly inefficient in case of damaged or low-contrast documents. Additionally, the approach is hard to improve on, notably in case of runtime speedup using parallelization.

By using the seam carving technique, we were able to achieve better versatility which will offer new grounds to improve the segmentation accuracy and ability to build more complex systems.

III. SEAM CARVING AND LINE SEGMENTATION

A. Seam Carving

The Seam Carving operator [8] was proposed first as a solution for resizing images. When re-targeting images, the classical way was to

either stretch the image by interpolating data, cropping the image, or by manual processing.

Seam Carving re-targets images automatically with the least distortion possible in order to preserve important areas dimensions, size ratios and details clarity in a content aware fashion. To achieve this goal, the method uses an energy map to calculate the least-cost seam to remove or duplicate. In more simple words, this seam travels from side to side through a path presenting the least of details on the image. The energy function used can be of several types such as gradient magnitude, visual saliency, and more.

B. Energy Map

In case of document images, the energy map must reflect accurately the position of pixels representing the text. One of the best energy maps is the gradient magnitude which offers good efficiency for low complexity, describing accurately the outline of text lines strokes. This map can be obtained by using a Sobel operator which is a matrix convolution of the original grayscale image by a weighted mask set adequately to get the local pixel gradient magnitude from the neighboring values.

After applying a Gaussian filter of a standard deviation σ for noise reduction purposes, the gradient magnitude is then computed using the simple energy function:

$$e(I) = \left| \frac{\partial}{\partial x} I \right| + \left| \frac{\partial}{\partial y} I \right| \quad (1)$$

with $I \in \mathbb{R}^{n \times m}$ the original document image converted to gray scale having n rows and m columns. This is done by calculating the Sobel operator for the vertical and horizontal axis by matrix convolutions:

$$\frac{\partial}{\partial x} I = \begin{pmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{pmatrix} * I \quad (2)$$

$$\frac{\partial}{\partial y} I = \begin{pmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} * I \quad (3)$$

After computing the energy map, a cost map is then computed using dynamic programming in order to deduce the least-cost path from one side of the image to the other.

C. Dynamic Programming

The cost map is generated from the energy map using a dynamic programming technique where each position on the cost map equivalent to the pixel position on the image has as value the sum of its own energy, and the least cost value of the 3 neighboring pixel positions from the last column processed:

$$C(I_{i,j}) = e(I_{i,j}) + \min_{k \in \{-1,0,1\}} (C(I_{i-1,j+k})) \quad (4)$$

Where I_{ij} designates the image value at the i^{th} row and the j^{th} column, with the origin at the upper left corner of the image, and $C(I_{i,j})$ the cost value of the position (i, j) . To find the least-cost seam, the algorithm backtracks starting from the position with the minimum cost value on the last column of the cost map (Fig. 1).

The resulting seams have no constraints while path-finding as shown in Fig. 2. In the case of text lines, the seam calculation must be guided to not overlap potential text by staying in between the lines. For this purpose, medial seams are introduced to guide the seam carving process, using lines' axis approximations.

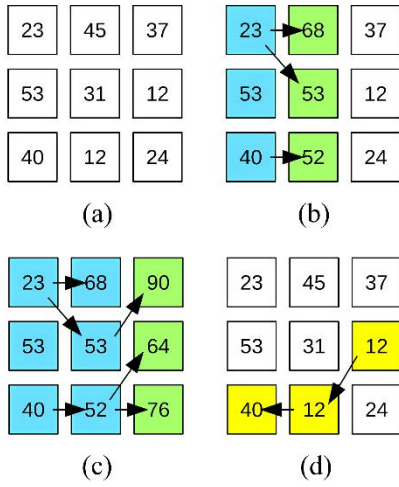


Fig. 1. Example of the dynamic programming used for Seam Carving: the original energy map (a), computing the second column costs (b), calculating the last column costs (c).

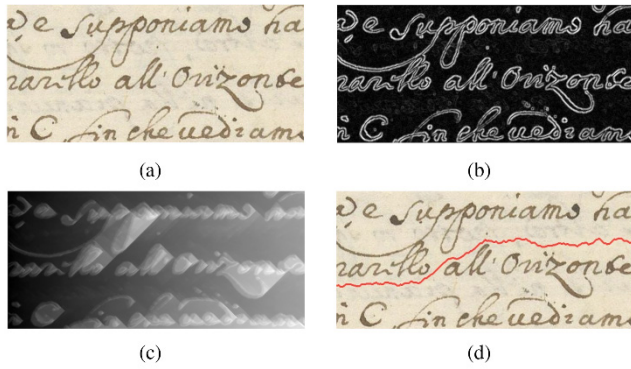


Fig. 2. Example of Seam Carving used on a manuscript image, where (a) is the original image, (b) the gradient magnitude of the image used as an energy map, (c) an intensity representation.

D. Medial Seams

The medial seams are approximations of the text lines axis computed by using a projection profile matching approach similar to the work presented in [9]. The image is split vertically into s slices of equal width w , then the energy map is used to approximate the axis positions on each slice. Calculating the projection profile is made by summing the values of the energy map for each row of the image slice, and finally extracting local maximum positions after smoothing the resulting data with a smoothing cubic spline filter (Fig. 3) using a smooth factor $smoothFactor$:

$$smooth = smoothFactor \times \sum_{i=1}^n PP(i) \quad (5)$$

Where n the number of rows present in the processed image, $PP(i)$ is the projection profile value at the i^{th} row, and $smooth$, the constraint used while calculating the smoothed spline SP , this constraint is defined as:

$$smooth \geq \sum_{i=1}^n (PP(i) - SP(i))^2 \quad (6)$$

Ensuring that the resulting smoothed histogram is true to the original as possible as it can be. Thus the $smoothFactor$ can be considered as

a maximal threshold limiting the difference between the original raw data and the smoothed representation.

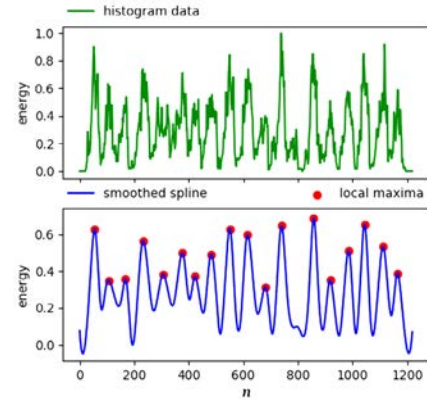


Fig. 3. The projection profiles of a document image slice, the smoothed cubic spline, and the local maximum positions found.

For each local maximums set of each slice, each local maximum 2D position is matched with the closest local maximum position from the adjacent slice using a distance matrix of size $n_1 \times n_2$ computed by calculating the distances between each pair of local maximums from each pair of adjacent slices where n_1 and n_2 is the number of local maximums from each slice:

$$(d_{ij}) = \begin{pmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,n_1} \\ d_{2,1} & d_{2,2} & \dots & d_{2,n_1} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n_2,1} & d_{n_2,2} & \dots & d_{n_2,n_1} \end{pmatrix} \in \mathbb{R}^{n_1 \times n_2} \quad (7)$$

d_{ij} denotes the Euclidean distance between the position of the local maximum of index i from the first slice and the position of index j from the other adjacent slice. To match these local maximums, all positions pairs of distance d_{ij} which values are simultaneously a minimum column-wise and row-wise on the distance matrix and where:

$$\frac{w}{d_{ij}} \geq \sqrt{2}/2 \quad (8)$$

are considered a matched pair of local maxima, and thus, the pair is considered as a piece representing the position of a text line axis passing over the two slices. The last constraint (8) translates as a skew degree limit (Fig. 4) while matching local maximums, with the value $\sqrt{2}/2$ equivalent to the cosine of a 45° angle. This value was inspired by the results of a neuroscience research in [10] answering questions about degraded text readability, showing a better reader response when the text skew angle is under 45° from the horizon in relation to the reader's eye.



Fig. 4. An example of the skew angle α between the 2nd and 3rd positions starting from the right on a single Arabic text line.

For deducing the complete lines axis approximations, the solution on [1] proposed a connected components analysis approach (Fig. 5) used to fuse piece-wise approximations with some drastic decisions to avoid problems such as ignoring any seam not starting from the first slice. This makes the algorithm inefficient in most of handwritten documents where text lines present noticeable irregularities.

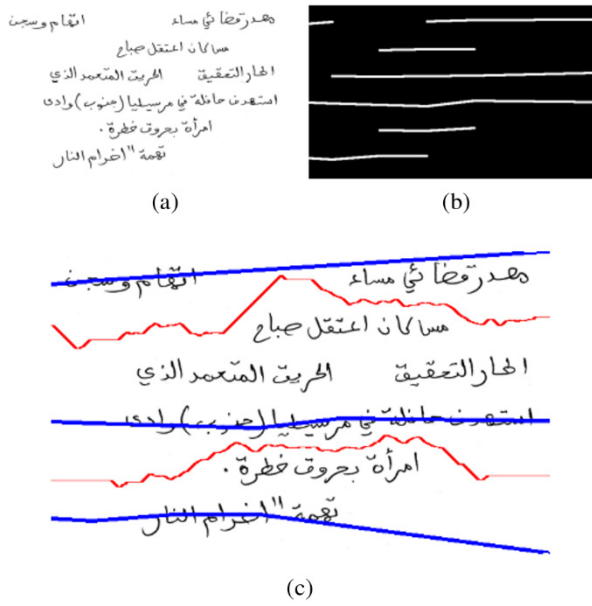


Fig. 5. Example of line extraction using the method on [1]: (a) the original image, (b) the piece-wise approximations detected represented as components, and (c) the resulting segmentation.

We propose a more numerical and geometrical approach, based on interline distances. After defining the matched maximums between all of the document image slices, the piece-wise approximations are assembled by matching extremities into full lines axis approximations using the algorithm on Fig.6, with an example shown on Fig.7.

In order to fuse discontinued lines and eliminate false positives in case of under-smoothing the projection profile histograms, all positions of each line are represented on a lines map, a sparse matrix of size $s \times l$ where s is the number of the vertical slices and l the number of lines detected, while filling the unavailable data with a blank value.

The blanks are then filled using interline distance approximation and the minimum interline distance $minh$ defined as two thirds of h the median of the distances between each vertically adjacent local maximums positions.

Algorithm

Input:
Pieces: list of detected pieces sorted by column
 Output:
Lines: list of detected lines axis approximations

```

for each piece in Pieces do
    unmatched ← True
    for each line in Lines do
        if line[-1]=piece[0] do
            line ← Concatenate( line, piece )
            unmatched ← False
            break
        end if
    end for
    if unmatched = True do
        Lines ← Insert( piece, Lines )
    end if
end for
    
```

Fig. 6. Algorithm used for assembling the piece-wise approximations to generate full lines axis approximations. Where line[-1] represents the last element of the line.

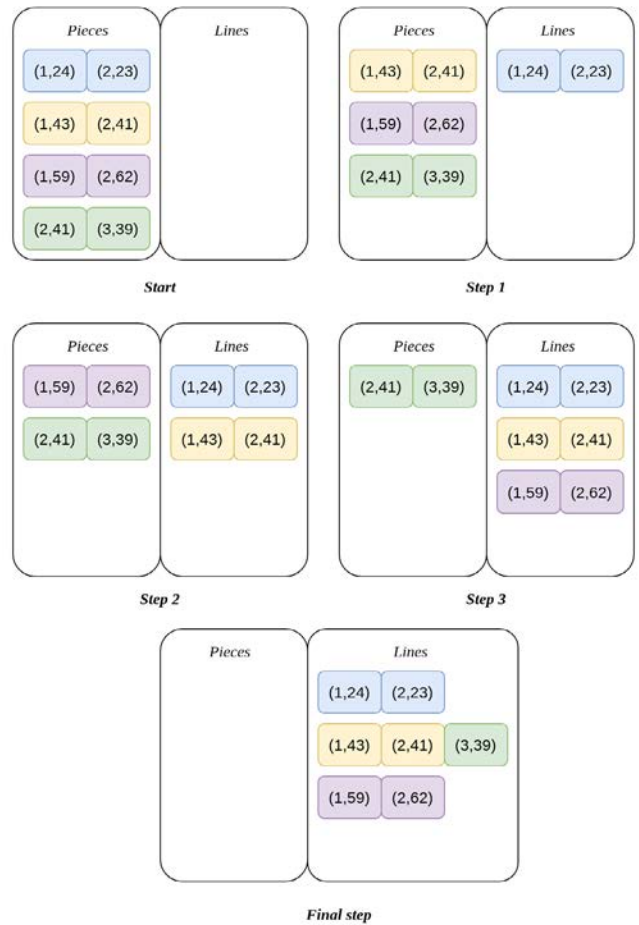


Fig. 7. An example showing the execution of the proposed piece-wise approximation matching algorithm. The values used represent the 2D coordinates of the local maximums positions detected by the projection profile method.

This distance $minh$ is inspired from the natural spacing of handwritten text lines (also in printed documents). To ensure an acceptable level of readability, a minimum distance must be kept between each two neighboring baselines (the virtual lines on which the text components forming text lines are arranged). Knowing that almost in every handwritten text, there is a number of characters that extend either upward or downward, these extending characters tend to be roughly 3/2 times taller than normal height characters, which makes the minimal distance between each two adjacent baselines around 2/3 of the typical interline distance as shown in Fig. 8.

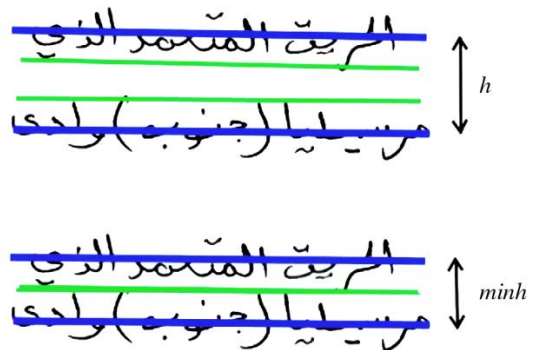


Fig. 8. The approximate minimum interline distance $minh$ is equal to 2/3 of h , the typical distance between adjacent text lines' axis across the text block.

The interline distance between each consecutive line is calculated by measuring the mean distance between the position on the same column. If no consecutive positions exist, the distance is then calculated using the mean of the ordinate values of each line. Then comes the time to fill the missing values by using the interline distance between the filled line and the line above, without exceeding the value on the line under. A practical example and the algorithm are shown on Fig. 9 and Fig. 10.

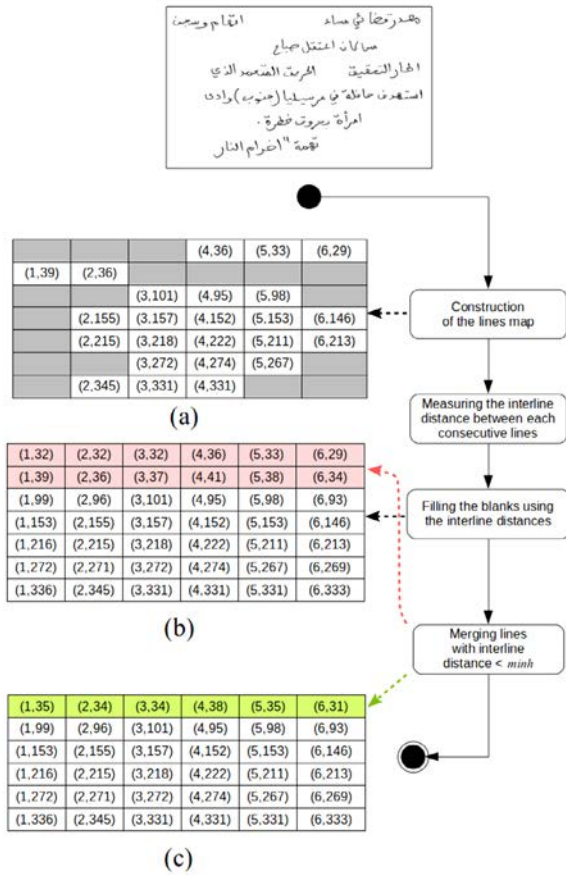


Fig. 9. Example showing the process used for post-processing the lines axis approximations data to generate the lines map: (a) the initial sparse matrix, (b) the filled lines map, (c) the final lines map generated from (b) by merging the lines with the interline distance less than $minh$.

Algorithm

Input:

LinesMap: map of detected lines' axis approximations

IntDist: interline distances

Output:

LinesMap: the completed map

```

for each line(i) in LinesMap do
  for each value(i,j) in line(i) do
    if value(i,j) = NULL do
      value(i,j) ← value(i-1,j)+IntDist[i]
      if value(i+1,j) ≠ NULL
        and value(i,j) ≥ value(i+1,j) do
          value(i,j) ← value(i+1,j)-1
        end if
      end if
    end if
  end for
end for
  
```

Fig. 10. The algorithm used to fill the unavailable values on the lines map.

This new method showed better results than the one used in [1], and was able to process many documents exhibiting multiple irregularities, approximating the lines axis accurately. Fig. 11 shows the results when processing the same image used in Fig. 5.

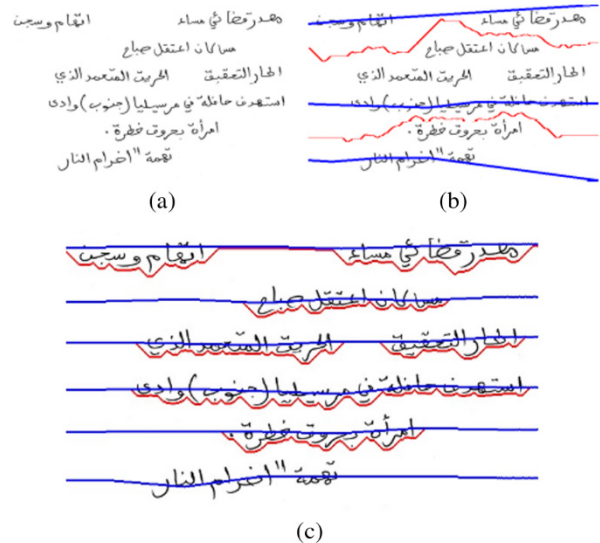


Fig.11. Comparison between method [1] and the method proposed using the same parameters: (a) the original image, (b) the resulting seams from method [1], and (c) the result using the proposed method.

IV. EXPERIMENTATION AND EVALUATION

To assess the accuracy of the proposed approach and to compare it to other methods, we faced a lack of test databases for gray-scale and color documents as it brings more challenges to define ground-truths in these cases.

Instead of developing a new database for gray-scale document images, we simply adapted the testing procedure proposed to the connected components oriented data base to suite our line extraction approach. This public dataset has 123 images of Arabic handwritten documents with a total of 1974 lines, available for download (Proximity Dataset, LMP Lab [11]). The method used to evaluate the performance of a system using this database is based on counting the number of matches between the extracted lines and the lines in the ground truth. And so, a criterion called Match Score is used to evaluate the number of detected zones according to the intersection of the sets of pixels on the results with those of the ground truth.

Unfortunately, this criterion is incompatible with the seam carving approach as the extracted lines are image stripes rather than connected components.

In order to circumvent the incompatibility issue, we used a new match score criterion MS2 in order to use the same approach of the harmonic mean F1-Score.

First, for each ground-truth of a document, we converted each line data into a set of pixels g_p , then, for each line zone extracted r_p , the new matching score criterion becomes:

$$MS_2(r_i, g_j) = 2 \times \frac{linePrecision \times lineRecall}{linePrecision + lineRecall} \quad (9)$$

$$linePrecision = \frac{TP_p}{TP_p + FP_p} \quad (10)$$

$$lineRecall = \frac{TP_p}{TP_p + FN_p} \quad (11)$$

$$TP_p = T(P(r_i) \cap P(g_j)) \quad (12)$$

$$FN_p = T(P(g_j)) - TP_p \quad (13)$$

$$FP_p = T(P(r_i) \cap (P(g_{j-1}) \cup P(g_{j+1}))) \quad (14)$$

Where TP_p is the number of pixels from the ground truth detected on the zone extracted (True Positive pixels), FN_p the undetected pixels from the line's ground truth (False Negative pixels), and FP_p the pixels from the ground truth of adjacent lines detected on the extracted zone (False Positive pixels). P corresponds to pixels that represent a zone and T is an operator that counts the number of pixels in each zone.

Finally, we continue the evaluation using the F1-Score (15) with the new criterion. If the MS2 score for a line is found above a threshold, it is counted as true positive TP. If the resulted lines do not match any line in the ground truth it is counted as false positive FP, the lines in the data set that are left unmatched are considered false negative FN.

$$F1\ Score = 2 \times \frac{precision \times recall}{precision + recall} \quad (15)$$

$$precision = \frac{TP}{TP + FP} \quad (16)$$

$$recall = \frac{TP}{TP + FN} \quad (17)$$

According to the tests, we obtained an F1-Score respectively of 94.4% and 90.2% for the MS2-threshold values of 90% and 95%. Out of 123 varied images in terms of text density, writing styles, and skew angles, 18 images showed an MS2 score less than 90%.

This is mainly related to the parameters values choice. The more vertical slices used, the more the algorithm is able to sample the irregularities of the document image. On the other hand, the cubic spline smoothing factor is a more delicate parameter where excessive smoothing can discard crucial local maximums and less smoothing can induce more false detections. As an example, we manually tried multiple parameters to process one image of these 18 images. Table I shows the different results for different parameters values (see Fig. 12 for visual results).

TABLE I. RESULTS OF USING DIFFERENT PARAMETER VALUES

| | PARAMETERS | | | F1-Score | Execution time (sec) |
|-----|------------|----------|--------------|-----------|----------------------|
| | s | σ | smoothFactor | | |
| | | | | MS2 = 90% | |
| (a) | 15 | 15 | 0.02 | 68.75% | 26.4 |
| (b) | 15 | 25 | 0.02 | 80.00% | 27.4 |
| (c) | 15 | 20 | 0.02 | 92.86% | 25.1 |
| (d) | 15 | 20 | 0.03 | 100% | 24.9 |

Results of using different parameter values on the same document image. s , σ , and $smoothFactor$, are respectively the number of vertical slices, the standard deviation used by the Gaussian filter, and the smoothing spline factor. See Fig. 12. (a), (b), (c), and (d) for illustrated results.

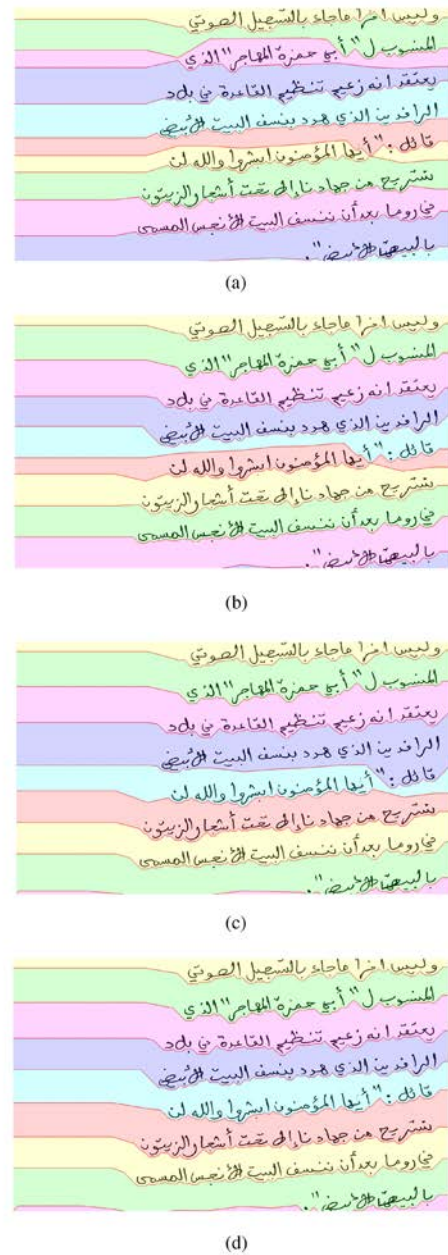


Fig. 12. Segmentation results of the tests shown in Table I from (a) the worst score to (d) the best score of 100%.

On a side note, some images' ground-truth on the dataset are not accurate. Some of these images show a text with a high amount of touching and overlapping components, but due to the way the ground truth is defined, the high redundancy of multi-labeled pixels makes the ground-truth nearly unusable (Fig. 13). Using a better quality dataset as a benchmark could reflect more accurately the efficiency of the proposed approach.

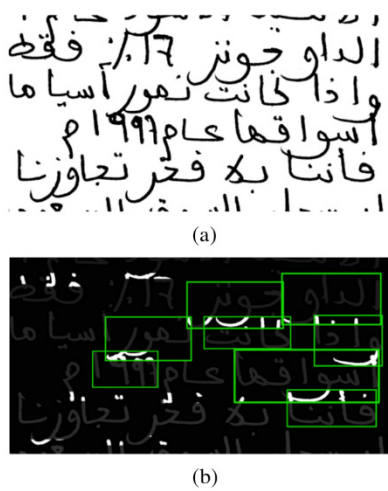


Fig. 13. Example of the overlapping components' ground-truth (convex envelopes in green) and the redundant pixels represented in white: (a) the original image, (b) the reconstructed image from the ground truth.

Even if the accuracy is sensible to parameters values, most of the inaccuracy is caused by undetected text lines due to their considerably short length ranging from single words to simple punctuation symbols.

Choosing different parameters manually for the 18 images, showed a non-negligible improvement of the detection rate. Table II compares the score of the proposed method with other methods tested on the same data set.

TABLE II. RESULTS OF DIFFERENT METHODS

| METHODS | F1-Score | |
|----------------------------|--------------|--------------|
| | MS2 = 90% | MS2 = 95% |
| Souhar et al. [2] | Not Reported | 93.3% |
| Boulid et al. [3] | 97.4% | 94.3% |
| Boulid et al. [4] | 95.8% | 90.5% |
| Kumar et al. [12] | 95.6% | 90.9% |
| Kumar et al. [13] | 98.8% | Not Reported |
| Zhang et al. [14] | 98.41% | Not Reported |
| The proposed method | 97.5% | 93.4% |

Results of different methods obtained on the original data in (Proximity Dataset, LMP Lab [11]).

In case of the matching score threshold of 90%, the precision and recall were respectively 98.44% and 96.39%. This shows that the proposed approach only suffers from false negatives due to the slices width that can exceed text lines length in some cases, making the detection of such lines impossible during the projection profile maxima matching step.

V. ROBUSTNESS

The text lines segmentation using seam carving use a global approach giving it more flexibility for adapting to various situations making it a language-independent method. It can tolerate different degrees of writing irregularities such as skew angles, irregular writing size, line discontinuities and low text-to-background contrast, frequent problems faced while processing handwritten document images and historical manuscripts (Fig. 14).

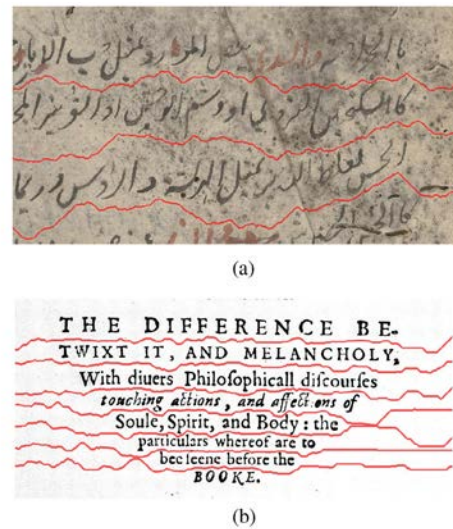


Fig. 14: Examples showing the seam carving text line segmentation on challenging documents: (a) damaged Arabic manuscript, and (b) a gray scale printed document with irregular font types and sizes.

In Fig. 15 many segmentation examples are shown. The method can be applied to languages written horizontally such as Arabic (right to left) or Latin (left to right), but also to exotic languages like Japanese and Chinese scrolls written vertically. This is simply achieved by using a transposed copy of the document image as input, and converting the horizontal seams to vertical seams afterwards like on Fig. 15. Another way is to calculate the cost map vertically instead of horizontally, in order to compute the vertical least-cost separating seams.



Fig. 15. Examples showing the adaptability of the seam carving text line segmentation, by processing documents of different languages, (a) Arabic manuscript, (b) Japanese scroll, and (c) Latin manuscript.

In case of binary document images, the Gaussian filter used for a noise reduction purpose before the gradient magnitude is computed, can affect the results. By increasing the Gaussian filter's kernel size it's possible to dilate the high energy areas, fusing the diacritics to the nearest text line and adding more clearance between the separating seam and the actual lines. Bigger kernel size, means less details are processed while computing the energy map.

To avoid this inconvenience, a small kernel size can be used only to reduce the image noise, then after computing the energy map, another Gaussian filter with a bigger kernel size can be applied on the energy map to dilate the high energy areas, an example is shown on Fig. 16 demonstrating the effect.

In the proposed implementation, the kernel size $ksize$ of the Gaussian filter is dynamically calculated relatively to the standard deviation σ :

$$ksize = (\sigma \times 2) + 1 \quad (18)$$

ensuring that the resulting convolution matrix can sample the variations of the Gaussian bell.

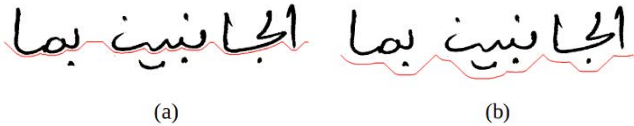


Fig. 16. The kernel size of the Gaussian filter used for a noise reduction purpose, can affect the result on document images with very high text-to-background contrast such as binary images: (a) use of a small kernel size can result in wrong segmentation of diacritics, this can be avoided by using a bigger kernel size (b).

VI. CONCLUSION

The seam carving approach to text line segmentation is an agile method that can be applied to different styles of writing whether it be printed or handwritten, and in different languages, which makes it a language-independent approach, and able to process multilingual documents.

It must be mentioned that the detection rate of the seam carving segmentation proves its efficiency and adaptability to the different handwriting styles present in the benchmarking dataset used, showing promising results.

In future works, we intend to improve this method on several points, notably improving the runtime performance of the implementation by using parallelism and simplifying the algorithm parameters choice based on each document image features, adding additional robustness.

ACKNOWLEDGMENT

We thank the Wellcome Library for providing a wide collection of old manuscript and digitized documents for free under creative commons license "Attribution 4.0 International (CC BY 4.0)" which grant researchers and enthusiasts the access to rare and interesting historical content, and we hope that other libraries take this as a model for sharing the knowledge and cultural heritage to the research community.

REFERENCES

- [1] Arvanitopoulos, N., & Süssstrunk, 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.
- [2] Souhar, A., Boulid, Y., Ameer, E., & Ouagague, M. (2017). Segmentation of Arabic Handwritten Documents into Text Lines using Watershed Transform. *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(6), 96-102.
- [3] Boulid, Y., Souhar, A., & Elkettani, M. Y. (2015, December). Arabic handwritten text line extraction using connected component analysis from a multi agent perspective. In *Intelligent Systems Design and Applications (ISDA)*, 2015 15th International Conference on (pp. 80-87). IEEE.
- [4] 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.

- [5] Louloudis, G., Gatos, B., Pratikakis, I., & Halatsis, C. (2008). Text line detection in handwritten documents. *Pattern Recognition*, 41(12), 3758-3772.
- [6] Nikolaou, N., Makridis, M., Gatos, B., Stamatoopoulos, 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.
- [7] Saabni, R., & El-Sana, J. (2011, September). Language-independent text lines extraction using seam carving. In *Document Analysis and Recognition (ICDAR)*, 2011 International Conference on (pp. 563-568). IEEE.
- [8] Avidan, S., & Shamir, A. (2007, August). Seam carving for content-aware image resizing. In *ACM Transactions on graphics (TOG)* (Vol. 26, No. 3, p. 10). ACM.
- [9] Liwicki, M., Indermuhle, E., & Bunke, H. (2007, September). On-line handwritten text line detection using dynamic programming. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on* (Vol. 1, pp. 447-451). IEEE.
- [10] Cohen, L., Dehaene, S., Vinckier, F., Jobert, A., & Montavont, A. (2008). Reading normal and degraded words: contribution of the dorsal and ventral visual pathways. *Neuroimage*, 40(1), 353-366.
- [11] Handwritten Arabic Proximity Datasets. Language and Media Processing Laboratory. <https://lampsrv02.umiacs.umd.edu/projdb/project.php?id=65>
- [12] 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.
- [13] 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.
- [14] Zhang, X., & Tan, C. L. (2014, September). Text line segmentation for handwritten documents using constrained seam carving. In *Frontiers in Handwriting Recognition (ICFHR)*, 2014 14th International Conference on (pp. 98-103). IEEE.



Mehdi Daldali

Mehdi Daldali received the License degree in mathematics and informatics in 2015 from University Ibn Tofaïl, Faculty of Science, Kénitra, Morocco. Currently, he is attending a Master's degree in Big Data and Cloud Computing at the same University, and expected to graduate in 2018. In 2017, he was certified as "Big Data Developer" and "Business Intelligence Analyst" by the IBM MEA Academy. His research interests include high performance computing and pattern recognition.



Abdelghani Souhar

Abdelghani Souhar is a full Professor of computer science at the University of Ibn Tofaïl, Faculty of science Kénitra 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.