A new Efficient Graphemes Segmentation Technique for Offline Arabic Handwriting

Hesham M. Eraqi and Sherif Abdelazeem
Electronics Engineering Department
The American University in Cairo
Cairo, Egypt
hesham.eraqi@gmail.com, shazeem@aucegypt.edu

Abstract

In this paper, the challenging problem of the segmentation of Arabic handwriting is addressed. We propose a new efficient technique for segmentation of offline Arabic handwriting into the basic graphemes of the handwritten Arabic script. The proposed technique applies the Douglas-Peucker algorithm on the skeletonized parts of the offline handwritten images to convert them into a set of piecewise linear curves where the local writing direction information and the neighborhood geometric characteristics are used for graphemes segmentation. As a pre-processing stage to the proposed segmentation technique, a reliable rule-based method is used for diacritics extraction that is based on the geometric properties and writing positions of those diacritics. The segmentation technique has been tested using 1400 Arabic handwritten words from the IFN/ENIT database and a comparison made against the recent existing methods in the literature and a promising segmentation performance has been achieved, indicating the effectiveness of the proposed technique.

1. Introduction

Arabic, one of the six United Nations official languages, is the native language for more than 221 million people in the world [1], and over 1 billion people use it in several religion and culture-related activities. The characters of Arabic script and similar characters are used by a much higher percentage of the world's population to write languages such as Arabic, Farsi (Persian), and Urdu. Arabic scripts are inherently cursive for both of its printed and handwritten forms; writing isolated characters in 'block letters' is an unacceptable and unused writing style. The Arabic alphabet contains 28 letters. Each has between two and four shapes, and some characters, especially in Arabic handwriting, may overlap with their neighboring characters forming what is called a "ligature". Arabic script is rich in diacritics that represent short vowels or other sounds and allow differentiating the notion of the letters. In this paper, we use the term "diacritics" even more broadly to also include the dots of the letters.

Offline handwriting recognition is the task of determining what letters or words are present in a digital image of a handwritten text. Recognizing unconstrained offline cursive writing has proven to be a very difficult task, mainly due to the difficulty of character segmentation [2], as the segmentation of words into characters has not yet attained an acceptable performance level [3]. Unfortunately, to the best of our knowledge, we have not seen any algorithm that is able to segment handwritten Arabic words into characters with a high level of accuracy. Building an unlimited lexicon recognition system for handwritten Arabic script requires an efficient segmentation algorithm that segments words into characters or parts of characters, i.e. graphemes, either explicitly or implicitly like the approach of using the hidden Markov model (HMM).

In this paper, we present a new efficient explicit technique for segmentation of offline Arabic handwriting into the basic graphemes of the handwritten Arabic script. The proposed technique applies the Douglas-Peucker algorithm on the skeletonized parts of the offline handwriting images to convert them into a set of piecewise linear curves where the local writing direction can be easily investigated and is used for graphemes segmentation.

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.162

The strategy used in the proposed segmentation technique lies in combining the local writing direction information and some of the neighborhood geometric characteristics in a way that makes use of the nature of Arabic script, which is proven to outperform the recent segmentation techniques in literature.

As a pre-processing stage to the proposed segmentation technique, a reliable rule-based method is used for diacritics extraction that is based on the geometric properties and writing positions of these diacritics.

2. Pre-processing

Firstly, the connected components of the image are extracted (connected components labeling [5]) by splitting the image into a set of connected components (regions of 8-connected black pixels) to be used in the coming analysis on this paper.

2.1. Skeletonization

Skeletonization is a process that reduces the width of a pattern shape to just a single pixel. Generally, for a skeletonization algorithm to be effective, it should ideally compress data and retain the significant features of the pattern. But for the case of handwritten Arabic, it is hard to find a robust and useful skeletonization algorithm that retains the significant features of Arabic writing styles [6], as shown in Figure 1. Thus, we use the skeletonized image during segmentation but the graphemes are extracted from the original image before skeletonization.

Thinning is the process of transforming a pattern from one form to another with less thickness while maintaining the connectivity of the original pattern [7], and this is why skeletonization can be achieved through a thinning process.

In our skeletonization, given the binary test image, a binary morphological template matching operation (hit-and-miss operation) is used to look for particular patterns of foreground and background pixels in the image and replace them with new patterns. These patterns are demonstrated in Figure 2, where the value of “N” has been empirically chosen to be from two to five. Once one of these patterns is found, one of the background pixels of the first row and any columns of “N” is converted to a foreground pixel. This step is essential to force the thinning algorithm not to eliminate some significant pattern shapes of many Arabic letters as shown in Figure 3, which results into improper segmentation afterwards. It should be noted that this pattern replacement process should not connect any unconnected components of the image.

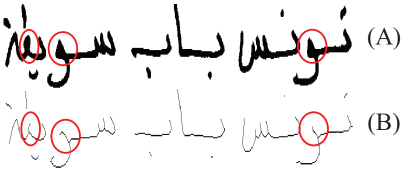


Figure 1. Skeletonization problems with Arabic handwriting. (A) The original image. (B) The Skeletonized image.

And then a thinning algorithm in [7] (page 879, bottom of first column through top of second column) is applied to the binary test image after applying the discussed pattern replacement process. The thinning algorithm is applied iteratively to the image until it converges; i.e., skeleton does not change nor vanish even if the iteration continues.

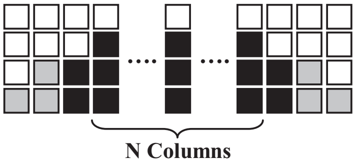


Figure 2. The pattern replacement target pattern, where the gray color pixels are ignored either they are foreground or background.

2.2. Lines Approximation

In this pre-processing stage, the skeletonized shape of each connected component of the test image is converted into a connected series of line segments that approximates the geometric shape of the connected component (a piecewise linear curve). Firstly, a hit-and-miss operation is applied to the skeletonized binary image of each connected component to obtain the end and intersection points. Then a set of paths is constructed by selecting any arbitrary end or intersection point and tracking the connected single pixel path of foreground pixels beginning from that point until meeting another intersection or end points (maybe the same beginning pixel in case of loops), then selecting another end or intersection point and constructing another path, and so on until the set of formed paths covers the entire skeletonized image. Then, the recursive Douglas-Peucker line-simplification algorithm [8] is applied on the sequence of pixel points of each path to obtain the piecewise linear curves of the connected component.

2.3. Diacritics and Noise Segments Extraction

2.4.1. Noise Segments and Dots Detection. The noise segments (the spurious segments that often appear in the binarized image) and the dot didactics are being detected according to the rules described in Table

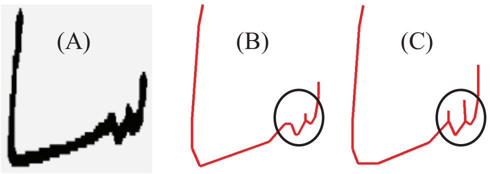


Figure 3. The importance of the pattern replacement step. (A) The original image. (B), (C) The resulting piecewise linear curve before and after the pattern replacement.

1. The threshold values obtained in the table are not optimum, statistics made on handwritten images extracted from samples of the IFN/ENIT database are used to define these thresholds. The conditions of Table 1 are going to be checked for all the connected components of the image after calculating the required features.

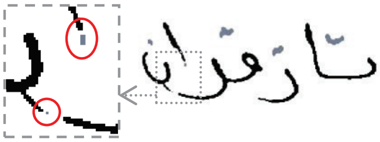
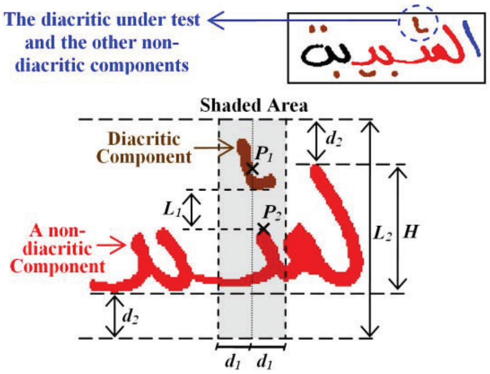


Figure 4. Examples of diacritics and noise segments. Noise segments are circled in red.

2.4.2. Other Diacritics Detection. The detected dots and noise segments are excluded from the estimation of the global average parameters of the test image, like the piecewise linear curves bounding box average area. Although the thresholds' values associated with the number of end and intersection points should have ideal values that are obtained from the standard writing styles of some diacritics (written in parentheses in Table 1), the actual values we used, obtained from the handwriting statistics, are different from those ideal values and give better detection accuracy.

2.4.3. Detection Verification. For all the detected diacritics of step (2.4.2) except for the "Assured Diacritics" category, a verification process that confirms or rejects the detection decision of those diacritics is conducted. The objective of this step is to filter the detected diacritics expect for the ones with clear geometric characteristics (we call "Assured Diacritics") and the dots by making use of the information of their relative writing positions to the connected components that are not detected to be diacritics ("non-diacritic"). Figure 5 shows an example of a valid diacritic component (The values of d_1 and d_2 are set empirically to 15 pixels). A diacritic component is valid if there exists a non-diacritic component such that they make a horizontal historgram overlap higher than 75% (if the diacritic component lie within L$_2$ it's 10% instead) and $L_1 \leq 0.35*Image\_Height$.



The diacritic under test and the other non-diacritic components
Shaded Area
Diacritic Component
P_1
P_2
d_2
L_1
A non-diacritic Component
L_2
H
d_2
d_1 d_1

Figure 5. Diacritics Verification. P$_1$ is the centroid of the diacritic. P$_2$ is the point in the non-diacritics component that is located within that shaded area and makes minimum distance to P$_1$.

3. Graphemes Segmentation

3.1. Horizontal Segments and 'V Shapes'

For horizontal segments detection, as shown in Figure 6, all the lines that make an angle of α_1 degrees with the positive x-axis direction, where: $|\alpha_1|\leq\theta_H$ or $|180^o-\alpha_1|\leq\theta_H$, are obtained (horizontal lines), except for the edge lines (lines having a floating end, i.e., end that is not connected to any other lines and of length less than a predefined threshold T_E). A horizontal segment is constructed from each group of horizontal lines that

TABLE 1
DIACRITICS DETECTION FEATURES

Conditions \ Diacritic Types	Noise Segments	Dot	Two-Dots	'Shaddah'	'Hamza'	Triangle	Other Diacritics	Assured Diacritics
Minimum number of end points	2	-	2 (2)	2 (3)	1 (3)	2 (2)	-	2
Maximum number of end points	-	-	3 (2)	3 (3)	4 (3)	3 (2)	3	3
Maximum number of end and intersection points	2	-	4 (2)	10 (4)	6 (4)	4 (3)	5	10
Maximum ratio of the connected component bounding box area to the average value of all the components	20%	40%	40%	40%	40%	40%	40%	40%
Maximum ratio of the lines-approximated graph bounding box area to the average value of all the graphs	3%	25%	35%	40%	35%	45%	20%	30%
Maximum ratio of the lines-approximated graph sum of lines lengths to the average value of all the graphs	8%	25%	50%	55%	60%	45%	20%	40%
Minimum aspect ratio (Width/Height)	-	-	1	0.8	-	-	-	1.5
Maximum ratio of the height of the component to the image height	-	20%	20%	30%	30%	30%	30%	30%
α; where 2/α ≤ (Width/Height) ≤ α	-	-	-	-	2.5	3	3.5	-
Maximum ratio of the distance between the mean of the connected component and the horizontal histogram peak to the image height	-	10%	10%	10%	10%	10%	10%	20%
Maximum ratio of the distance between the mean of the connected component and the top of the image to the image height	-	-	-	60%	50%	60%	-	40%

are connected in a single shape. For two horizontal lines to join the same horizontal segment, they should have a common end where no other vertical up line is intersecting with them in it. A vertical up line is characterized by making an angle with the positive direction of the x-axis between $90^o-\theta_C$ and $90^o+\theta_C$. Let the centroid of each horizontal segment be called a central point *CP*, as shown in Figure 7.

Then, the shapes constructed from two lines having a common end and look like the 'V' Latin letter ('V shape') are detected. As shown in Figure 6, a 'V shape' is detected according to the slopes of the two intersecting lines, where a valid 'V shape' should satisfy all the following three conditions simultaneously:

1- The intersection point (VP point) has only two lines intersecting in it that are not edge lines.

2- One line of the two lines makes an angle between 0º:90º with the positive x-axis direction, while the other one makes an angle between 90º:180º.

3- The angle between the two lines α_2 is less than θ_V.

3.2. Horizontal Segments Special Shapes

All the detected horizontal segments are checked if they verify one of the two shapes shown in Figure 7. The first shape, 'U shape', is characterized by a horizontal segment that is composed of a single horizontal line and there are two up intersecting lines (not edge lines) connected to both of its left and right ends (P_L and P_R), as shown in Figure 8(A). The second shape is a segment beginning with a vertical down line (a line that lies within the range defined by the angle $90^o-\theta_d$ to $90^o+\theta_d$, as shown in Figure 8(B)).

3.3. Arbitrary Segmentation Points

This step is partially similar to the first step of the online Arabic handwriting segmentation algorithm in [9]. The role of this stage of the algorithm is to eliminate some of the obtained CP and VP points by checking how much each point is covered from above and below, then the remaining group of the accepted CP and VP points is called: the group of arbitrary segmentation points (SP points).

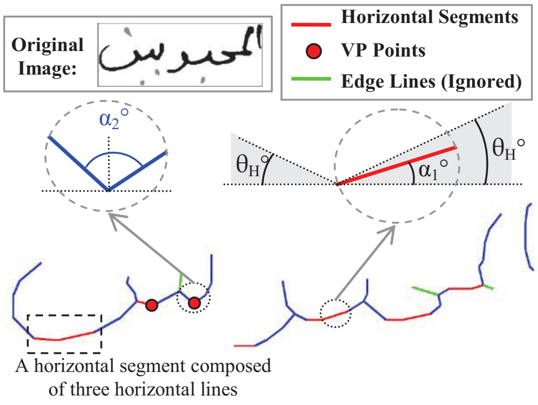


Figure 6. The detection of horizontal segments and 'V shapes.

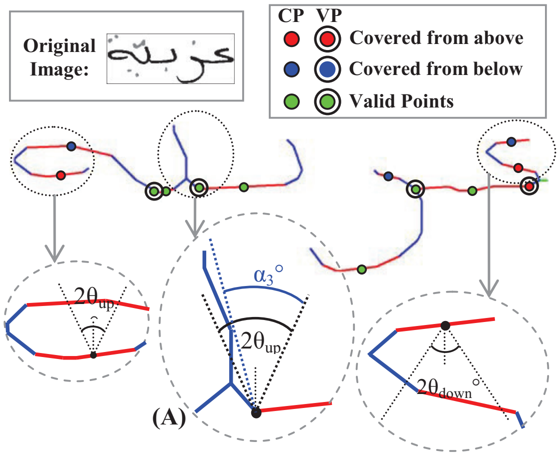


Figure 7. Obtaining the arbitrary segmentation points by eliminating some of the CP and VP points.

For the upper check, the up range defined as shown in Figure 7 ($2\theta_{up}$) is scanned, and it's considered an SP point if there is at least θ_{Tup} range within it that is not covered from above. The same thing is done for the down check with a range of $2\theta_{down}$ and a threshold angular range of θ_{Tdown}. Sometimes the check ranges are less than $2\theta_{up}$ and $2\theta_{down}$ as shown in Figure 7(A), where the up range is equal to α_3 and not $2\theta_{up}$.

It should be noted that all the SP points of the horizontal segments that verify the first special shape, 'U shape', are forced to be accepted SP points and those verifying the second special shape forced to be eliminated from the group of SP points.

3.4. Horizontal Segments Combining

In this step of the algorithm, some of the obtained segmentation points (SP) that correspond to some horizontal segments are combined into a single segmentation points. The SP points of two segmentation junctions are combined only if there exists a single line that connects the two segments with each other as shown in Figure 9, such that: $H_1 \leq T_{H1}$, $H_2 \leq T_{H2}$, $L \leq T_L$, and the connecting line is not connected to any lines other than the two lines it connects from the two segments.

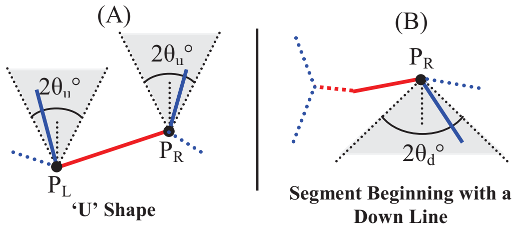


Figure 8. Horizontal segments special shapes characteristics.

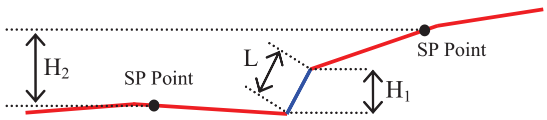


Figure 9. Horizontal Segments Combining.

3.5. Final Segmentation Points

Based on all obtained SP points, we compute a linear regression by determining the parameters of the linear equation: $y = a*x + b$. Then we remove all the least fitting points. This correction step is repeated until the obtained line does not change nor vanish even if the iteration continues.

For all the SP points corresponding to CP points, the product of the segment length and aspect ratio (width/height): $L_s * A_s$, and the distance between the SP point and the line obtained from the linear regression process D_{s1} are calculated. The points that verify the condition: $L_s*A_s/D_{s1}>W_{min1}$ are removed unless the 'U shape' points. And for the SP points corresponding to VP points, the distance between the SP point and the line obtained from the linear regression D_{s2} is calculated, and the points verifying: $1/D_{s2}>W_{min2}$ are removed.

Finally, any SP point corresponding to a 'V shape' that is connected directly with a horizontal segment of another SP point is eliminated as long as there is no up vertical line intersects with them in their intersection point. A test is done on the leftmost SP point of the piecewise linear curve of each connected component of the image to reject or accept it. This step solves the common problem of a false SP point associated with the tail stroke of many Arabic letters like 'ل' in their end and isolated shapes. For this purpose some features are obtained according to that leftmost SP point, including the length and slopes of the tail lines, the maximum height of the tail relative to the maximum height of the whole piecewise linear curve, and the direction of the last line of the tail.

3.6. Graphemes Extraction

Separating the image's graphemes by splitting the image using some vertical line separators defended by the x-coordinates of the SP points is not considered a convenient way for extracting graphemes from Arabic handwriting images for two main reasons:

1- These SP points are not guaranteed to be exactly on the graphemes' edges (as shown in Figure 10(A)).

2- Many Arabic graphemes in Arabic handwriting are not written within the vertical bounds of the SP points that it lies in-between (like the letter 'ـك', and the ligatures).

TABLE 2
THE PREDEFINED THRESHOLD EMPIRICAL VALUES

T_E=25 pixels	θ_H=25°	θ_C=40°	θ_V=160°
θ_u=50°	θ_d=65°	θ_{up}=45°	θ_{down}=60°
θ_{Tup}=3°	θ_{Tdown}=25°	T_{H1}=10 pixels	T_{H2}=15 pixels
T_L= 20 pixels	W_{min1}=3.5	W_{min2}=0.1	L_W=10 pixels

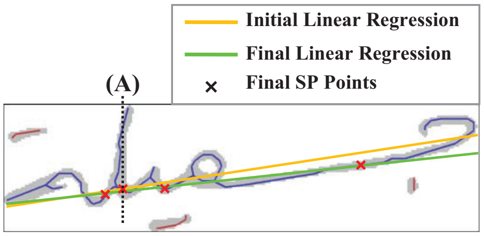


Figure 10. Linear regression. (A) SP points are not guaranteed to be exactly on the graphemes' edges after being projected on the original image.

The obtained final SP points are used to extract the graphemes from the original test image. Firstly, the final SP points are projected on the original test image without any modifications done on its coordinates, as shown in Figure 10. An iterative procedure is used for extracting the graphemes from the original image given the projected SP points of each connected component of the image (as described in the example of Figure 11 for the SP point shown in Figure 10(A)). For each SP point SP_i:

1- A vertical window W of one pixel width centered on SP_i and of height $2L_W+1$ pixels is constructed.

2- Start from SP_i and go upwards in W until meeting the first background pixel or the end of the window W, that pixel is called: up pixel P_U.

3- Similarly, a down pixel P_D is obtained downwards of SP_i (P_M is the pixel in middle between P_U and P_D).

4- IF P_U and P_D are background pixels, THEN the current window W is called the target window T.

5- ELSE: repeat from step 2 on two new windows to the right and the left centered according to the nearest P_M points.

6- The T window pixels between P_U and P_D are converted to background pixels (isolate the graphemes). Then the target pixel near P_M of T is determined and the connected component containing it is extracted to represent the grapheme corresponding to SP_i.

Table 2 shows the empirical values of the parameters and thresholds used in the proposed segmentation algorithm.

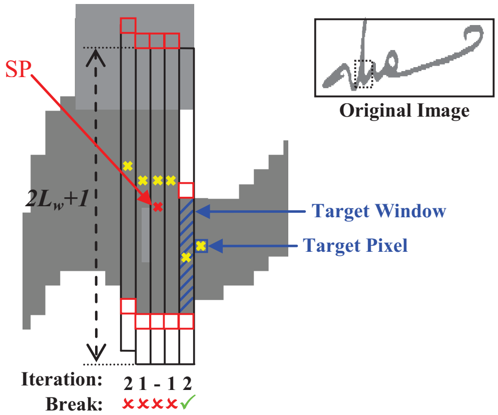


Figure 11. The process of graphemes extraction, where the points P_U and P_D of each window are in red and P_M in yellow.

4. Results

Experiments have been carried out using 1000 images containing 1402 Arabic handwritten words and 7960 Arabic handwritten graphemes taken from the IFN/ENIT database [10]. The choice of the words has been carefully selected to cover all shapes of the Arabic graphemes. For each image the number of graphemes and the number of the fully correct segmented graphemes are manually evaluated by observing the result of the algorithm and cross checked by two independent Arabic speaking observers.

The results obtained show that 91.27% of the graphemes are correctly segmented. In Table 3, a comparison is made against the recently existing methods in the literature and it is clear from the table that a promising segmentation performance has been obtained indicating the effectiveness of the proposed technique.

TABLE 3
Comparison of Our Results with Previous Works

Authors	Method	#Words	Accuracy
Lawgali et al. [11]	Extracting baseline	800	87.900%
Al-Hamad et al. [12]	Over-segmentation & ANN	500	82.980%
Sari et al. [13]	Morphological Analysis	100	86.000%
Proposed Technique	Local Writing Direction	1402	**91.274%**

Table 3 confirms that the strategy used in the proposed segmentation technique which lies in combining the local writing direction information and the neighborhood geometric characteristics in a way that makes use of the nature of Arabic script is a convenient and efficient technique for segmentation of Arabic handwriting. Recognition errors analysis shows that the segmentation errors caused from the wrongly connected letters represent more than 50% of the total graphemes segmentation errors. Figure 12 shows some results applied on handwritten Arabic words.

5. Conclusion

In this paper, novel diacritics extraction and graphemes segmentation techniques are proposed. The strategy used in the proposed segmentation technique lies in combining the local writing direction information and the neighborhood geometric characteristics in a way that makes use of the nature of Arabic script, which has proven to achieve a promising segmentation performance that outperforms the most recent segmentation techniques in the literature.

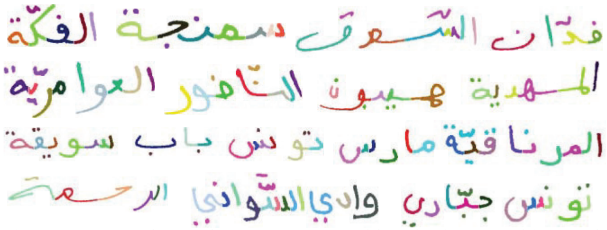


Figure 12. Sample results of a segmented handwritten Arabic words Diacritics and noise segments are in the same color in each image.

References

[1] M. Paul Lewis, *Ethnologue: Languages of the World*, 16th ed. SIL Int'l, 2009.

[2] Shaikh, N., Mallah, G.A., Shaikh, Z., "Character Segmentation of Sindhi, and Arabic Style Scripting Language using Height Profile Vector", *Australian Journal of Basic and Applied Sciences*, pp. 4160-4169, ISSN. 1991-8178, Oct-Dec 2009.

[3] Nafiz Arica and Fatos T. Yarman-Vural, "An Overview of Character Recognition Focused on Off-Line Handwriting," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 31, no. 2, pp. 216-233, May 2001.

[4] Khorsheed, M.S., "Offline Arabic character recognition -a review," *Pattern Anal. Appl*., 5, 31–45, 2002.

[5] Yebin Fan, Shengsheng Yu, Hualong Zhao, "A novel line based connected component labeling algorithm," *IEEE Inter. Conf. on Computer Science and Info. Tech. (ICCSIT)*, vol. 2, pp. 168-172, 9-11 July 2010.

[6] Wshah, S., Zhixin Shi, Govindaraju, V., "Segmentation of Arabic Handwriting Based on both Contour and Skeleton Segmentation," *10th International Conf. on Document Analysis and Recogn. (ICDAR '09)*, 2009. pp. 793-797, 26-29 July 2009.

[7] Lam, L., Seong-Whan Lee, and Ching Y. Suen, "Thinning Methodologies-A Comprehensive Survey," *IEEE Transactions on Pattern Anal. and Machine Intelligence*, vol. 14(9), 1992.

[8] D.H. Douglas, T.K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the International Journal for Geographic Information and Geovisualization*, vol. 10, pp. 112–122, 1973.

[9] H. M. Eraqi and S. Abdelazeem, "An On-Line Arabic Handwriting Recognition System Based on a new On-line Graphemes Segmentation Technique," *In proc. of the 11th Inter. Conf. on Document Analysis and Recogn.* (ICDAR '11), pp. 409-413, 2011.

[10] M. Pechwitz, S. S. Maddouri, V. Mrgner, N. Ellouze, and H. Amiri, "IFN/ENIT-database of handwritten Arabic words," in proc. of CIFED, pages 129–136, 2002.

[11] A. Lawgali, A. Bouridane, M. Angelova, and Z. Ghassemlooy, "Automatic segmentation for Arabic characters in handwriting documents," 18 IEEE Intern.Conf. on Image Processing, 2011.

[12] H. A. Al-Hamad and R. Abu Zitar, "Development of an efficient neural-based segmentation technique for Arabic handwriting recognition," Pattern Recognition, 43(8):2773–2798, 2010.

[13] T. Sari, L. Souici, and M. Sellami, "Off-line handwritten Arabic character segmentation algorithm: Acsa," in proc. 8th Inter. Workshop on Frontiers in Handwriting Recogn. , 2002.