Off-line Handwritten Arabic Words Segmentation Based on Structural Features and Connected Components Analysis

Moftah Elzobi, Ayoub Al-Hamadi, Zaher Al Aghbari*
Institute for Electronics, Signal Processing and Communications
Otto-von-Guericke-University Magdeburg, Germany
**Department of Computer Science, University of Sharjah, UAE*
{*Moftah.Elzobi, Ayoub.Al-Hamadi*}*@ovgu.de*

***Abstract*— A precise and efficient segmentation for handwritten Arabic text is a vital prerequisite for the accuracy of the subsequent recognition phase. In this paper, we present a dual-phase segmentation approach. The proposed approach starts first by detecting and resolving sub-words overlapping, then a topological features based segmentation is applied by means of a set of heuristic rules. Because of its crucial importance, the segmentation phase is preceded by a handwritten specific pre-processing phase, that considers issues like word's skew- and slant- correction. The proposed approach has been successfully tested on a database of handwritten Arabic words, that contains more than 3000 words images. The results were very promising and indicating the efficiency of our approach.**

***Keywords*- Arabic Handwriting Segmentation, Handwriting Topological Features, Pattern Recognition.**

I. INTRODUCTION

PEOPLE nowadays expecting that, modern as well as historical human knowledge and cultural resources are digitally available as electronic text, which can be fast, efficiently and easily accessed. Resources that are not converted properly to digital text, e.g., Unicode, ASCII, and etc., soon will become obsolete or even inaccessible for researchers, scholars and general public. This means lose of an important and huge amount of the human cultural memory.

Off-line Optical Character Recognition (OCR) is the technological means used for converting handwritten, typewritten and printed text into a digital text. Latin alphabet based and Chinese characters based scripts have been the subject of extensive research since decades, that lead to significant achievements in the field [1], [2]. Despite the fact of being the world second most used alphabet, reported works that address the off-line OCR issues related to Arabic alphabet based scripts, e.g., Arabic, Persian, Urdu, Ottoman, Kurd, and etc., are relatively less in quantity and quality.

The complexities that hindering fast progress in this research field can be related to the cursive written form of Arabic script and to the high variability of character's forms with regard to their positions (isolated, begin, middle and end) within a word, that makes an efficient segmentation hard and error-prone process. It is also observed that techniques proved successful for Latin and/or Chinese, cannot be directly applied without fundamentals changes [1], [3], [4], [5], [6].

In this paper, we introduce a topological feature based segmentation approach for handwritten Arabic words. In order to compensate limitations of extraction steps such as binarization, our methodology starts by pre-processing steps such as small holes filling and smoothing the out-cropping pixels. Then to reduce the extreme variability of handwritten words, normalization issues such as Slant- and Skew- correction is considered. The segmentation then conducted through a dual-phase procedure. In the first phase, a connected component analysis is performed, in order to resolve sub-words overlapping. Then topological features based segmentation is carried out to segment the word into identifiable units representing their constituent characters.

In literature relatively few works are addressing the problem of Arabic text segmentation. Hereafter we will briefly discuss the most important published related works. In [3] a recognition system for off-line cursive handwriting is presented. The system is segmentation based one, thinned and smoothed images of the strokes are processed and two representations for each stroke are generated. The first representation is a direct straight-line approximation. The other is what they called a reduced graph with loops reduced to vertices, then temporal information of the strokes is extracted by following their straight-line representation from right to left. Finally, cursive stroke representatives are segmented to small parts called tokens that passed to the recognizer.

In an attempt to avoid over-segmentation, [7] propose an analytical segmentation approach, that is trying to extract the whole stroke that represents the character by means of the so called Character Key Feature Set (KF), which is the set of End-Points, Branch-Points, Loop-Points and Dot-Points from the word thinned image. First, the minima and maxima

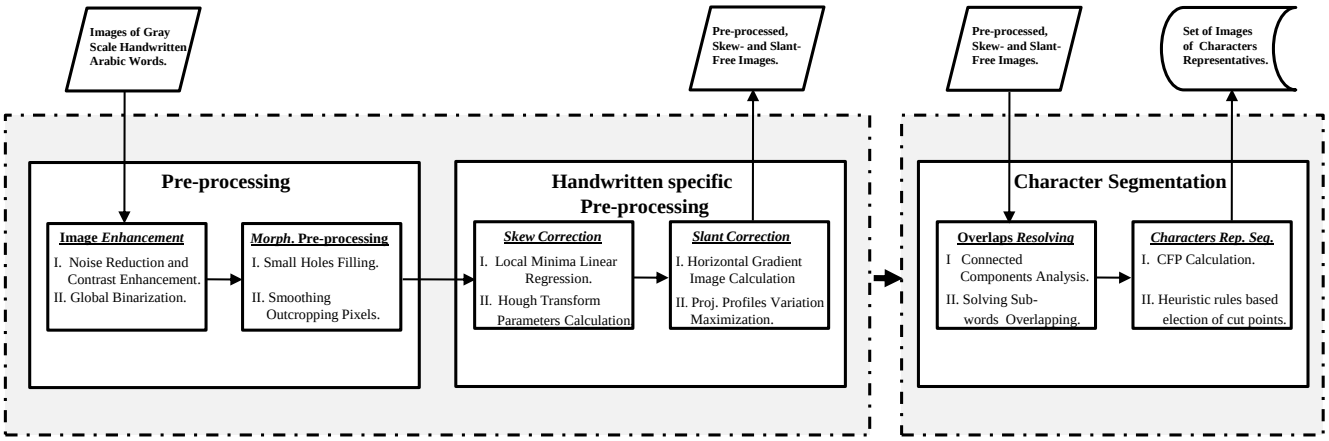


Figure 1. Diagram of the proposed Methodology

of the thinned image are calculated, then the so called Key Features Segments (KFSg) are determined. Secondly, A set of heuristic rules that employ KF set, are applied on the set of the minima in order to elect cut candidates among them.

Ref. [8], proposed an approach, in which a tentative over-segmentation is first performed on the text image, the result is a set of what they called "graphemes", the approach differentiates among three types of graphemes. The segmentation decisions are confirmed upon the recognition results of the merged neighboring graphemes; if recognition failed another merge will be tried until successful recognition.

Ref. [4], presenting an algorithm for printed text segmentation, in which the vertical projection histogram of each line of the source binary image is computed, which then processed to generate a string indicating relative variations in pixels. Finally, a search for patterns in variations is conducted in order to segment the characters' representatives.

II. Methodology

Our methodology consists mainly of two phases, in the first phase the issue of pre-processing is considered. In which traditional pre-processing steps e.g. filtering, binarization, etc., as well as Handwriting specific issues like skew correction and slant correction are conducted. The second phase is the segmentation, which performed in dual-phase procedure.

Given the fact that most Arabic words are consisting of multiple sub-words [9], those are both specially disconnected and vertically overlapping, connected components based analysis and subsequently resolving of sub-words overlapping, are vital for the following character segmentation phase. Ultimately, a topological features based heuristics are applied in order to segment the words into their constituent character representatives. Fig.1 is depicting the proposed methodology.

A. Pre-processing

The words images that we experimented on are gray scale images, taken from an under construction database; conventional flatbed scanner is used to extract the text with 350 dpi resolution. To suppress noisy pixels, whilst preserving edges a median filter is applied on the gray scale images [10]. Then a global threshold (Otsu's method based) is used to produce binary versions. As a consequence of the extraction and binarization processes, issues like smoothing out outcropping pixels and small holes filling should be dealt with. Morphological based operations such as $Close$ and combination of $Open$ and $Reconstruction$ are employed respectively to solve those issues.

To reduce the amount of information to be processed, to the minimum necessary for conducting our segmentation, and also to ease the process of extraction the critical features points, thinning operation is applied on the enhanced binary words' images. The thinning approach that we adapted is based on the Zhang-Suen's thinning algorithm [11]. In the following two subsections, we will discuss and suggests improvements for two off-line handwritten specific pre-processing issues, namely skew- and slant- correction.

1) Skew correction and baseline estimation: Skew correction and baseline discovering is proven to be of critical importance for segmentation of handwritten Arabic text. Various techniques have been reported in literature; each with its pros and cons [12], [13]. Hough transform (HT) is one of such methods, that is relatively insensitive to noise and tolerates gabs within Arabic words [14]. As for the baseline detection HT is insensitive to line direction. Consequently, it performs badly when the longest stroke is not parallel to the word baseline.

Another method is based on the linear regression of local minima of the word image skeleton (LMR) [15]. Benefiting from the fact, that most of local minima (LM) points are usually occurring on, or near of the baseline; the problem of

finding the baseline can be reduced to a linear fitting problem of local minima points. Even though LMR is not as accurate as HT, its main advantage is the relatively insensitivity to strokes' direction that is not parallel to the baseline.

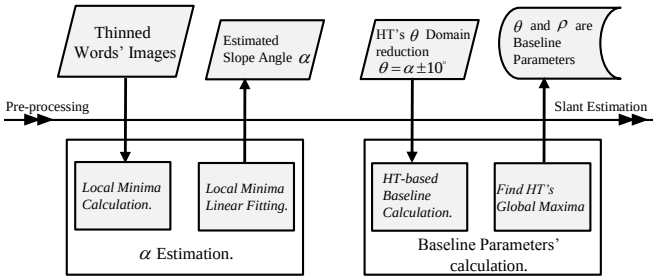


Figure 2. Diagram illustrates the proposed steps for words baseline estimation

Experimentally, we noticed that reducing the Domain of HT's θ parameter according to a priori direction estimation, firstly, increases accuracy, and secondly, reduces the computation power needed. Thus we propose a HT based technique combined with a LMR, for baseline estimation. The LMR is used for a priori estimation of the HT's θ parameter. Fig. 2, shows a diagram depicting the propose approach for baseline estimation.

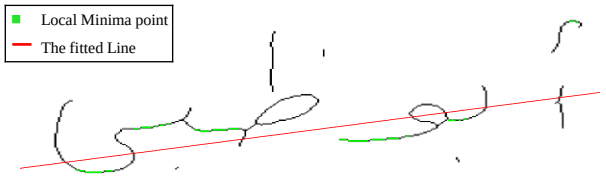


Figure 3. A thinned text image with all possible LM points and their correspondence fitting line.

The first step in the proposed technique starts by calculating the fitting line of LM points according to Eq.1,2, and 3.

$$y = a + bx \tag{1}$$

where a, b coefficients calculated as follow

$$b = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{2}$$

$$a = \bar{y} - b\bar{x} \tag{3}$$

where $\bar{x}$ and $\bar{y}$ are the statistical means of x and y coordinates respectively.

Fig. 3 shows a thinned image with LM points and their correspondence fitted line. The slope angle α of the fitted line is then calculated according to Eq.4

$$\alpha = \arctan(b) \tag{4}$$

The second step is to calculate the baseline using the HT, with θ's domain reduced to be $[\alpha \pm 10^{\circ}]$,where $[\pm 10^{\circ}]$ is the empirically observed inaccuracy of LM. We first discretize the θ and ρ parameters and then for each point (x_i, y_i) in the image space we calculate $\acute{\rho}$ as stated in Eq.5:

$$\acute{\rho} = x_i \sin\acute{\theta} + y_i \cos\acute{\theta} \quad \forall\acute{\theta} \in [\alpha - 10^{\circ}, \alpha + 10^{\circ}] \tag{5}$$

Next, each point in the image space will vote for bins that could have generated it in the hough accumulator A, and votes will be accumulated in A according to Eq.6

$$A(\acute{\rho}, \acute{\theta}) = A(\acute{\rho}, \acute{\theta}) + 1 \tag{6}$$

Finally, $\acute{\rho}$ and $\acute{\theta}$ with the maximum number (global maxima) of votes will be considered as the parameters of the word baseline as showed in Eq.7.

$$\arg\max_{\acute{\rho},\acute{\theta}} A(\acute{\rho}, \acute{\theta}) \tag{7}$$

Fig. 4, shows an example of the results, where Fig. 4(A) is the original image, Fig. 4(B) is the skew corrected image and the estimated baseline according to LMR only. Fig. 4(C) shows the result of the skew correction and the estimated baseline of the word using the proposed technique. The reader can clearly see the improvement.

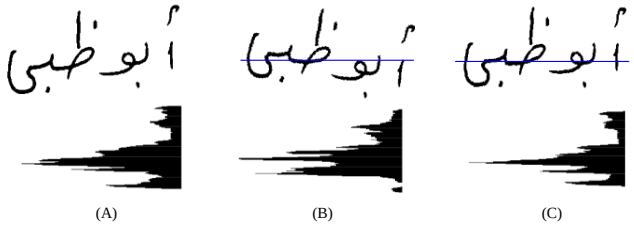
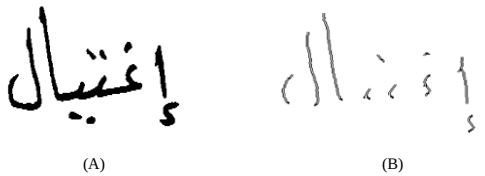


Figure 4. Skew correction and baseline estimation, (A) Original binary image and its corresponding horizontal projection profile, (B) Corrected version using LMR only, (C) Corrected version using the proposed technique.

2) Slant Correction: Slant angle is the angle, which vertical strokes make with the absolute vertical direction. In order to reduce variability within handwritten characters' classes, it is necessary to normalize slant variations [16]. As for the segmentation process, slant correction improves accuracy, since spaces between vertical strokes will be increased.

Since it is being observed, that vertical projection histograms of the slant free images have higher and clear peaks compared to the slanted ones. Thus projection profile based technique calculated upon the horizontal gradient image at various shearing angles in the range $[\pm 45]$, is used for estimation of the slant angle. The reasons behind choosing the horizontal gradient image for calculation are, first, vertical strokes will be emphasized at the expense of horizontal ones, as Fig. 5(B) shows. Second, computation cost will be reduced, since relatively fewer pixels need to be processed.



(A) (B)

Figure 5. Horizontal gradient, (A) The binary word image, (B) The horizontal gradient counterpart.

Given (x, y) the coordinates of pixel in an image i, their sheared counterparts $(\acute{x}, \acute{y})$ in the sheared image $\acute{i}$ are calculated according to Eq.8

$$\acute{x} = x - y.\tan(\alpha), \quad \acute{y} = y \tag{8}$$

where $\alpha \in [\pm 45]$ is the shearing angle.

For each sheared image we calculate vertical histogram H as stated in Eq.9.

$$H(\acute{x}_L; \alpha) = \sum_{k=0}^{\infty} \acute{i}(\acute{x}_L, \acute{y}_k) \tag{9}$$

Then the variation for every two consecutive profiles is calculated as in Eq.10.

$$V(\alpha) = \sum_{L=0}^{\infty} [H(\acute{x}_L; \alpha) - H(\acute{x}_{L+1}; \alpha)]^2 \tag{10}$$

And the sheared angle $\acute{\alpha}$ is calculated as the angle associated with maximum variation, according to Eq.11.

$$\acute{\alpha} = \arg\max_{\alpha} V(\alpha) \tag{11}$$

Fig. 6(A) shows a slanted word image and its corresponding vertical histogram, and Fig. 6(B) shows its slanted free version and its vertical histogram.

(A) (B)

Figure 6. Slant correction, (A) A binary word image and its correspondence vertical projection profile, (B) The Slant corrected version and its correspondence vertical projection profile.

B. Characters Segmentation

As mentioned above our segmentation approach conducted in two steps. In the first step, a careful analysis of the x-axis coordinates of the connected components is performed. The result is words images with resolved subword overlapping. The second step is to perform topological feature based segmentation for characters' representatives.

Before detailing our approach, it is helpful to start by recalling some definitions that are thought to be necessary for the clarity of subsequent definitions and notations.

Firstly, let P refers to any foreground pixel in the thinned word image $g(x, y)$, and let $N_8(P)$ denotes the 8-neighborhood set of P. Secondly, by examining each $P \in g(x, y)$ a set of feature points are identified, which we call **C**ritical **F**eature **P**oint (CFP). CFP set contains further four subset that are listed below:

i. The first subset is **E**nd **P**oints (EP) depicted in Blue in Fig. 7, which are all pixels with only one pixel in its 8-neighborhood set.
$$EP = \{P | N_8(P) = 1\} \tag{12}$$

ii. The second subset is Branch Points (BP) depicted in bright Green in Fig. 7, which are all pixels where its 8-neighborhood set contains only 3 or 4 pixels.
$$BP = \{P | N_8(P) = 3 \vee P | N_8(P) = 4\} \tag{13}$$

iii. The third subset is Dot Points (DP), which is the union of the set of all isolated pixels, and the set of pixels that belong to connected components (CC) that are less in size than an adaptive threshold T proportional to the estimated character size calculated upon the thinned text image. Depicted in Cyan in Fig. 7
$$\begin{aligned} DP = &\{P | N_8(P) = 0\} \\ &\cup \{P | P \in CC \wedge size(CC) < T\} \end{aligned} \tag{14}$$
where $T \leq \mu$, and μ is the mean of the area of all CC, that are not intersecting the baseline.

iv. The fourth and last subset is the Loop Points (LP), which are all On remained pixels of the thinned text image after performing the $flood-fill$ (ff) algorithm on it. depicted in Red in Fig. 7
$$LP = \{P | P \in ff(i(x, y))\} \tag{15}$$

Given the aforementioned four subsets the CFP set can be defined as the union of all the four subsets. Fig. 7 shows a thinned text image with all possible CFPs, that will be utilized later to guide the characters' segmentation process.

$$CFP = \cup\{LP, EP, BP, DP\} \tag{16}$$

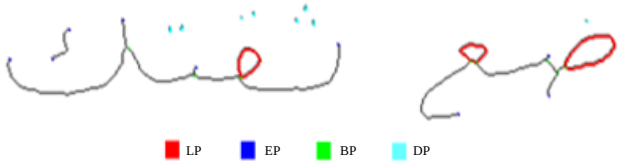


Figure 7. CFP Feature Points, thinned text image with all possible CFPs.

Next, we present the first step of the proposed segmentation approach.

1) Resolving of Sub-words' Overlapping: For resolving the sub-words overlapping, we first find the word baseline as stated above. Then upon finding the baseline, we differentiate between two types of connected components (CC). The first is what we call main (CC)s, which are all (CC)s that intersecting with the baseline's "y" coordinate. The second are what we call auxiliary (CC)s, which are all CCs that are not intersecting the baseline's "y" coordinate. Fig. 8 shows an example of word images, where the main CCs are (1,2,4, and 6), and the auxiliaries are (3,5,7, and 8) and the horizontal blue line representing the baseline.

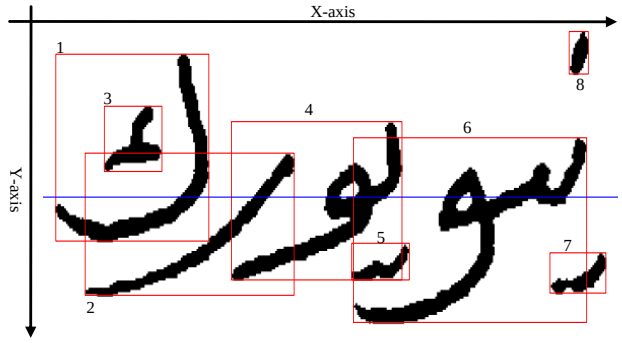


Figure 8. Sub-words overlapping example.

After identifying the main CCs, we conduct a distance analysis on their bounding boxes along the x-axis, in order to identify the baseline overlapped main CCs and their correspondence overlapping distances. In Fig. 8 for example, main CCs that are overlapping are (1,2), (2,4), and (4,6). Another distance analysis is performed against the auxiliary (CC)s, so each can be assigned to its correspondence main CC according to the following rules:

i. If an auxiliary CC is overlapping only a given main CC along the x-axis, then assign the auxiliary to the main. So in Fig. 8 for example, auxiliaries number (8, 7) will be assigned to the main CC number "6".
ii. If an auxiliary that is above the baseline, is completely contained in the bounding box of a main CC, then assign the auxiliary to the main regardless of any main CC that may overlap it along the x-axis. So "3" will be assigned to "1" in Fig. 8.
iii. if two main CC are overlapping an auxiliary under baseline, like in case of "5" that is overlapped both "4" and "6", we calculate the absolute distance along the y-axis, between lower bounding box of the auxiliary, and the lower bounding box of the overlapping main CCs; the one with minimal distance wins the auxiliary. So "4" wins "5" in Fig. 8 for example.
iv. In case auxiliary is above the baseline and overlapping multiple main CCs, the absolute distance along the y-axis is measured between its lower bounding box "y" coordinate and the upper "y" coordinates of the overlapping main CCs bounding box; the main CC with the minimum distance wins.

Even though the aforementioned rules solve for almost all the cases, there are some extreme cases where auxiliaries are not overlapping any main CC. In these cases, auxiliary is assigned to the direct next main CC on the left [1]. After assigning the auxiliaries to their corresponding main CCs, we computed the sub-words borders along the x-axis against all its elements (auxiliary CCs and main CCs). The sub-word's bounding box left border, is computed to be the farthest left border among all sub-word elements. Likewise, the right border is selected to be the farthest border to the right.

Eventually, a final distance analysis is preformed against the new sub-word borders and the overlapping solved by shifting away the overlapped sub-words. Fig. 9 shows two examples (A) and (B) and their correspondence sub-word overlapping free version. As a result of this pre-segmentation step, sub-words are separated by empty columns that make their segmentation a straight forward process.

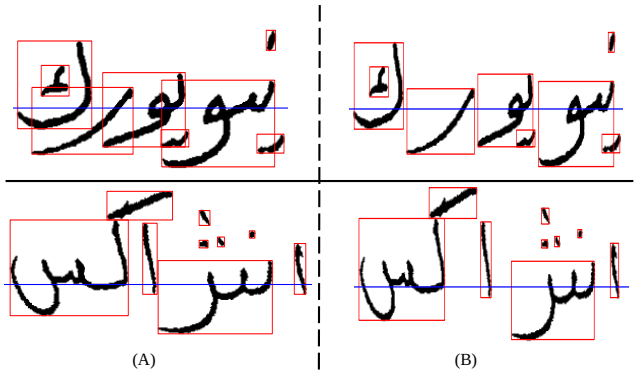


Figure 9. Resolving sub-word's overlapping, (A) Overlapped sub-words versions, (B) Results of overlap resolving procedure.

2) Segmentation of Character Representative : Providing sub-word overlapping is solved, we turn to the issue of segmentation of sub-words into their constituent characters' representatives. Given that Arabic characters have their boundaries in columns with the minimum number of pixels (only one pixel in the thinned version), our segmentation approach starts by generating a set C of columns' indices as candidates for segmentation, where the elements of C are all column indices within the thinned image $g(x, y)$, containing only one foreground pixel. We developed the segmentation algorithm presented in [7], in a way that we use broader set of segmentation candidates instead of using

[1]This is due to the fact that Arabic text is written from right to left, and writers usually writing main CC first, then auxiliaries. As a result auxiliaries are appearing shifted to the right away from their correspondence sub-words.

the set of only the contour's local minima. We observed that using local minima as candidates for segmentation is risking good segmentation, because local minima often occur inside many of Arabic characters main strokes. So we decided to generate a large set of segmentation candidates, and we did not restrict our candidates to be only local minima.

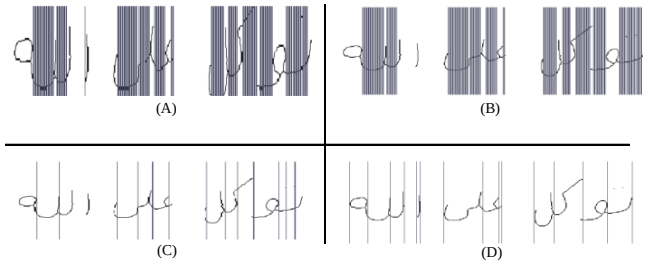


Figure 10. Cut candidates election, (A) Depicts all possible cut candidates. (B) Candidates after excluding column that contains EP, LP, BP or DP. (C) Candidates after excluding candidates with direct left neighbor. (D) The result after applying the proposed heuristic rules

Fig. 10(A) is depicting the columns' candidates for segmentation of the text image, the reader can notice that each column containing only one pixel is elected as a candidate. The next step is to exclude from the candidates set all columns that are intersecting with any CFP, this is to say that LP, EP, BP and DP columns cannot be in the same time a cut point, otherwise we lose important character features. Fig. 10(B) shows a text image after excluding columns' candidates that are intersecting CFP columns. The reader can notice in Fig. 10(B) that very few candidates are excluded comparing to Fig. 10(A), this is because, it is quit seldom that an LP or a BP column contains only one pixel, so it will not be chosen as a candidate in the first place.

The next step is to scan the list of candidates, starting from the most right one to the left, electing the left neighbor from each two adjacent segmentation candidates. Fig. 10(C), shows the result, we can easily notice the significant reduction in the number of candidates for segmentation. The candidates set obtained so far is an important improvement over the previous candidate sets. However, in order to resolve issues like, for example the over-segmentation in Fig. 10(C) (*the letter ت (TAA), the first letter from the right is over-segmented into three parts*). We formulate four conditions to increase segmentation accuracy.

To ease notation of conditions, we will write m a subscript to CFP or/and $CFPs$ elements, to refer to the respective column index. Also, we will use c_i , c_j to refer to any two column indices in the thinned image $g(x, y)$, that are chosen to be candidates. The finale election process is performed by applying the following condition on the candidates:

i. First condition, is saying that if there are two consecutive cut candidates and there is no CFP in between then delete from the list the one on the right, this condition can be formulated as following;

$$\forall\{c_i, c_j\} \in \{C\}|c_i > c_j, \\ if\ \{CFP_m\} \notin [c_i, c_j] \Rightarrow c_i \notin C \tag{17}$$

ii. Second condition, if the direct neighboring on the left is a column contain pixel of DP, then delete the candidate from the list. This can be notated as following:

$$\forall c_i \in C\ if\ \exists(c_i + 1) \in DP_m \Rightarrow c_i \notin C \tag{18}$$

where DP_m, is the set of columns contain DP pixels.

iii. Third condition saying that if there is a branch point column BP or Loop point column LP before encountering another candidate then we elect the candidate as cut point, the notation version of the condition is:

$$\forall\{c_i, c_j\} \in \{C\}|c_i > c_j, \\ if\ \exists(BP_m \vee LP_m) \in [c_j, c_i] \Rightarrow c_i \in C \tag{19}$$

iv. If the next column contains an end point EP_1, which is in the same time not an end of stroke, then flow the contour starting from EP_1 down to the left, if another end point EP_2 is encountered (before BP or LP) which, not on the contour and is an end of the stroke, then elect the candidate as a cut point.

Finally, we insert a cut candidate direct before and after every connected component.

Fig. 10(D) illustrates the final segmentation results, and Fig. 11, shows zoomed in segmentation's result of an Arabic sentence.

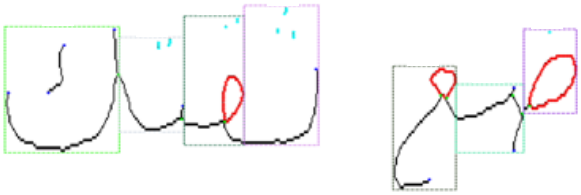


Figure 11. Characters segmentation, characters' representatives are bounded by rectangles.

III. Experimental Results

We experimented our proposed methodology on an under construction database of handwritten Arabic words, that contains more than 3000 Arabic words images, collected from more than 30 persons. Our results are very satisfactory, and to our knowledge outperforming literature available results so far. We have tested a system implemented according to the proposed approach on a set of 200 different words' images. Fig. 12, illustrates some of the results, where complete success is reported in 72% of cases, this means the system accurately discovers the character representatives' borders. Fig. 12(A), Fig. 12(B), Fig. 12(C), Fig. 12(D) illustrates examples of 100% success segmentation of character representatives, where each is bounded in a rectangle. Partial

success is reported in 28% of cases, we have empirically noticed that 9% of such cases are generated when CFPs occur inside the character instead of on its borders, leading to what we call an over segmentation, where a part of stroke is regarded as a character representative, which, in fact, it is not. This problem is specific to characters س and ش (SIEN and SHIEN), and Fig. 12(A) illustrates an example, the black arrow is pointing out to where it occurs. The other 19% of cases happen when CFPs cease to exist between two consecutive characters leading to segment them as a representative for one character. Fig. 12(F), shows such problem, where its position indicated with the arrow. This problem is called under-segmentation, and it is specific for cases, when the second character to left is connected ا (ALF) or connected ل (LAM) with sheared distortion angle to the left. We think that those problems can be solved, either by expanding the CFP set to contain more features points like Local minima points for example and then accordingly modify and adding heuristic rules, or they can be solved in subsequent recognition phases like in the post-processing phase for instance, where the recognition results can be corrected against lexicons using different text retrieval techniques.

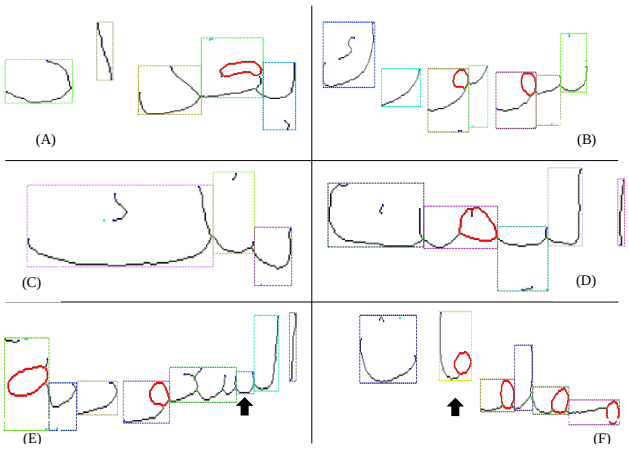


Figure 12. Characters segmentation, (A)-(D) Successful segmentation of characters representatives, (E)-(F) Partial success of segmentation.

IV. Conclusion and Future Works

In this paper, we proposed dual-phase segmentation approach that starts first by sub-word borders' identification and resolving overlapping among them, and then topological features based segmentation is taking place according to a set of heuristic rules. The dual-phase segmentation is preceded by an extensive handwriting specific pre-processing phase in which issues like morphological enhancement, skew correction, and slant correction are dealt with. Even though results were quite satisfactory, future works may investigate issues like, expanding CFP set by adding more topological features and correspondingly more heuristics. Moreover, a cyclic segmentation-recognition based approach is expected to improve results further. In such approach character representatives segmentation proved against the results of the recognition phase. As an in-between approach, of segmentation based approaches with their relatively high error-prone tendency and holistic based approach with their small and restricted domains [17]. A holistic sub-words based approach can be another alternative to attack the problem of off-line handwritten Arabic text recognition.

References

[1] L. M. Lorigo and V. Govindaraju. "Offline Arabic handwriting recognition", *Pattern Analysis and Machine Intelligence, IEEE Transaction on*, Vol. 28, No. 5, pp.712724, May 2006.

[2] M. Elzobi, A. Al-Hamadi, L. Dinges, B. Michaelis. "A Structural Features Based Segmentation for Off-line Handwritten Arabic Text", International Symposium on Image/Video Communications over fixed and mobile networks (ISIVC2010). Rabat, Morocco, Sep.30-Oct.02, 2010.

[3] I. Abuhaiba, M. Holt and S. Datta. "Recognition of Off-Line Cursive Handwriting", Computer Vision and Image Understanding, Vol. 71, No. 1, pp. 19-38, July 1998.

[4] N. A. Shaikh, Z. A. Shaikh, and G. Ali. " Segmentation of Arabic Text into Character for Recognition", IMTIC 2008, CCIS 20, Springer Berlin/Heidelberg, pp.11-18, 2008.

[5] H. Almuallim, S. Yamaguchi. " A Method of Recognition of Arabic Cursive Handwriting", *Pattern Analysis and Machine Intelligence, IEEE Transaction on*, Vol. 9, No. 5, pp.715-722, Sep. 1987.

[6] Z. Al Aghbari, S. Brook. "HAH manuscripts: Aholistic paradigm for classifying and retrieving historical Arabic handwritten documents", Journal of Expert Systems with Applications , Vol. 36, Nr. 8, (2009), pp. 1094210951.

[7] A. A. Atici and F. T. Yarman. "A Heuristic Algorithm for Optical Character Recognition of Arabic Script", Signal Processing., Vol. 62, No. 1, pp.87-99, Oct. 1997.

[8] X. Ding, and H. Liu. "Segmentation-Driven Offline Handwritten Chinese and Arabic Script Recognition", Springer Berlin/Heidelberg, LNCS 4768, pp.196-217, 2008.

[9] D. Motawa, A. Amin, and R. Sabourin. " Segmentation of Arabic Cursive Script", ICDAR, pp.625-628, 1997.

[10] E. Arias-Castro and D.L. Donoho. "Does median filtering truly preserve edges better than linear filtering?", Annals of Statistics, Vol. 37, No. 3, pp.11721206, 2009.

[11] L. Lam, S. Lee, and C. Suen. "Thinning Methodologies - A Comprehensive Survey", *Pattern Analysis and Machine Intelligence, IEEE Transaction on*, Vol. 14, No. 9, pp.869-885, Sep. 1992.

[12] Z. Razak et al. "Off-line Handwriting Text Line Segmentation: A Review", IJCSNS, Vol. 8, No.7, Jul. 2008.

[13] V. Beusekom, F. Shafait and T. M. Breuel. "Combined orientation and skew detection using geometric text-line modeling", IJDAR, Vol.13, No.2, pp.79-92, Springer-Verlag Berlin, Heidelberg, Jun. 2010.

[14] T. M. HA and H. Bunke. "Image processing methods for document image analysis," *in Handbook of Charcter Recognition and Document Image Analysis*, Singapore:World Scientific Publishing Co. Pte. Ltd.,1997, pp.1-47.

[15] M. Wienecke. ”Videobasierte handschrifterkennung”, Ph.D. dissertation, Bielefeld university, Bielefeld, Germany, 2003.

[16] N. Arica and F. T. Yarman-Vural.”An Overview of Character Recognition Focused on Off-Line Handwriting”, *Systems, Man and CyberneticsPart C: Applications and reviews, IEEE Transaction on*,Vol. 31, No. 2, pp.216-233, May 2001.

[17] V. Lavrenko, T. M. Rath and R. Manmatha. ”Holistic Word Recognition for Handwritten Historical Documents”, DIAL04, pp.278-287, 2004.