IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 4, Ver. IV (July – Aug. 2015), PP 23-30
www.iosrjournals.org

A New Approach and Algorithm for Baseline Detection of Arabic Handwriting

Abdullah Baz[1], Mohammed Baz[2]
[1](Computer Engineering Department, College of Computer & Information Systems, Umm Al-Qura University, Saudi Arabia)
[2](Computer Engineering Department, College of Computer & Information Technology, Taif University, Saudi Arabia)

Abstract : *Automatic baseline detection of handwritten Arabic words is a crucial task for OCR. It is extensively used in many preprocessing processes such as text normalization, skew/slant correction, and letters segmentation. Due to its importance, many techniques and algorithms were proposed in the literature to meet this need. However, most of the proposed methods cannot correctly detect the baseline of words that are skewed, short, have stacked-letters, and/or have unbalanced letters sizes. This paper addresses this gap by introducing a novel approach and algorithm that is capable of correctly detect the baseline of handwritten Arabic words regardless of any challenges involved in the word. The paper also verified the proposed approach via implementing it in a C code program and then applied it to the AHDB dataset. The obtained results show a successful rate more than 98%.*

Keywords: *Arabic handwritten; OCR; baseline detection.*

I. Introduction

Optical Character Recognition (OCR) is the process of digitizing characters whether they are handwritten or machine-printed. Converting texts to digital form allowing searching, editing, and compactly storing of texts. It also helps in displaying scripts in different formats. Moreover, digital text can be utilized for automatic translation and converting text to speech [1].

Baseline is the line over which word are written/printed and letters are connected. Baseline detection helps in determining the orientation of the word as well as in the skew/slant correction that might occur during writing or scanning of the words. Furthermore, most OCR techniques require the detection of the baseline in the preprocessing stage [2]. Implementing baseline detection in interactive software can improve the writing skills of language learners (e.g. kids).

Arabic is the official and co-official language of 24 countries and the first language of about 200 million people. Importantly, Arabic script is the second most widespread script in the world, where 14 languages use Arabic script [1]. Arabic words contain special challenges that does not exist in many other languages, which make the process of baseline detection a difficult task. Challenges include the probable existence of dots, diacritics, stacked letters, and/or unbalanced letters sizes [1][2].

Due to the importance of baseline detection, many approaches and methods were proposed in the literature to achieve this task. However, due to the mentioned challenges involved in Arabic language, most of the proposed techniques cannot correctly detect the baseline of many handwritten words as will be demonstrated in the third section [2].

The main contribution of this paper is the proposing of a novel approach called Multiple-Angle Histogram (MAH) that can be utilized to automatically detect the baseline of Arabic words. Despite that the method is mainly proposed and tested for Arabic words, it can be easily applied to many other languages including English.

The reminder of the paper is organized as follows. Next section introduces Arabic letters and script, so the reader will be able to understand how the proposed technique allows detection of the baseline. Third section covers the related work of baseline detection for Arabic handwriting words. Fourth section introduces the Multiple-Angle Histogram (MAH) approach and shows how it can be utilized to detect the baseline of Arabic words. Fifth section proposes a possible software implementation of the MAH. Sixth section shows the results of applying the proposed technique to the AHBD dataset. Seventh section concludes this research and summarizes the feature work.

II. A Brief About Arabic Script

Arabic script is written from right to left and Arabic letters are all one case, there is no upper or lower cases. Arabic alphabet comprises 28 letters as listed in TABLE 1, which has three columns. The first column represents the order of the letter according to the ALPHBA order, second column lists the name of the letter (how to pronounce), and third column contains the isolated shape of the letter [1][2].

TABLE 1. Arabic Alphabets

Order	Name	Letter
1	Alif	ا
2	Baa	ب
3	Taa	ت
4	Thaa	ث
5	Jeem	ج
6	Haa	ح
7	Khaa	خ
8	Daal	د
9	Dhaal	ذ
10	Raa	ر
11	Zaay	ز
12	Seen	س
13	Sheen	ش
14	Saad	ص
15	Daad	ض
16	Taa	ط
17	Dhaa	ظ
18	Ayn	ع
19	Ghayn	غ
20	Faa	ف
21	Qaaf	ق
22	Kaaf	ك
23	Laam	ل
24	Meem	م
25	Noon	ن
26	Haa	هـ
27	Waaw	و
28	Yaa	ي

Isolated shape means that the letter is not connected to any other letters nor before neither after it. As can be seen from TABLE 1, letters number 2, 3, and 4 share a common primary shape and varies only in the number and position of the dots. Letter number 2 has only one dot below its primary shape, letter number 3 has two dots above its primary shape, and letter number 4 has three dots above its primary shape. Similarly, letters number (5, 6, & 7), (8 & 9), (10 & 11), (12 & 13), (14 & 15), (16 & 17), and (18 & 19) share the same common primary shape and varies only in the number and position of dots. This makes the process of Arabic character recognition a challenging task [1][2].

In order to write a meaningful word, these letters have to be connected together. Connecting letters change its primary shape depending upon the position of the letter inside the word as shown in TABLE 2, which has four columns. First column represents the order of the letter. Second column shows the shape of the letter when it is written at the beginning of the word. Third column shows the shape of the letter when it is written in the middle of the word (middle here does not necessary mean exact middle but rather means in a position where one/more letters is/are before it and one/more letter is/are after it). Fourth column shows the shape of the letter when it is written at the end of the word.

TABLE 2. Arabic Alphabets According to Their Position

Order	Initial position	Medial position	Final position
1	ا	ـا	ـا
2	بـ	ـبـ	ـب
3	تـ	ـتـ	ـت
4	ثـ	ـثـ	ـث
5	جـ	ـجـ	ـج
6	حـ	ـحـ	ـح
7	خـ	ـخـ	ـخ
8	د	ـد	ـد
9	ذ	ـذ	ـذ
10	ر	ـر	ـر
11	ز	ـز	ـز
12	سـ	ـسـ	ـس

13	شـ	ـشـ	ـش
14	صـ	ـصـ	ـص
15	ضـ	ـضـ	ـض
16	طـ	ـطـ	ـط
17	ظـ	ـظـ	ـظ
18	عـ	ـعـ	ـع
19	غـ	ـغـ	ـغ
20	فـ	ـفـ	ـف
21	قـ	ـقـ	ـق
22	كـ	ـكـ	ـك
23	لـ	ـلـ	ـل
24	مـ	ـمـ	ـم
25	نـ	ـنـ	ـن
26	هـ	ـهـ	ـه
27	و	ـو	ـو
28	يـ	ـيـ	ـي

As can be seen from TABLE 2, some letters have three shapes and some others have only two shapes according to their positions [2].

Arabic words are written by connecting two or more letters over a virtual line called baseline. Detection of the baseline is very important for readers as well as for automatic character recognition. The first stage in determining the word orientation is the detection of baseline. Next section covers the most important approaches proposed in the literature for the purpose of automatic baseline detection of Arabic words.

III. Related Work

As mentioned earlier, the baseline is the line over which letters are connected. Consequently, a straightforward technique to detect the baseline is via vertical projection of the binary image, where the baseline is determined by the maximal peak of the histogram of pixels as shown in Fig. 1. This method is based upon the assumption that the word was horizontally written and was converted to pixels without skew or slant. Accordingly, this technique might benefit machined-printed text that is perfectly scanned but it is ineffective for handwriting text.

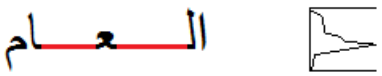


Fig. 1. (a) An example of Arabic word, red lines show how letters are connected together over the baseline. (b) Vertical projection of the pixels of the word in (a).

The authors of [3] proposed a method that comprises four main steps. The first step is extraction of the word features (e.g. area of bounding box, degree of vertices, length of edges, and deviation of skeleton length). The second step is removing of connected components that are not part of the primary letters shapes (e.g. diacritics, and dots). The third step is initial estimation of the baseline. The fourth and last step is final estimation of the baseline. Despite that this approach is complicated comparing to the previous one, it failed in many cases including short and long words. Furthermore, this technique is very expensive to implement and its complexity increases with the font size.

Another approach that is based on the minima points of the word contour was introduced in [4]. However, this method failed in case the diacritics is large relative to the word unless diacritics were detected and removed, which increases the complexity of the method. This technique also failed if the words have stacked-letters.

The idea of utilizing Voronoi diagram to detect the baseline of Arabic words was proposed in [5]. The idea requires edge detection and word contour tracing. However, the authors of [5] just proposed an abstract idea, no results were given about applying the method on real dataset.

The authors of [6] proposed a technique that contains two phases. The first phase includes removing of noise, dots, and diacritics and the second phase includes horizontal histogram project of each sub-word, text thinning, and circle shape detection. Each of these sub-phase processes requires its own algorithm. This approach was tested on words that are relatively short or contains diacritics, no attempts were made for skewed words or words that contained stacked letters.

In conclusion, the literature does not have an efficient and easy to implement approach that can automatically detect the baseline of handwritten words that have several challenges such as short (e.g. less than four letters), contains stacked letters, and/or skewed during writing or scanning.

IV. Utilizing Multiple-Angle Histogram (MAH) For Baseline Detection Of Arabic Words

As described in the previous sections, Arabic letters are connected over a virtual line called baseline, where the connection occurs by the means of straight line. Therefore, if the word is written and scanned without skew, vertical projection of the words pixels allows determining the baseline via determining the maximal peak of the histogram of pixels. However, skew or slant during writing and/or scanning render this method inefficient. Accordingly, this paper tried to optimize this method in order to determine the baseline of Arabic words regardless of skew that might occur during writing or scanning. The following text describes the proposed method in details.

Image histogram is a statistical description for the distribution of pixels that have specific features. The relative frequency of occurrence of these pixels can be plotted versus different parameters to extract many useful information from the image.

In this paper, we propose a new type of histogram called Multiple-Angle Histogram (MAH), which can be defined as the distribution of pixels that have specific features and pass through any straight-line in the image. It is worth highlighting that horizontal and vertical histogram are special cases of the MAH, where the straight-line are horizontal and vertical respectively.

If the word is captured by a black-and-white image (black pixels represent the word and the white pixels represent the background), then determining the maximal peak line of MAH is the maximum likelihood baseline.

V. An Algorithmic Implementation Of MAH

The following algorithm is a possible efficient way of implementing MAH for baseline detection:

```
For i from 0 to image_height
      For j from 0 to image_height
            CountPoints[]=GetPoints(0,i,image_width,j)
      End loop
End loop
For i from 0 to image_width
      For j from 0 to image_width
            CountPoints[]=GetPoints(i,0,j,image_height)
      End loop
End loop
Index=GetIndex(CountPoints,Max(CountPoints))
```

The functions utilized in this algorithm is defined as:

`GetPoints(x1,y1,x2,y2)` returns the number of black points pass through the straight line determined by (x1,y1) and (x2,y2)

`Max(A)` returns the largest element of an array A

`GetIndex(a, B)` returns the first index of the element that has value a in an array B

VI. Results And Discussion

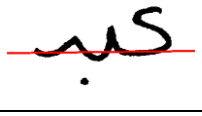
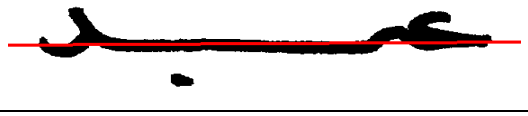
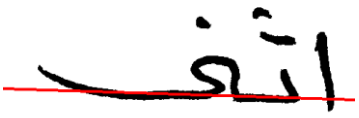
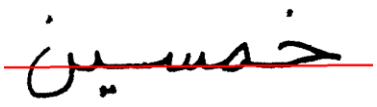
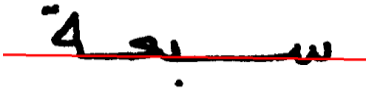
For validation purpose, the proposed approach is implemented in a C code program as described in the above algorithm. Then the program is utilized to test all samples in the Arabic Handwriting Data Base (AHDB) [7]. This dataset has been collected in Qatar University for the purpose of Arabic Handwriting Recognition tasks. Additionally, some other samples collected from different other sources are also utilized to test the proposed technique and algorithm. Each sample is an image that contains an Arabic word handwritten in black pixels over a white background and has a red straight line. The red line is drawn by the developed program during testing to represent the maximum likelihood baseline.

In order to analyze the features, advantages, and disadvantages of the proposed technique, the tested samples are divided into several categories according to the difficulty level of baseline detection. Each group contains handwritten words that have number of challenges. The following sub-sections define each group and shows the results of testing some samples in that group.

A. Unskewed Words

This group contains words that were written horizontally and scanned without any skew or slant. The baseline of these words is easy to detect by many techniques including the ones introduced in [3][4][5][6]. TABLE 3 contains the results of testing ten samples of this category. As can be demonstrated by the provided samples, the successful rate of baseline detection of words in this category is above 98%.

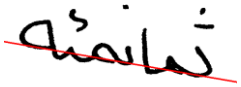
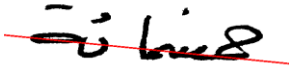
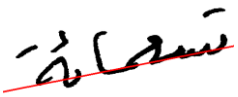
TABLE 3. Ten Samples of Unskewed Words Tested by the Proposed Approach

B. Skewed Words

This group contains words that were skewed either during writing or during scanning. The baseline of these words is hard to detect. The technique in [3] cannot detect the baseline of these words. TABLE 4 contains the results of testing ten samples of this category. As can be demonstrated by the provided samples, the successful rate of baseline detection of words in this category is above 98%. The results also show that the proposed technique can detect the right baseline regardless of the skew angle and direction.

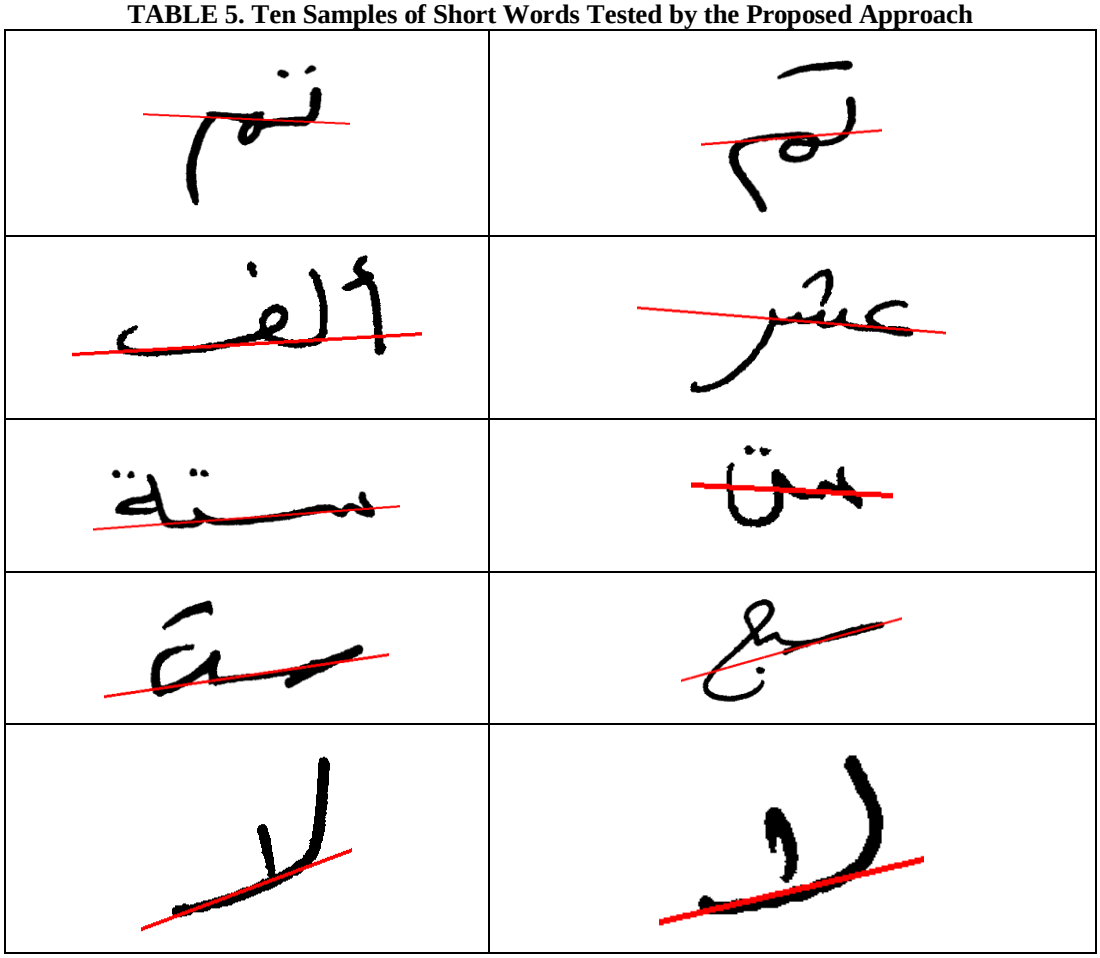
TABLE 4. Ten Samples of Skewed Words Tested by the Proposed Approach

C. Short Words

Short words are words that have less than four letters. In Arabic handwritten scripts, automatic baseline detection of these words is very hard due to difficulty involved in determining the orientation of the word. The baseline of words in this group cannot be detected by the techniques proposed in [3][6]. TABLE 5 contains the results of testing ten samples of this category. As can be demonstrated by the provided samples, the successful rate of baseline detection of words in this category is more than 98%. Despite that some samples contain only two letters, the proposed approach was able to correctly detect their baseline. The results also show that the proposed approach can detect the right baseline regardless of the existing skew in some samples.

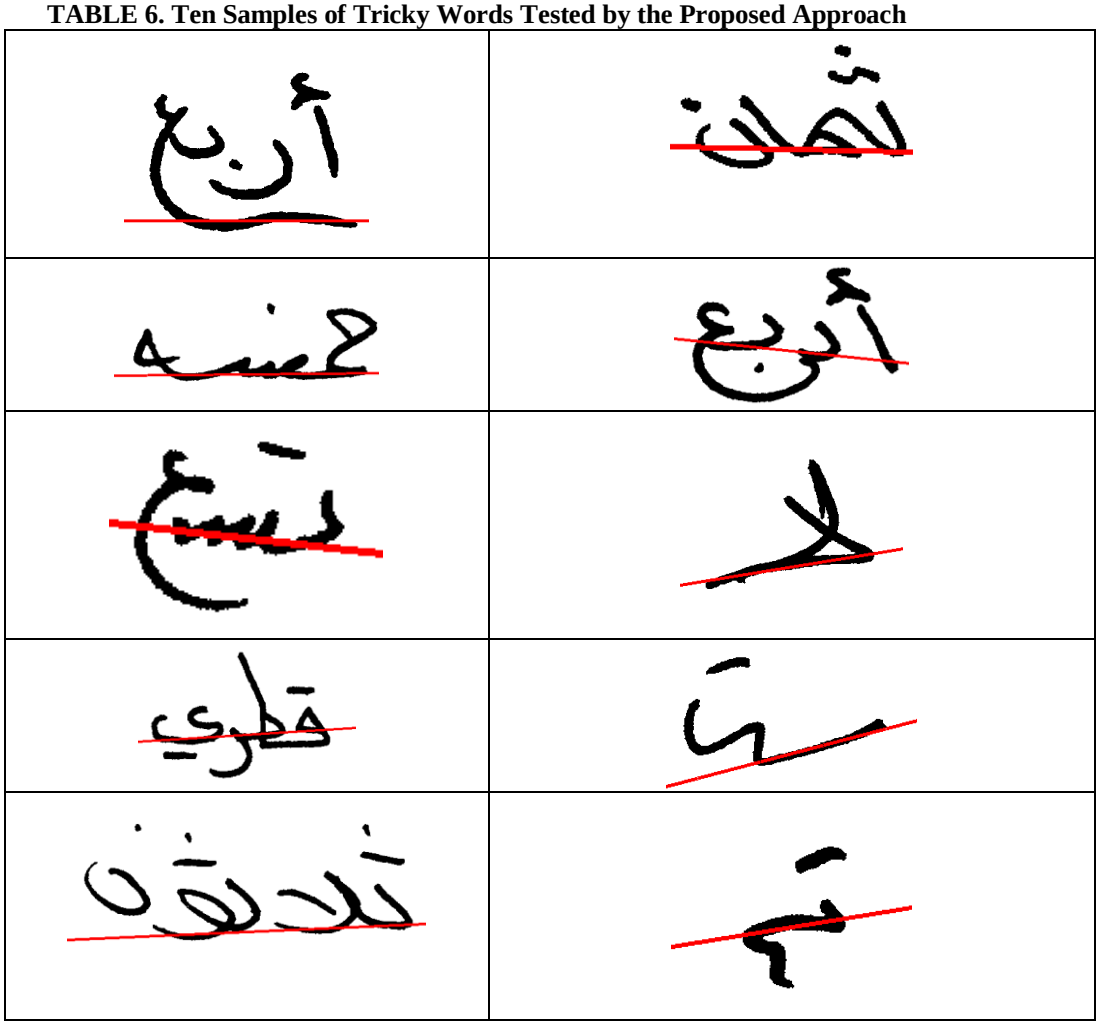
TABLE 5. Ten Samples of Short Words Tested by the Proposed Approach



D. Tricky Words

This group contains words that have special challenges like stacked letters, unbalanced letters sizes, or significantly curved letters in addition to previous challenges (e.g. skewed or short words). The baseline of these words cannot be detected by the techniques proposed in [3][4][6]. TABLE 6 contains the results of testing ten samples of this category. As can be demonstrated by the provided samples, the successful rate of baseline detection of words in this category is more than 98%.

TABLE 6. Ten Samples of Tricky Words Tested by the Proposed Approach



The proposed approach is also tested on short sentences that contain several words. Fig. 2 shows an example of a sample that contains three words skewed during writing. The technique proposed in [3] was not able to correctly detect the right baseline of this sentence. However, the proposed approach successfully detect the right baseline.

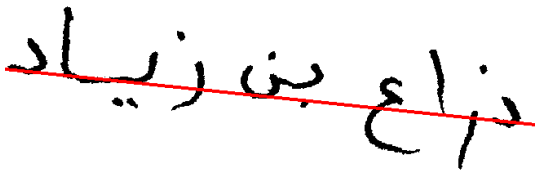
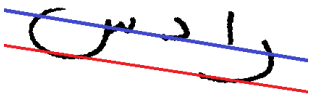
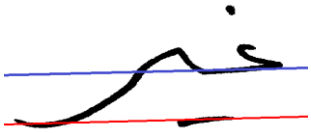


Fig. 2. A sample contains three words tested by the proposed approach.

Finally, it is worth mentioning that the proposed approach failed to detect the right baseline of some words. Some examples are given in TABLE 7. For each sample, in addition to the red line that represent the maximum likelihood baseline detected by the proposed approach, a blue line is drawn to represent the right baseline of the word. Comments are given beside each sample to describe the failure mechanism.

TABLE 7. Five Samples of Words Tested by the Proposed Approach, which Failed to Detect Their Right Baseline

	The proposed technique failed to detect the right baseline due to two main reasons: 1) the word is skewed and 2) the relative sizes of letters are not balanced. Despite that, the detected baseline is parallel and very near to the right one.
	The proposed technique failed to detect the right baseline due to two main reasons: 1) the word is very short and 2) the relative sizes of letters are not equal. Despite that, the detected baseline is parallel and very near to the right one.
	The proposed technique failed to detect the right baseline due to the reason that the word was not written on the same virtual line. However, the difference between the detected line and the right line is less than 5o.
	The proposed technique failed to detect the right baseline due to two main reasons: 1) the word is too compressed, and 2) the word has two stacked letters out of three. However, the detected baseline is parallel and very near to the right one.
	The proposed technique failed to detect the right baseline due to two main reasons: 1) the word is very short, and 2) the relative sizes of the letters are not equal. Despite that, the detected baseline is parallel and very near to the right one.

VII. Conclusion And Future Work

Baseline detection of handwritten or printed words is an important step for OCR and it has several useful applications. Arabic script is the second most widespread script in the world. This research papers proposed a novel approach that is capable of detecting the baseline of handwritten Arabic words with high successful rate (> 98%). This figure was obtained by applying the proposed technique on the AHDB dataset. The proposed technique failed in some cases due to some challenges in the written words themselves. Even in those rare failed cases, the obtained baseline is very near to the right one.

The future work of this research will concentrate on utilizing the proposed technique in the recognition of handwritten Arabic long texts.

References

[1] Märgner, V.; El Abed, H., Guide to OCR for Arabic Scripts (Springer London).
[2] Lorigo, L.M.; Govindaraju, V., Offline Arabic handwriting recognition: a survey, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 28(5), 2006, 712-724.
[3] Pechwitz, M.; Margner, V., Baseline estimation for Arabic handwritten words, Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop, 2002, 479-484.
[4] Farooq, F.; Govindaraju, V.; Perrone, M., Pre-processing methods for handwritten Arabic documents, Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on, 1(29), 2005, 267-271.
[5] Al-Shatnawi, A.; Omar, K., Detecting Arabic handwritten word baseline using Voronoi Diagram, Electrical Engineering and Informatics, 2009. ICEEI '09. International Conference on, 2009, 18-22.
[6] Abu-Ain, T.; Sheikh Abdullah, S.; Bataineh, B.; Omar, K.; Abu-Ein, A. A Novel Baseline Detection Method of Handwritten Arabic-Script Documents Based on Sub-Words, Soft Computing Applications and Intelligent Systems, Communications in Computer and Information Science, 67-77.
[7] http://handwriting.qu.edu.qa/dataset/.