Arabic Handwritten: Pre-Processing and segmentation

Makki Maliki, Naseer Al-Jawad, Harin Sellahewa, and Sabah Jassim
The University of Buckingham
Buckingham, MK18 1EG, UK
Email: {makki.maliki; naseer.aljawad; harin.sellahewa sabah.jassim}@buckingham.ac.uk

ABSTRACT

This paper is concerned with pre-processing and segmentation tasks that influence the performance of Optical Character Recognition (OCR) systems and handwritten/printed text recognition. In Arabic, these tasks are adversely affected by the fact that many words are made up of sub-words, with many sub-words the same shape or other characters may be connected with their sub-word’s body; there could be multiple instances of sub-words overlap. To overcome these problems we investigate and develop segmentation techniques that first segment a document into sub-words, link the diacritics with their sub-words, and removes possible overlapping between words and sub-words. We shall also investigate two approaches for pre-processing tasks to estimate sub-words baseline, and to determine parameters that yield appropriate slope correction, slant removal. We shall investigate the use of linear regression on sub-words pixels to determine their central x and y coordinates, as well as their high density part. We also develop a new incremental rotation procedure to be performed on sub-words that determines the best rotation angle needed to realign baselines. We shall demonstrate the benefits of these proposals by conducting extensive experiments on publicly available databases and in-house created databases. These algorithms help improve character segmentation accuracy by transforming handwritten Arabic text into a form that could benefit from analysis of printed text.

Keywords: Arabic Handwritten, OCR, Baseline estimation, Slope correction, Projection, Segmentation, Overlap problem.

1. INTRODUCTION

Arabic Handwritten Optical Character Recognition (AHOCR) is a challenging task in computer pattern recognition and image processing. AHOCR have a variety of applications including: digitising historic archives, security, authentication [1] [2], forensic [3], and more. In such applications, however, recognising words and sentences are the main objectives. Traditionally, documents are first segmented into lines, words, and then characters, using labelling connected component strategy. Words in Arabic, Farsi, Urdu, Kurdish, and other similar languages consist of different types of characters. Some characters can be connected with their neighbours on one or both sides, while other characters may be completely disconnected from its neighbours. Indeed, words in such languages are aggregation of sub-words that consist of one or more characters. Moreover, some characters may have a diacritic that distinguish it from characters of the same shape or the same pronunciation. The shape of the diacritic can be either small dot or small symbol. In printed texts, this structure makes a word consist of multiple sub-words separated by narrow spaces [4], as shown in Figure 1.

![Figure 1: Printed Arabic Statement: (a) Words are indicated by overlines, (b) Sub-words are underlined](image)

In printed or handwritten text, there are two approaches to OCR: holistic or segmentation-based. In holistic approaches words/sub-words are recognised as a whole without segmenting them into characters [5]. In a segmentation-based approach, characters are recognised individually after extraction from words or sub-words [4]. OCR problems are dealt with using a list of known procedures: pre-processing, segmentation, feature extraction, recognition, and post...
recognition. Pre-processing consists of filtering, binarisation, line segmentation, word segmentation, slope correction, slant removal, baseline estimation, and size normalizing as shown in Figure 2.

![Figure 2: Block diagram for a Typical OCR System](image)

The main focus of this paper is the segmentation of Arabic handwriting text into characters. We shall investigate and develop two interleaved steps approach to deal with this problem. Firstly, documents will be segmented into sub-words and overlaps between sub-words will be eliminated. We shall adopt a Labelling the Connected Component algorithm (LCCA) for the sub-words segmentation. The second step, referred to as the pre-processing step, works by removing/correcting the slope/slant form the extracted sub-words to improve the chance of accurate character segmentation. For efficiency, the slope/slant correction algorithm does not wait for the entire document to be sub-word segmented. The developed algorithms form the core of our proposed AHOCR scheme, depicted by the block diagram shown in Figure 3, which is a much slim version of a traditional OCR scheme as depicted above.

![Figure 3: The proposed AHOCR System](image)

The rest of this paper is organised as follows; Section 2 covers the segmentation stage where overlapping between sub-words is discussed, section 3 discusses the pre-processing stage where the slope/slant removal are performed, in section 4, the experiments will be discussed, and finally, the character segmentation, conclusions will be the last step of this paper. For the benefit of the non-Arabic readers, we will present the printed version of the handwritten texts.

### 2. WORDS AND SUB-WORDS SEGMENTATION

The sub-word structure in handwriting generates challenging problems resulting in segmentation errors which may in turn leads to incorrect character/word recognition. These problems are mostly due to textual overlaps between lines [6], neighboring words and sub-words, between sub-words on consecutive lines, or even between characters. Removal of overlaps between sub-words is expected to reduce character overlaps.

#### 2.1. Sub-words segmentation and overlap removal

Generally segmentation of sub-words exploits the presence of a small vertical gap between sub-words. In the case of printed text, this task is fairly straightforward due to the fact that each character has its own space in the form of a vertical box and boxes for two characters can touch but overlap. Berkani & Hammami, [7], segmented printed words to sub-words depending on the vertical projection as shown in Figure 4. However, unlike printed words, it has been noticed that handwritten sub-words are mostly overlapped, even after the pre-processing procedure is performed as shown in Figure 5. For handwritten text, connected component techniques have been used for sub-words segmentation, but without paying reasonable attention to the overlapping problem. Alkhateeb et al [8] developed a connected component...
algorithm to extract sub-words from words for handwritten text using 8-neighbours pixels connectivity. The algorithm encloses detected sub-word components in rectangular boxes. They tested the algorithm on 200 images from IFN/ENIT database, and reported an accuracy rate of 85%.

Figure 4: Sub-words vertical segmentation [7]

Figure 5: (a) Arabic Printed Words, (b) Arabic Handwritten Words. The encircled areas are overlapping sub words

3. PRE-PROCESSING STAGE: SLOPE/SLANT REMOVAL

The pre-processing stage aims to improve the appearance of segmented handwritten sub-words to make them look like printed texts. Sloped and slanted words may cause difficulties in OCR especially in character segmentation stage. In this section we first review existing pre-processing methods, and propose alternative modifications.

3.1. Slope/slant/baseline existing methods

The slope of a word (or a sub-word) is the inclination of its baseline, i.e. the angle the baseline makes with the horizontal line [9]. Slope removal is used to realign the baseline. A number of methods have been developed in the literature for baseline detection including horizontal projection [10], word skeleton [11], word contour representation [12], and principal components analysis (PCA) [13]. These methods have different complexity and process time.

Slants refer to deviation of strokes from the vertical axis, which varies between different writers [9]. Many approaches try to determine the slant angle by detecting the vertical strokes, the parts that lay above baseline zone, and taking the average angle of those as the shear angle. Based on this shear angle the whole vertical strokes will be corrected [9], [14], [15], [16], [17], and [18].

3.2. Slope/slant removal proposed algorithms

In this section we shall investigate and develop algorithms for correcting sloped/slanted sub-words using two alternative techniques that are used for correcting sloped/slanted words: linear regression applied to the word pixels as data points in two dimensional spaces, and multiple incremental rotations is used.


Many researchers have applied slope and slant removal on all words by using regression [12]. Here we suggest doing it on part of sub-word. The best linear model can be calculated by equations 1, 2, and 3.

\[ \tilde{y} = \text{slope} \times x + \text{intercept} \]  
\[ \text{slope} = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2} \]  
\[ \text{intercept} = \tilde{y} - \text{slope} \times \bar{x} \]

\( \tilde{y} \) is the mean of \( y \), and \( \bar{x} \) is the mean of \( x \).
The naive regression algorithm for rotating a sub-word image is based on the use the slope of the regression line of its pixels, i.e. slope = tan(0), where 0=tan-1(slope) is the angle shown in Figure 6. This is illustrated in Figure 7.

Figure 7: Naïve regression algorithm. A. Printed sub-word, B. The original sloped Sub-word, and C Rotated handwritten sub word image by using linear regression equation.

Figure 7, demonstrate benefits of using the naïve regression algorithm but it is not suitable for some sub-words, especially when we have long ascenders or descenders (upper or lower tails) as shown in Figure 8A. Indeed the experimental results given in Table 1, below, demonstrate the shortcoming of this algorithm. To achieve better results, we should first remove tails and only then calculate linear regression for the remaining part. Tails are easily determined from the sub-word horizontal projections as illustrated in Figure 8A (d). The angle defined by the slope of the no-tails regression line can be used to rotate the original sub-word, see Figure 8B. This modification of the naïve regression algorithm will be referred to as the No-tails Regression algorithm and experimental results shown in Table 1, below, demonstrate significantly improved results but there are still more room for improvement.

Figure 8: No-tails Regression algorithm. (a) and (b) printed sub-words, (d) and (e) example of sloped sub-words. (c) Projection of (d). Circled areas are tails. B: (a) and (b) after slope removal, by using linear regression after cancelling tails.

The success of this method is a consequence of the fact that we exclude outliers (the strokes in this case) from the regression line estimator. This motivates testing algorithms that limit the data input to the regression lines calculation by removing outliers in terms of vertical and horizontal boxes around the central value or even high density. We have implemented the corresponding three modified versions of the regression line strategy: Regression in a vertical box (Figure 9A), Regression in a horizontal box (Figure 9B), and Regression of the highest density region (Figure 9C).
3.3. The proposed slope/slant removal using incremental rotations

The multi rotating procedure has been applied by Al-Rashaideh [19] to find the maximal horizontal projection peak of words which defines the slop of the word. This algorithm, referred to as the horizontal peak projection (HPP) algorithm, does not give any consideration to vertical projection peaks which we find to improve accuracy of slope/slant removal for sub-words.

We have first implemented the HPP algorithm restricted to sub-words as shown in Figure 10B. Moreover, in order to increase the accuracy of slope and slant removal, we extend this algorithm by determining both the highest horizontal peak (HHP) and the highest vertical peak (HVP), and then combining them to produce the highest total peak (HTP) in every rotating angle as shown in Figure 10C. We shall refer to this algorithm as the HTP-rotation algorithm, and it works by calculating the HTP in:

a) The current sub-word with zero rotation.
b) The one degree rotated sub-word in clockwise direction.
c) The one degree rotated sub-word in the anti-clockwise direction.

From the three above, if the highest of the HTP is (a) then no extra rotation is needed. If the highest of the HTP is in the clockwise direction (b) then a clockwise increase of three degree will be performed four times and the degree related to highest of the HTP will be considered as the slope/slant of that sub-word. And same will be applied for (c) but in anti-clockwise direction. The output from the HHP and HTP algorithms are illustrated in Figure 10, below. In each case we display a printed word, a handwritten version of it, the corresponding projection(s) and the corrected word.

4. EXPERIMENTAL AND DISCUSSION

4.1. Databases

In order to test the performance of our procedures of sub-word detection and character segmentation, two types of databases are used: the IFN/ENIT database [20] and a new database of handwritten text that we collected at the University of Buckingham. The IFN/ENIT database contains 26,459 images of single words representing Tunisian towns’ names, written by 411 different writers. In total IFN/ENIT contains more than 210000 characters. Unlike the IFN/ENIT database, our in-house database contains 120 multi-sentence documents written by 60 persons whose ages are between 8 and 85 years old. Each writer produces two different documents. This database has 6080 words, 12580 sub words and 26820 characters.
4.2. Experimental results

So far, there are no objective methods for testing the accuracy of extracting sub-words, slopes corrections, slants removal or baseline estimation [12]. In our testing, we have developed a GUI system to subjectively examine, randomly, 6000 words from IFN/ENIT, and 6080 words from Buckingham database. Table 1 shows the accuracy of the proposed algorithms for the two databases.

The error rate in the proposed LCCA sub-word segmentation is less than 3%, this is because of the touches between subwords. The HTP (Incremental Rotation) for slope/slant removal is encouraging. But it needs more processing time because of the multiple rotations compared with the others. The worst case scenario is six time rotations for each sub-word while the best case is only zero rotation. So we are recommending using it in off line recognition, while other algorithms can be used by online recognition.

Table 1. Experimental Results

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Buckingham DB</td>
</tr>
<tr>
<td>LCCA Segmentation</td>
<td>97.45%</td>
</tr>
<tr>
<td>Naïve Regression</td>
<td>46.1%</td>
</tr>
<tr>
<td>No-Tails Regression</td>
<td>80%</td>
</tr>
<tr>
<td>Restricted Vertical Regression</td>
<td>84.65%</td>
</tr>
<tr>
<td>Restricted Horizontal Regression</td>
<td>76.48%</td>
</tr>
<tr>
<td>High-Density region Regression</td>
<td>85%</td>
</tr>
<tr>
<td>HHP – slope/slant removal</td>
<td>94%</td>
</tr>
<tr>
<td>HTP – slope/slant removal</td>
<td>98.11%</td>
</tr>
</tbody>
</table>

5. SEGMENTATION OF SUB-WORDS INTO CHARACTERS

To demonstrate the positive effect of our proposed algorithms on segmentation accuracy, we test characters segmentation approaches of overlapped sub-words. There are many segmentation strategies to extract characters from words. These strategies may depend on baseline [21], vertical/horizontal projection [22], contour tracing [23], and skeleton/thinning representation [24], [25]. Some researchers combined some of these strategies together like in [26].

We are currently developing a character segmentation algorithm that combines the skeleton method, by thinning each sub-word, with the baseline and the vertical projection methods. The thinning helps to find the vertical junction points across the baseline zone that separate characters. See Figure 11B for an illustration of this process that segments the sub-words to characters.

Clearly segmenting sub-word without solving overlapping problem, will definitely lead to faulty characters segmentation as shown in Figure 11A.

![Figure 11](image-url)

Figure 11: A: (a). Arabic Printed Words, (b). Arabic Handwritten Words, (c). Slope/Slant Removal then character segmentation. The Rectangle areas are faulty segmentation. B: (a). Arabic Handwritten Words, (b). segmentation words to sub-words by using LCCA, (c) Slant Removal followed by character segmentation.
6. CONCLUSION

We investigated sub-words segmentation in Handwritten Arabic text, and we paid a special attention to the overlapping problem that occurs between handwritten sub-words as a mean of improving recognition of handwritten Arabic text. We have modified the well-known LCCA algorithm, by adding a vertical search in order to include with the segmented sub-words all their diacritics. We have shown that this procedure increases character segmentation accuracy significantly too more than 97% in comparison to the 85% accuracy reported by Al Alkhateeb for a relatively small subset of the IFN/ENIT database. We have also developed a number of algorithms to remove slope/slant from sub-words. Existing techniques process slope and slant independently, whereas we do them simultaneously. We test versions of regression line algorithms for this purpose which achieves high accuracy rates, but our HTP algorithm that is based on incremental multi-rotation is attains superior results with 98% accuracy for slope/slant correction. We have also demonstrated the positive effect of the developed methods for segmenting characters. Indeed, our algorithms seem to transform handwritten text to near printed format.

REFERENCES