Interactive Knowledge Discovery for Baseline Estimation and Word Segmentation in Handwritten Arabic Text

Jawad H AlKhateeb, Jianmin Jiang, Jinchang Ren and Stan Ipson
University of Bradford
United Kingdom

1. Introduction

Electronic document management systems provide great benefits to society. Software tools such as word processors are used in the generation, storage, and retrieval of documents in specific formats. Using such tools, documents can be edited, printed, or distributed electronically across networks. However, with paper documents, the previous tasks cannot be accomplished by computers, so there is a need to extract the information in documents to store them in a computerized format. The solution for this task is in the branch of pattern recognition known as Document Analysis and Recognition (DAR). The main aim here is to imitate the human ability in reading text with high speed and accuracy. Optical Character Recognition (OCR) is the most crucial part of DAR (Khorsheed 2002) .

As time passes, computers become more powerful, and tasks can be done quicker. It is still however necessary to make computers more versatile, by enabling them to carry out tasks that are natural to humans, such as the ability to read the machine printed or handwritten text. The automatic recognition of a document requires transferring the text in an image file. This process causes the system to lose any temporal information relating to the text {Khorsheed, 2002 #14}.

Automatic recognition has enabled many applications such as office automation, banking in terms of verification of cheques, data entry and mailing services in terms of post/zip codes {Lorigo, 2006 #2}. In such applications, the interaction between the man and the machine can be improved by implementing character recognition systems (Amin 1997).

Handwritten text recognition has significant potential for such applications. More importantly, it may be used as a natural form of human-computer interaction. In general, this task can be divided into online based or offline based systems. Recognition in the online based systems is based on pen movements, which is the dynamics of writing. However, recognition in the offline based systems is based solely on the written text image. Offline recognition is the more difficult of two because it cannot make use of additional information available to online systems such as the strength and sequential order of the writing [1]. In this paper, the focus is on offline recognition of handwritten Arabic text.A large number of research papers have been written relating to Latin, Chinese, and Japanese handwriting. On

the other hand, relatively little research has been done on Arabic handwriting. This is due to the complexity of Arabic text and to a lack of Arabic databases. The automated methods for the recognition of Arabic text are at the early stage compared to the methods of recognition of Latin, Chinese, and Japanese texts. In addition, there is a major challenge in the Arabic writing recognition systems due to the cursive nature of the data. In this chapter, we emphasize on offline recognition of handwritten Arabic text.

Arabic is written by more than 250 million people (Amin 1997). By nature, Arabic text is cursive, which makes its recognition rate lower than that of printed Latin. In a similar way to English, Arabic writing uses letters. The Arabic alphabet consists of 28 letters, and text is written from right to left in a cursive way. Each Arabic letter has either two or four shapes depending on its position in the text. The shapes are classified based on their position which can be start, middle, end, or alone {Amin, 1996 #22}. Table 1 shows each shape for each letter. For example letter Ayn (ع) has the following shapes: start عـ, middle ـعـ, end ـع, and alone ع. In addition, Arabic language uses diacritical marking such as fattha, dumma, kasra, hamza(zigzag), shadda, or madda. The presence or absence of vowel diacritical indicates different meaning {Amin, 1998 #20}. For example some words are written in the same way, but they are different in the meaning such as: مدرسة , which can be school or teacher; كلية , which can be college or kidney; حب , which can be love or seeds. Normally, the diacritical marking are not written in the handwriting, but if the words are isolated, diacritical marking are essential to differentiate between the possible meanings. Using dots makes some Arabic letters special {Amin, 1998 #20; Lorigo, 2006 #2; Amin, 1997 #15} as follows:

- Ten Arabic letters have one dot (ب،ج،خ،ذ،ز،ض،ظ،غ،ف،ن)
- Three Arabic letters have two dots (ت،ق،ي)
- Two Arabic letters have three dots (ث،ش)
- Several Arabic letters presents loop (ص،ض،ط،ظ،عـ،غـ،ف،ق،م،مـ،و،ة)

It is worth knowing that removal of any of these dots will lead to a misrepresentation of the character. So, efficient pre-processing techniques have to be used in order to deal with these dots without removing them and changing the identity of the character. There are six letters which are not connected from the left resulting in the separation of the word into sub-words or pieces of Arabic words (PAW) {Lorigo, 2006 #2}. Figure 1 shows examples of Arabic words with one, two, and three sub-words.

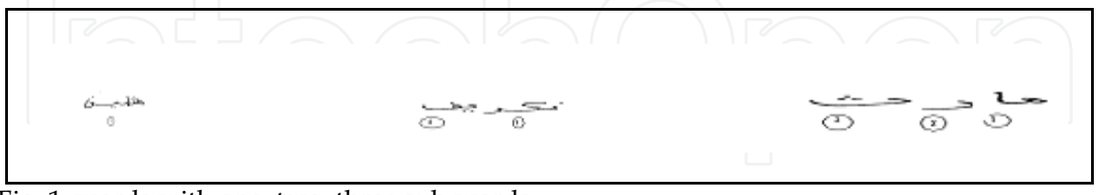

Fig. 1. words with one, two, three sub-words

Generally, the handwritten text is written on a page divided into lines which are further divided into words. There are spaces between the lines, and there are spaces between the words. The spaces between the words define the word boundaries. Normally, the space between the sub-words is one third of the space between the words. This is done consistently in printed text, but it varies in handwritten text {Amin, 2000 #19}.

To this end, a range of attempts have been reported in the literature on Arabic text recognition. Almuallim and Yamaguchi {Almuallim, 1987 #70} proposed a structural recognition technique for Arabic handwritten words. Their method thinned and segmented the words into strokes. They used coordinates to represent the continuous stroke curves. Their method extracted each stroke's start and end points. Finally, the strokes were classified based on their topological and geometrical features. They tested their method on a database of 400 words written by two persons. However, their system failed in most cases due to incorrect segmentation of words.

This chapter focuses on the pre-processing phase of Arabic handwritten text recognition and introduces new methods for baseline estimation and word segmentation. The rest of this chapter is structured in four sections, where section 2 describes state of the art; section 3 describes the proposed methods including baseline detection, Extracting connected components and sub-words, and Segmentation of Words; section 4 presents experimental results and discussion, and finally, section 5 provides concluding remarks.

Name	Alone (isolated)	Start	Middle	End
Alif	ا	ا	ـا	ـا
Baa	ب	بـ	ـبـ	ـب
Taa	ت	تـ	ـتـ	ـت
Thaa	ث	ثـ	ـثـ	ـث
Jeem	ج	جـ	ـجـ	ـج
Haa	ح	حـ	ـحـ	ـح
Khaa	خ	خـ	ـخـ	ـخ
Dall	د	د	ـد	ـد
Dhaal	ذ	ذ	ـذ	ـذ
Raa	ر	ر	ـر	ـر
Zaay	ز	ز	ـز	ـز
Seen	س	سـ	ـسـ	ـس
Sheen	ش	شـ	ـشـ	ـش
Saad	ص	صـ	ـصـ	ـص
Daad	ض	ضـ	ـضـ	ـض
TTaa	ط	طـ	ـطـ	ـط
Dhaa	ظ	ظـ	ـظـ	ـظ
Ayn	ع	عـ	ـعـ	ـع
Ghyan	غ	غـ	ـغـ	ـغ
Faa	ف	فـ	ـفـ	ـف
Qaaf	ق	قـ	ـقـ	ـق
Kaaf	ك	كـ	ـكـ	ـك
Laam	ل	لـ	ـلـ	ـل
Meem	م	مـ	ـمـ	ـم
Noon	ن	نـ	ـنـ	ـن
Haa	ه	هـ	ـهـ	ـه
Waw	و	و	ـو	ـو
Yaa	ي	يـ	ـيـ	ـي

Table 1. Arabic letter shapes**.**

2. Previous work

Amin and Alsadoun (Amin and Al-Sadoun 1992) proposed a new technique for segmenting hand printed Arabic text using binary trees and a parallel thinning algorithm {Guo, 1989 #72} for producing the skeleton of the image. They traced the thinned image from right to left using a 3×3 window and recorded the structure of the traced parts. They used the Freeman code {Freeman, 1961 #73} to describe the primitives. A binary tree consisting of several nodes is constructed using specified rules. Each node is used to describe the shape part of a connected component. After construction of the binary tree, smoothing is done in order to minimize the number of nodes, minimize the Freeman code string, and to minimize any noise in the thinned image. Finally, they implemented segmentation by dividing the binary tree into several sub-trees in which each sub-tree represents a character. Advantages of their proposed technique are the abilities to segment overlapping characters and characters which have short connection between them. Motawa et al. (Motawa, Amin et al. 1997) introduced an algorithm for segmenting Arabic words into characters by applying mathematical morphological techniques. Several pre-processing tasks were performed on the input images including binarization, slant correction and connected components construction. The slant correction process detected the slope first using a single erosion operation before correcting it. Finally, connected components were found and contours applied to extract sub-words and the complementary characters. The segmentation algorithm performed first a filtering operation for noise removal. This was done by two successive morphological operations (closing followed by opening). Second, singularities were found by applying opening to the word image. Third, regularities were found by subtracting the singularities from the original image. For the recognition process, hidden Markov models (HMMs) were used to test the algorithm on a few hundred words resulting in a good recognition rate of 81.88%. Abuhaiba et al. {Abuhaiba, 1996 #62} dealt with several problems in the processing of binary images of handwritten text documents. First, applying the distance transform to the thinned image, they created an algorithm which extracts the straight line of a textual stroke. The goal of this method is to identify the spurious points from the thinned images. The extracted straight lines keep the structural information of the original pattern. Second, a threshold is calculated in order to remove outlying pixels whose distance exceeds the threshold. Finally, a method is developed to extract lines from pages of handwritten text by finding the shortest spanning tree of a graph formed from the set of main strokes. Then main strokes of extracted lines are arranged in an order similar to their written order by following the path in which they are contained. Then, every secondary stroke is assigned to the closest main stroke. By the end, a list of main strokes with their relevant secondary strokes is achieved resulting in a combination of main-secondary strokes. Each element in the list can be the input to the classifier. Their method proved to be powerful and suitable for variable handwriting. Al-Badr and Hararlick (Al-Badr and Haralick 1995) introduced a holistic recognition system which recognizes Machine printed Arabic word without segmentation. Their system is based on describing the shape primitives as symbols. The instances of the predefined shape primitives are detected by applying the erosion operation on the word image. The system locates the best spatial arrangement of symbol models by applying a state space search. The detected primitives are matched with symbol models. The system was tested on a lexicon of 42000words, and the recognition rate achieved was 99.4% on noise free text and 73% for scanned text. Alma'adeed et al. {Alma'adeed, 2002 #84} introduced a system for classifying Arabic

handwritten words based on HMM. First, the word images were normalized by removing variations which did not affect the identity of the word. The normalization included stroke width, slope, and the letter height yielding a uniform height of one pixel wide stroke. Second, the skeleton of the image was constructed, and 29 features extracted. Finally, a classification process based on the HMM was used. Since there was no standard Arabic database, this system was tested on a special database (Al-Ma'adeed, Elliman et al. 2002) of 4700 handwritten words written by 100 writers. The recognition rate achieved was 45% because some words conflict with each other.

Alma'adeed et al. {Alma'adeed, 2004 #85} introduced a system for unconstrained Arabic handwritten word recognition based on multiple HMMs. First, pre-processing tasks were performed similar to the work in {Alma'adeed, 2002 #84}. In order to improve the recognition rate in {Alma'adeed, 2002 #84}, global features, such as numbers of upper dots, numbers of lower dots, and the numbers of segments, ascenders and descenders, were used to differentiate the words from each other. By using these features and the multiple HMM, in which each HMM used a different set of features the system removes all the variation in the images. Second, the skeleton of the image was performed, and 29 features were extracted. Finally, a classification process based on the HMM was used. This system was tested on a database (Al-Ma'adeed, Elliman et al. 2002) of 100 handwritten words written by 1000 writers. The recognition rate achieved was 60% before using post processing. The codebook size was chosen after testing and selected different words for each group. There were eight groups where the first group had 90 words, second had 100 words, the third had 80 words, the fourth had 90 words, and eighth had 120 words. The recognition rate was different for each group. The first group had a 97% recognition rate, while the eighth group had only 60% recognition rate. Alma'adeed (Alma'adeed 2006) introduced a system for unconstrained Arabic handwritten word recognition using a neural network classifier (NN). This system used the pre-processing and features in {Alma'adeed, 2002 #84; Alma'adeed, 2004 #85}. The NN had 8 neurons for the input layer, 40 neurons for the middle layer, and the output layer had 70 neurons since the NN classifier used 70 different words. The accuracy achieved was 63%.

Khorsheed and Clocksin (Khorsheed and Clocksin 1999) presented a technique for extracting the structural features from Arabic cursive text. Several pre-processing tasks were performed including: thinning based on Stentiford's algorithm (Parker 1997) and skeleton centroid calculation to find a reference point relative to all segment locations. The features were extracted in three steps. First, segment extraction was done using the skeleton graph of the word image which consists of a number of segments. There are feature points where a segment starts and ends. Second, loop extraction in which the loops are divided into three categories: a simple loop, a complex loop, and a double loop. The loops are checked during the segment extraction. Third, segment transformation is done after extracting the segment and the loops. The Viterbi algorithm {Rabiner, 1986 #88} is used to form a codebook by portioning the training samples into several classes, and the codebook includes 76 symbols. The technique was tested with a lexicon of 294 words acquired from a different text sources by using the HMM, and recognition rates of up to 97% were achieved.

Khorsheed and Clocksin (Khorsheed and Clocksin 2000) presented a holistic recognition system for recognizing Arabic cursive words. First, Fourier coefficients are extracted from a word image after converting it into a normalized polar image. Using the average coefficient values for sample training, each word was represented by template form. The recognition

was done by using word template with Euclidean distance and assigning the unknown word to the closest word template. The recognition rate achieved was over 90%. However, this system fails for many fonts. Khorsheed (Khorsheed 2003) presented another holistic recognition system for recognizing Arabic handwritten words. Pre-processing tasks performed included using the Zhang-Suen thinning algorithm {Zhang, 1984 #91} to generate the skeleton graph. Structural features for the handwritten script were extracted after skeletonization by decomposing the word skeleton into a sequence of links with an order similar to the word writing order. Using the line approximation (Parker 1997), each line was broken into small line segments, which were transferred into a sequence of discrete symbols by using vector quantization (VQ) {Gray, 1989 #92}. With this system, the HMM recognizer was applied with image skeletonization to the recognition of an old Arabic manuscript which can be found in (Khorsheed 2000). One HMM was performed from 32 character HMMs, each with no restriction jump margin. The system was tested on 405 character samples of a single font extracted from a single manuscript. The recognition rates achieved were 87% and 72% with and without spell checking respectively. Khorsheed (Khorsheed 2007) presented a recognition system based on the HMM to recognize Arabic text. Pre-processing was performed, using a slow median filter, to reduce salt and pepper noise. Statistical features were extracted from text image and fed to the recognizer. The recognizer was built on the HMM toolkit (HTK) (Young, Evermann et al. 2001). The advantage of this system is the lexicon free approach which offers open vocabulary recognition. The system was able to learn complicated ligatures and overlaps. Different text images with different fonts were tested, and the recognition rate achieved was up to 92.4%. A tri-model implementation showed a better system performance than a mono-model implementation.
In comparison with existing work, our proposed methods illustrate significant advantages, which can be highlighted as: (i) by using the knowledge of potential positions of the base line, an improved projection based method is employed for baseline detection; (ii) statistical analysis distribution of the word and sub-words distances is obtained to determine an optimal threshold for word segmentation; (iii) a component-based method for word segmentation is used to provide a practical way in accurately segmenting words from the text line instead of segmenting the words into characters, and using the segmentation free systems.

3. Proposed Methods

3.1 Baseline Detection

Previous work on baseline detection can be summarized as follows. Pechwitz and Margner (Pechwitz and Margner 2002) approximated the skeleton by a piecewise linear curve and detected the baseline as the line that best fits the edges. Farooq et al. (Farooq, Venu et al. 2005) used the IFN/ENIT database and entries into documents in order to simulate skew, line separation, and other features. Their method is based on the local minima points of words. Their method generally works well but fails to find the baseline in situations where the diacritics are large relative to the word. The removal of diacritics is suggested as a potential solution. Al-Rashaidehn {Al-Rashaideh, 2006 #110} found the baseline based on the assumptions that the baseline is rotated horizontally within a range of angles between +20° to -20° and keeping in mind that the maximum number of pixels is located along the baseline. M. Syiam et al. (Syiam, Nazmy et al. 2006) presented a complete Arabic OCR

system which uses a histogram clustering method for segmenting the Arabic word. In the present research, the baseline is detected by using a horizontal projection of input images. This is defined as the sum of foreground pixels perpendicular to the x axis and is represented by the vector $H(y)$ of size M. Let $p(x,y) \mid x \in [1,M], y \in [1,N]$ denote one input image, its horizontal projection is defined as follows:

$$H(y) = \sum_{x} p(x,y) \tag{1}$$

Where $H(y)$ denotes number of effective pixels when p is a binary image. Normally, the position of the baseline is indicated by a peak in $H(y)$. For most the cases, this simple rule works in determining the baseline. However, it fails in some cases as illustrated in Figure 2 where the global peak in $H(y)$ is not the baseline. To solve this problem, we apply

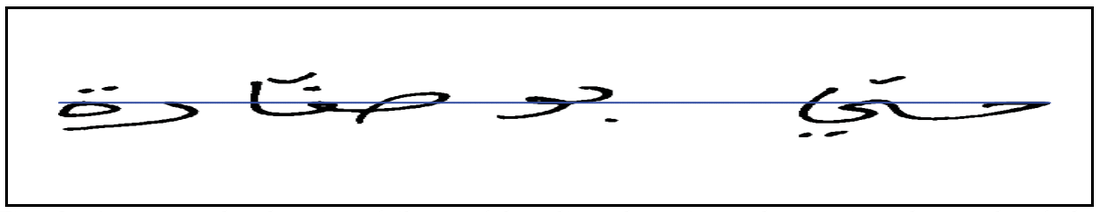

Fig. 2. An example showing failure of baseline detection when using the peak of the horizontal projection of the image

knowledge based constraints. We know that the baseline should appear below the middle line of the image. Therefore, we modify the algorithm to find the peak in $H(y)$ only in the bottom half of the images, i.e.

$$b = \arg\max_{y \in [N/4, N/2]} H(y) \tag{2}$$

With the modification applied, the corresponding baseline is successfully located as shown in Figure 3

Fig. 3. Detected baseline from the image in Figure 2 using the knowledge-based modified algorithm

3.2 Extracting connected components and sub-words

Segmentation is an essential step which separates the text image objects for the recognition phase. The typical segmentation of a binary document is based on the histogram projection analysis and regrouping of the connected components {Amin, 1998 #20; Lorigo, 2006 #2}.

Arabic writing is cursive such that words are separated by spaces. However, a word may contain several sub-words which are portions of the word including one or more connected letters. The connected components (CCs) for the line image must be determined. Each CC is enclosed in a minimum sized rectangular box. The objective of the CCs phase is to form rectangles around all the connected objects in the image. The algorithm used to obtain the CCs is an iterative procedure which checks any black pixels for connectivity with another. Bounding rectangles are extended to enclose any grouping of connected black pixels.

In our systems, the 8 – neighbours are used for extracting the connecting components by scanning the image pixel by pixel checking for pixel connectivity. In order for two pixels or more to be considered connected, the pixel values are in the same set V, V={1}. The 8 – neighbours are defined by

$$N_8(P) = N_4(P) \bigcup N_D(P) \tag{3}$$

Where N4(p)= { (x+1,y), (x-1,y), (x,y+1), (x,y-1)} and

$N_D(p)$ = { x+1,y+1), (x+1,y-1), (x-1,y+1), (x-1,y-1)}

Figure 4(a) shows the identified CCs for some example images. Starting from extracted connect components, sub-words are segmented as follows. Firstly, small parts like dots in the image are temporally ignored as shown in Figure 4(b). Secondly, components whose coordinates overlap in the *x* direction are merged to produce a combined large component, namely sub-word. Thirdly, the distance of each pair of consecutive sub-words is obtained, which is used to segment words in the next section

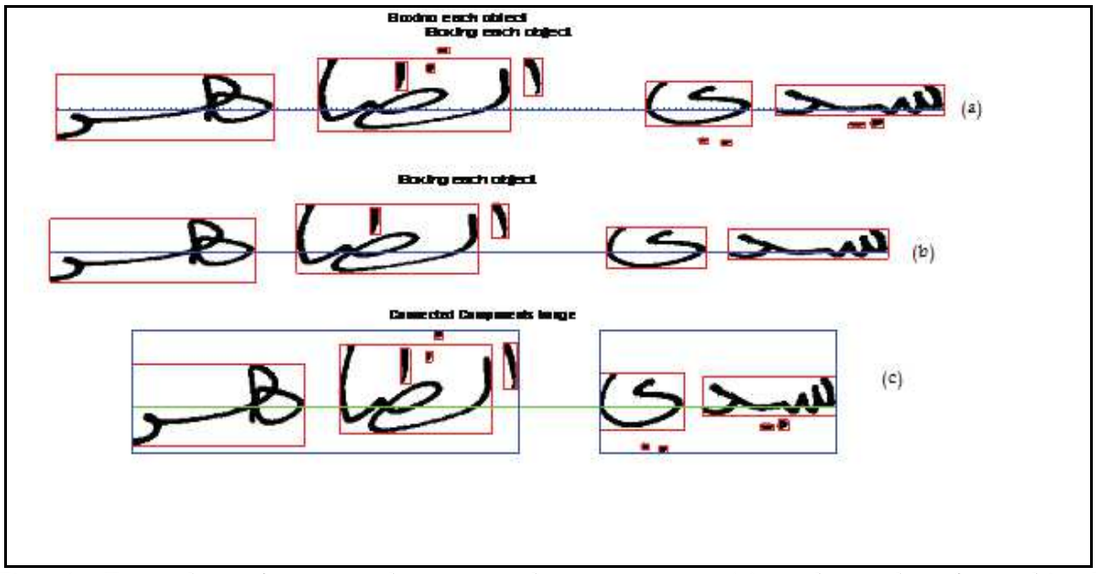

Fig. 4. Examples of extracted connected components (a), sub-words of combined components (b), and detected words (c).

3.3 Extracting connected components and sub-words

Basically, there are two categories of systems for the recognition of Arabic scripts: character-based and word-based systems. In the first category, words need to be further segmented into characters or letters and these characters are then used for recognition. The second category does not need such segmentation and whole words are used for recognition. In both categories, segmentation of words from the text is necessary.

Several algorithms have been presented for the segmentation of Latin cursive script. However, Arabic script segmentation has not received as much attention. In 1992, Amin and Al-Sadoun (Amin and Al-Sadoun 1992) proposed a segmentation technique for Arabic text using the binary tree. In 1995, AlBader and haralick (Al-Badr and Haralick 1995) presented a system which recognizes a machine printed Arabic word without prior segmentation. In 1997, Motawa et al (Motawa, Amin et al. 1997) presented an automatic segmentation of Arabic words using Mathematical Morphology tools. They applied their algorithm based on the assumptions that characters are usually connected by horizontal lines. In 2005, Lorigo and Govindaraju (Lorigo and Govindaraju 2005) presented a segmentation system which used derivative information in a region around the baseline to over segment the words.

Segmenting a line of text into words is known as word separation. In the machine printed case, word separation is easier than in the handwriting case because the space between words is uniform, and larger than the space between sub-words. In handwriting case, the space between words is not always uniform and moreover, the same amount of space may be present between the words and sub-words on a line.

In our system, each image is segmented into words using vertical histograms. Words have varying length; therefore after taking the vertical histogram as shown in Figure 5, the line can be classified into words and sub-words depending on distances between groups of peaks along the x axis.

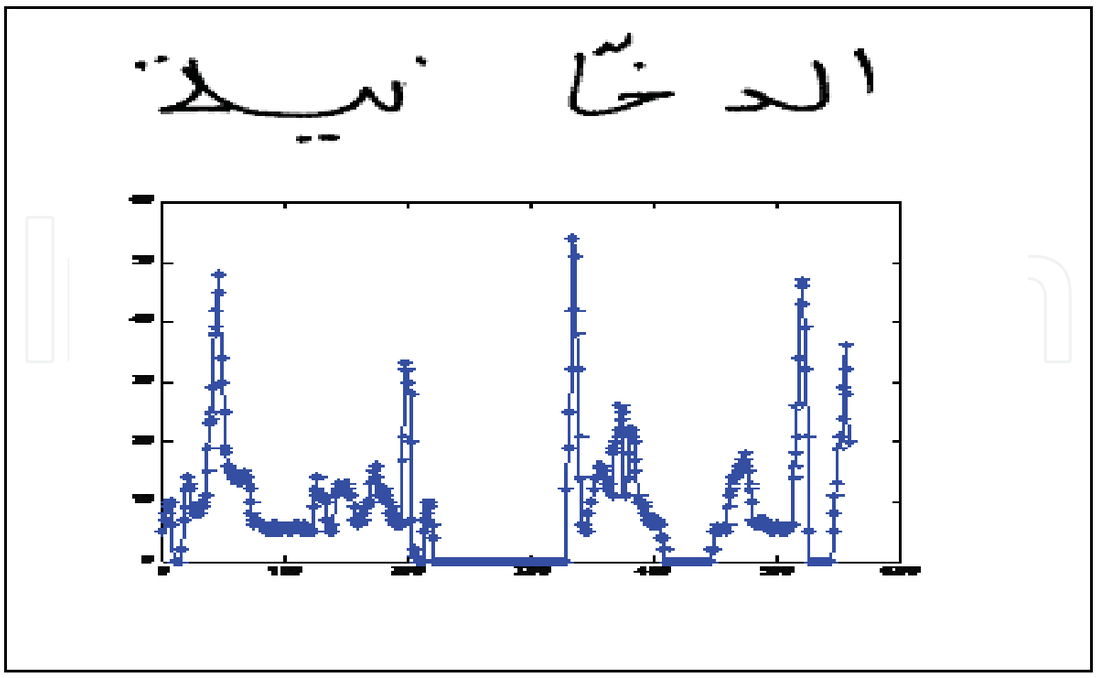

Fig. 5. Vertical Histogram

The vertical projection defined as the sum of foreground pixels perpendicular to the y axis; this is represented by the vector v_j of size N defined by

$$v(j) = \sum_{j} p(i, j) \tag{4}$$

where $p(i, j)$ is a pixel of the binary image of the script and is either 0 or 1, i refers to rows and j refers to columns.

Arabic writing is cursive; therefore, words and sub-words are separated by spaces, so word boundaries are always represented by a space. However, six letters can be connected from the right side only. Using this knowledge and the vertical histogram, spaces can be detected by calculating the zero distance (gaps) on the x axis as shown in Figure 5. The distances between words are generally larger than the distances between sub-words. This distance is used to decide the number of word(s) in the image based on a threshold.
To determine a suitable threshold, the Bayesian criterion, of minimum classification error, is employed as follows. Given a distance d, the probabilities that represent separation of words or sub-words are denoted as $p_w(d)$ and $p_{s-w}(d)$, respectively. These two conditional probabilities were obtained by manually analyzing over 100 images containing more than 250 words. Taking $p_w(d)$ for example, we found all possible distances separating a word, calculated their histogram and estimated $p_w(d)$ from this histogram. Illustrations of both $p_w(d)$ and $p_{s-w}(d)$ are given in Figure 6.

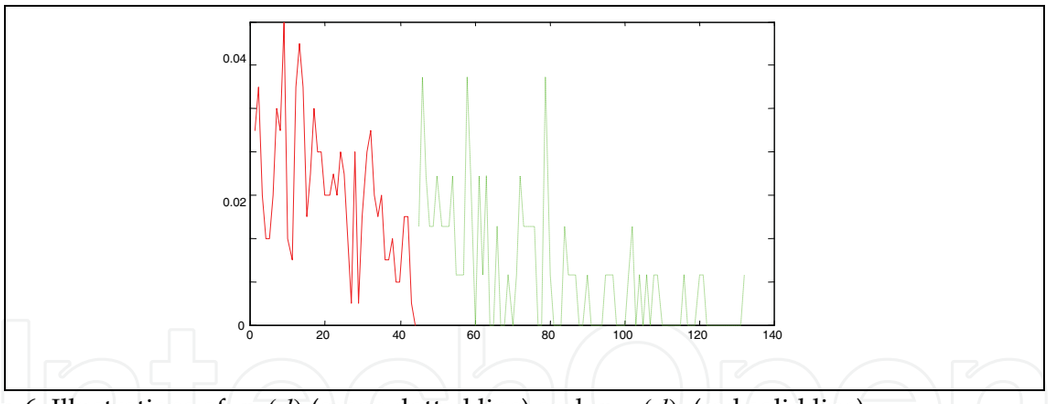

Fig. 6. Illustrations of $p_w(d)$ (green dotted line) and $p_{s-w}(d)$ (red solid line).

Finally, an optimal distance d_0 is obtained under the Bayesian minimum classification error criteria:

$$d_0 = \arg\min_{d}(err(d)) \tag{5}$$

$$err(d) = \int_{d}^{\infty} p_{s-w}(x)dx + \int_{0}^{d} p_w(x)dx) \tag{6}$$

The segmentation of words is completed by simply comparing the distance d with this optimal distance or threshold d_0. The case $d > d_0$, identifies two words and the alternative case identifies two sub-words.

4. Experimental Results and Discussions

Text recognition systems can be classified into two areas, printed text or handwritten text. Printed texts have similar shapes if printed many times using different devices, however, due to different writer styles, handwritten texts have high variability. The main goal of a handwriting recognition system is to determine the class of the character or word. It is a more difficult task to design a recognition system which can recognize the handwriting of many people instead of just the handwriting of a single writer. Also, in order to evaluate handwriting recognition systems, the accuracy and the speed have to be measured and compared to those of an average of human reader. In the literature, some recognition systems were reported with high recognition rates. This is generally due to their testing data which consisted of a small set of words written by few writers, rather than a standard database. Any recognition system needs a large database to train and test the system. Real data from banks or the post code are confidential and inaccessible for non commercial research. Although some work has been conducted on Arabic handwritten words, this generally used the authors' own small databases or databases which were unavailable to the public. Most recognition systems have been developed for certain applications such as the reading of postal addresses or cheques. An example of a large standard English database suitable for the development and training and testing of recognition systems is the one created 14 years ago by Hull (Hull 1994). This was developed for the centre of Excellence for Document Analysis and Recognition (CEDAR) at the State University of New York at Buffalo and consists of 5000 city names, 5000 state names, 10000 ZIP codes, and 50000 alphanumeric characters. The National Institute of Standards and Technology (NIST) has also provided a handwritten database which includes English letters in lower and upper cases, number digits and the computer and Communication Research Laboratory of the Industrial Technology Research Institute in Taiwan have released a handwritten Chinese characters database written by 2000 writers {Huang, 1993 #106}.

The data set for the experiments is the IFN/ENIT database. Pechwitz et al. (Pechwitz, Maddouri et al. 2002) released the

Any recognition system needs a large database to train and test the system. Real data from banks or the post code are confidential and inaccessible to non commercial research. Early work conducted using Arabic handwritten words, generally used small individual databases or presented results on databases which were unavailable to the public. Consequently, there was no benchmark to compare the results obtained by researches. This situation changed in 2002 when the IFN/ENIT database (www.ifnenit.com) became available free for non commercial research. The IFN/ENIT database, is very important in this context and has been used as a standard test set [9]. In total more than 1000 different people were selected to write their names and to fill in one or more forms with handwritten pre-selected names of Tunisian town/villages and the corresponding postcode. All the forms were scanned at 300 dpi and converted to binary images.

The database consists of 937 Tunisian town/villages names together with their postcodes. In total more than 1000 different writers were used. Each writer was asked to fill in one or more forms with handwritten pre-selected names of Tunisian town/villages and the corresponding postcode. All the forms were scanned at 300dpi and converted to binary images. The images are divided into five sets so that researchers can use some of them for training and some for testing. Some pre-processing tasks including noise removal, text block segmentation, binarization and word segmentation have been done during the development

of the IFN/ENIT database to make cropped binary images of the names of towns and villages available.
Corresponding to the test data sets as described above, we design two phases of experiments to evaluate the proposed algorithms, which include: Phase-1: experiments to evaluate performances of baseline estimation; and Phase-2: experiments to evaluate performances of connected component analysis and the performances of word segmentation;
In phase-1, our experiments focus on the baseline estimation. Due to the fact that the baseline is a part of the ground truth of the IFN/ENIT database, so it is possible to evaluate the baseline. Figure 7 shows the phase-1 experimental results on baseline estimation, from which it can be seen that the proposed algorithm worked well when applied to 4000 images using the four different sets (the first 1000 image form sets a,b,c,and d was selected). The results for baseline estimation reach 97.675% of accuracy, which makes the proposed algorithm more effective in estimating a word baseline. Table 2 summarizes the experimental results for the baseline estimation proposed algorithm.

Set	a	b	c	d	average
Percentage (%)	97.4	97.8	97.9	97.6	97.675

Table 2. Performance of the baseline estimation algorithm

In comparison with the existing work, our baseline algorithm performs better in estimating the baseline. Table 3 summarizes the results of our algorithm compared to the results of existing work.

Method	Hough Projection [31]	Skeleton Based [31]	Proposed Algorithm
Percentage (%)	88	88	88.9

Table 3. Performance of proposed algorithm vs. other methods

In general, calculating the baseline error is used to estimate the baseline quality. The error is calculated as the area between the ground truth baseline and the estimated baseline in pixels. Figure 8, shows an example of calculating the baseline error, while Figure 9 shows the relation between the estimated baseline and the ground truth baseline.

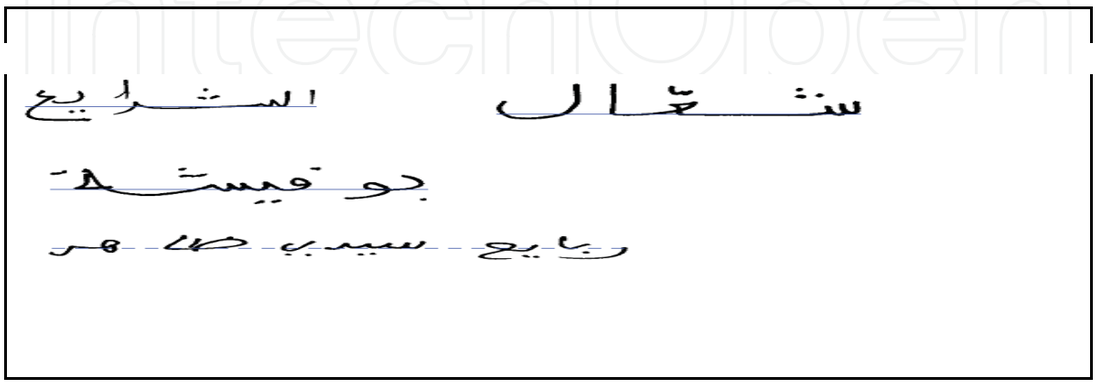

Fig. 7. Example baseline estimation results

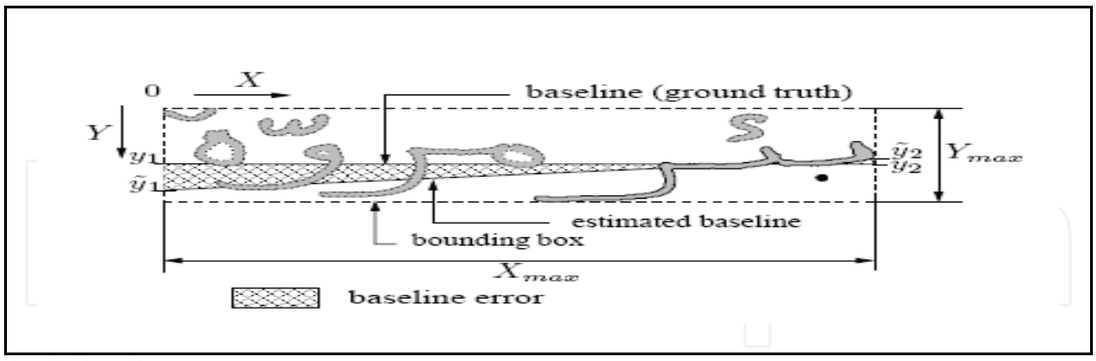

Fig. 8. Baseline error

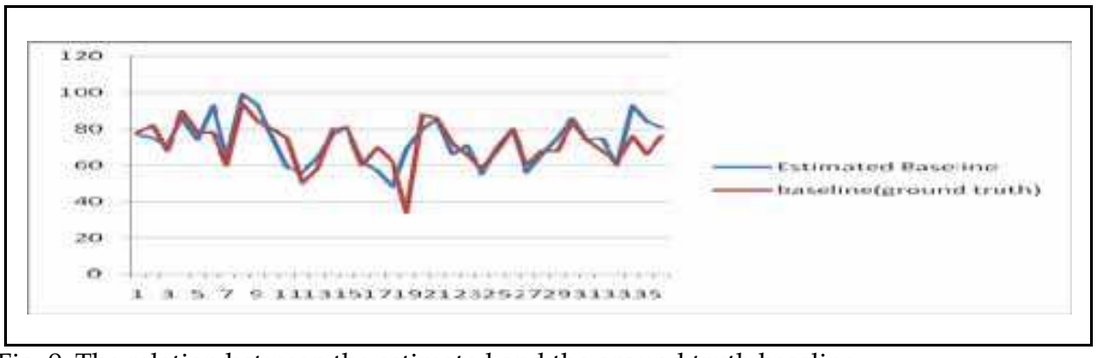

Fig. 9. The relation between the estimated and the ground truth baseline

To complete phase-2 experiments, vertical histogram and connected component analysis are carried out for word segmentation. Word segmentation approaches are based on the assumption that the text lines are straight. This works well for machine printed documents, but it fails on the handwritten documents having curvilinear text lines. Here, the distances between sub-words are measured and compared to an optimal threshold to determine if the distance corresponds to separation of two words or not. The segmentation algorithm searches for horizontal gaps between the connected components on a pre-estimated threshold. In comparison with the existing work, our word segmentation algorithm illustrates significant advantage, which can be highlighted as: in the case of miss-spaced words, where the algorithm failed to determine bounding boxes, spaces were automatically adjusted not adjusted manually using graphical tools.

In general, there are several types of error that occur during the process of segmentation whatever the approach used. These errors can be summarized as:

1) Over segmentation, when the number of segments is greater than the actual number.
2) Under segmentation when the number of segments is less than the actual number.
3) Misplaced segmentation when the number of segments is right but the limits are wrong.

We have tested our techniques on a test set of 500 images and the results are compared to the ground truth based on the grouping of the bounding boxes into words. Table 4 summarizes the word segmentation results; some of the results are presented in Figure 10.

From Table 4 we can see that the correct segmentation rate achieved for images is 85%. The segmentation error of 15% is due to the variations in handwriting, especially irregular spaces between sub-words and words, such as too small spaces between words (which will lead under segmentation by incorrectly merging two words together) or too large spaces between sub-words (which may be wrongly taken as two words and lead to over-segmentation). Examples of these errors are illustrated in Figure 11.

In comparison with the existing work, it is difficult to compare our work to [35] since they have used some other criteria and they have chosen 200 images. They did not mention which 200 images of the database which make our algorithm can not be implemented to the same data. Moreover, in the case of miss-spaced words, where the algorithm failed to determine the bounding boxes, our algorithm perform better since it reduces the numbers of such errors. The distance between words and sub-words were automatically normalized by using knowledge of the Arabic language not adjusted manually using the graphical tools, in which the word case can be determined. For example, the word in Figure 11 (a) was over segmented, but after distance normalization the word image is now segmented correctly as shown in Figure 12.

In addition, the rules of Arabic Language writing can be exploited and applied to the distance normalization. The original image is scanned from right to left column by column, and the white (blank) columns are detected and adjusted in size in order to reduce the distances between sub-words as Illustration of distance normalization is given in Figure 12 show. After applying the distance normalization, the Arabic words are correctly segmented. Since each handwritten image has Ground Truth (GT) information for evaluation purposes, the results are compared with the IFN/ENIT GT information.

No. of Images	Correct segmentation	Under segmentation	Over segmentation	Misplaced segmentation
500	85%	9%	4%	2%

Table 4. Overall segmentation results

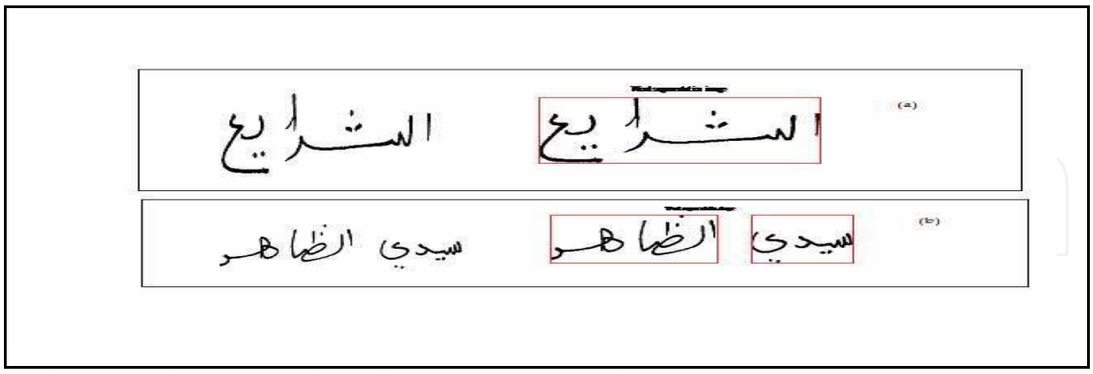

Fig. 10. Example successful word segmentation results

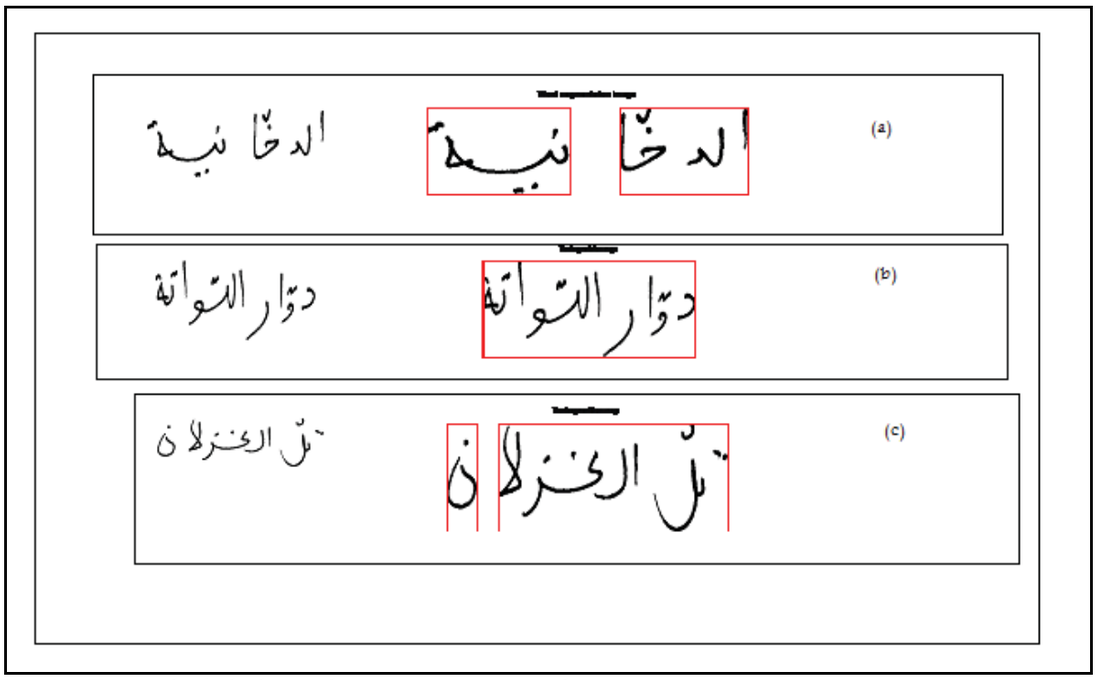

Fig. 11. Example failed word segmentation results

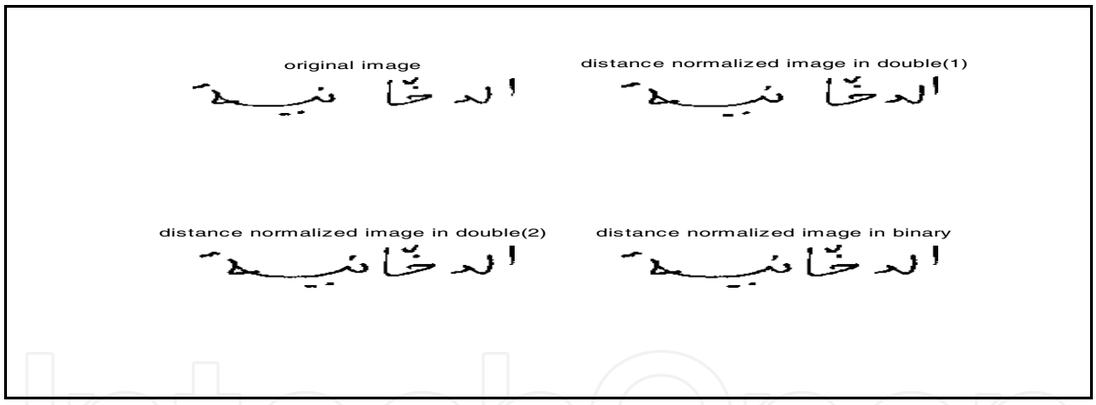

Fig. 12. Example failed word segmentation results

5. Conclusion

Arabic handwriting recognition depends on accurate pre-processing and segmentation. This chapter proposes a robust method for baseline estimation and a statistical analysis to determine an optimal threshold for word segmentation. By using knowledge of potential positions of the baseline, more accurate results are obtained in comparison with those without knowledge support. In addition, the optimal threshold obtained is found to be very effective for robustly segmenting words in Arabic text.

A component-based method is introduced to segment words from handwritten Arabic texts. Since many people have emphasized either segment-free based methods or letter or stroke based approaches, words segmentation has not be well addressed. Here, our work provides

a practical way of accurately segmenting words from the text. This is useful and more flexible than segment-free based approaches as it can make good use the component parts of images in further recognition. Also, this approach is simpler and more robust than letter-based methods because the letter has much difficulty in effectively segmenting arbitrary handwritten characters. We have found that distance information is very useful for segmenting words, but improvements are still desirable. A distance normalization technique making use of knowledge of the language was applied to reduce the numbers of over and under segmentation errors. Further investigations will aim to further improve word segmentation by using language knowledge for validation.

6. References

[Abuhaiba, I. S. I., M. J. J. Holt, et al. (1996). "Processing of binary images of handwritten text documents." Pattern Recognition 29(7): 1161-1177.

Al-Badr, B. and R. M. Haralick (1995). Segmentation-free word recognition with application to Arabic. Proceedings of the Third International Conference on Document Analysis and Recognition.

Al-Ma'adeed, S., D. Elliman, et al. (2002). A data base for Arabic handwritten text recognition research. Eighth International Workshop on Frontiers in Handwriting Recognition

Al-Rashaideh, H. (2006). "Preprocessing phase for Arabic Word Handwritten Recognition." Information Transmissions in Computer Networks 6: 11-19.

Alma'adeed, S. (2006). Recognition of Off-Line Handwritten Arabic Words Using Neural Network. Geometric Modeling and Imaging--New Trends

Alma'adeed, S., C. Higgens, et al. (2002). "Recognition of Off-Line Handwritten Arabic Words Using Hidden Markov Model Approach " 16th International Conference on Pattern Recognition (ICPR'02) 3: 481-484.

Alma'adeed, S., C. Higgins, et al. (2004). "Off-line recognition of handwritten Arabic words using multiple hidden Markov models." Knowledge-Based Systems 17(2-4): 75-79.

Almuallim, H. and S. Yamaguchi (1987). "A method of recognition of Arabic cursive handwriting." IEEE Transactions on Pattern Analysis and Machine Intelligence 9(5): 715 - 722

Amin, A. (1997). Off line Arabic character recognition: a survey. Fourth International Conference on Document Analysis and Recognition.

Amin, A. (1998). "Off-line Arabic character recognition: the state of the art." Pattern Recognition 31(5): 517-530.

Amin, A. (2000). "Recognition of printed arabic text based on global features and decision tree learning techniques." Pattern Recognition 33(8): 1309-1323.

Amin, A., H. Al-Sadoun, et al. (1996). "Hand-printed arabic character recognition system using an artificial network." Pattern Recognition 29(4): 663-675.

Amin, A. and H. B. Al-Sadoun (1992). A new segmentation technique of Arabic text. Proceedings., 11th IAPR International Conference on Pattern Recognition, 1992. Vol.II. Conference B: Pattern Recognition Methodology and Systems, .

Farooq, F., G. Venu, et al. (2005). Pre-processing methods for handwritten Arabic documents. Proceedings Eighth International Conference on Document Analysis and Recognition.

Freeman, H. (1961). "On the encoding of arbitrary geometric configuration." IEEE Trans. Electronic Computer 10: 260-268.

Gray, R. M. (1989). "vector quantization." IEEE Trans. ASSP(1): 4-29.

Guo, Z. and R. W. Hall (1989). "Parallel thinning with two-subiteration algorithms." Communications of the ACM 32(3): 359 - 373

Huang, J. S. (1993). Optical handwritten Chinese character recognition. HANDBOOK OF PATTERN RECOGNITION AND COMPUTER VISION World Scientific Publishing Co., Inc: 595-624.

Hull, J. J. (1994). "A database for handwritten text recognition research." Pattern Analysis and Machine Intelligence, IEEE Transactions on 16(5): 550-554.

Khorsheed, M. S. (2000). Automatic Recognition of Words in Arabic Manuscripts Computer Laboratory, University of Cambridge. P.h.D: 220.

Khorsheed, M. S. (2002). "Off-Line Arabic Character Recognition – A Review " Pattern Analysis & Applications 5(Volume 5, Number 1 / May, 2002): 31-45.

Khorsheed, M. S. (2003). "Recognising handwritten Arabic manuscripts using a single hidden Markov model." Pattern Recognition Letters 24(14): 2235-2242.

Khorsheed, M. S. (2007). "Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK)." Pattern Recognition Letters 28(12): 1563-1571.

Khorsheed, M. S. and W. F. Clocksin (1999). Structural Features Of Cursive Arabic Script. the Tenth British Machine Vision Conference, The unversity of Nottingham, UK.

Khorsheed, M. S. and W. F. Clocksin (2000). Multi-font Arabic word recognition using spectral features. Proceedings 15th International Conference on Pattern Recognition, 2000. .

Lorigo, L. and V. Govindaraju (2005). Segmentation and pre-recognition of Arabic handwriting. Eighth International Conference on Document Analysis and Recognition. .

Lorigo, L. M. and V. Govindaraju (2006). "Offline Arabic handwriting recognition: a survey." Pattern Analysis and Machine Intelligence, IEEE Transactions on 28(5): 712-724.

Motawa, D., A. Amin, et al. (1997). Segmentation of Arabic cursive script. The Fourth International Conference on Document Analysis and Recognition.

Parker, J. R. (1997). Algorithms For Image Processing and Computer Vision John Wiley and Sons, Inc

Pechwitz, M., S. S. Maddouri, et al. (2002). IFN/ENIT - Database of Arabic Handwritten words. Colloque International Franco-phone sur l'Ecrit et le Document (CIFED).

Pechwitz, M. and V. Margner (2002). Baseline estimation for Arabic handwritten words. Eighth International Workshop on Frontiers in Handwriting Recognition

Rabiner, L. and B. Juang (1986). "An introduction to hidden Markov models." ASSP Magazine, IEEE [see also IEEE Signal Processing Magazine] 3(1): 4-16.

Syiam, M., T. M. Nazmy, et al. (2006). Histogram clustering and hybrid classifier for handwritten Arabic characters recognition. Proceedings of the 24th IASTED international conference on Signal processing, pattern recognition, and applications

Young, S., G. Evermann, et al. (2001). The HTK Book, Cambridge University Engineering Department.

Zhang, T. Y. and C. Y. Suen (1984). "A fast parallel algorithm for thinning digital patterns." Communications of the ACM 27(3): 236 - 239

IntechOpen

Recent Advances in Technologies

Edited by Maurizio A Strangio

ISBN 978-953-307-017-9

Hard cover, 636 pages

Publisher InTech

Published online 01, November, 2009

Published in print edition November, 2009

The techniques of computer modelling and simulation are increasingly important in many fields of science since they allow quantitative examination and evaluation of the most complex hypothesis. Furthermore, by taking advantage of the enormous amount of computational resources available on modern computers scientists are able to suggest scenarios and results that are more significant than ever. This book brings together recent work describing novel and advanced modelling and analysis techniques applied to many different research areas.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jawad H AlKhateeb, Jianmin Jiang, Jinchang Ren and Stan Ipson (2009). Interactive Knowledge Discovery for Baseline Estimation and Word Segmentation in Handwritten Arabic Text, Recent Advances in Technologies, Maurizio A Strangio (Ed.), ISBN: 978-953-307-017-9, InTech, Available from: http://www.intechopen.com/books/recent-advances-in-technologies/interactive-knowledge-discovery-for-baseline-estimation-and-word-segmentation-in-handwritten-arabic-

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen