Multi-Oriented Text Line Extraction from Handwritten Arabic Documents

Nazih Ouwayed, Abdel Belaïd

Multi-Oriented Text Line Extraction from Handwritten Arabic Documents

Nazih OUWAYED, Abdel BELAÏD
LORIA, University Nancy 2
Vandoeuvre-Lès-Nancy, France
{nazih.ouwayed,abelaid}@loria.fr

Abstract

In this paper, we present a novel approach for the multi-oriented text line extraction from handwritten Arabic documents. After image pre-processing, the local orientations are determined in small windows obtained by image paving. The orientation of the text within each window is estimated using the projection profile technique considering several projection angles. Then, the windows which close angles are gathered into largest zones. We use the Wigner-Ville Distribution (WVD) to estimate the global orientation of each zone. The WVD is more precise than the classical projection profile technique. Afterwards, the text lines are extracted in each zone basing on the follow-up of the baselines and the proximity of connected components. The experimental results prove the efficiency of the proposed scheme. It has been evaluated on 50 documents reaching an accuracy of about 97.6%.

1. Introduction

The text line segmentation is seen as a required step in document analysis field before word recognition. The difficulty of this task comes from the characteristics of handwritten documents especially where they are ancient (see Figure 1). These documents present irregular spacing between the text lines. The lines can overlap or touch when their poles and jambs regions belong to each other. Furthermore, lines in the form of annotations can hold in the margins. These lines are in general oblique because of the space reduction which constitutes new orientations. In Figure 1, there are 4 orientations for the annotations lines. The massive presence of the diacritical symbols complicates in more this task. Also, the Arabic handwritten writing presents great variations, in the forms of the letters or the words, and in page-setting.

The paper is organized as follows. In section 2, we will see the related work on text line extraction. The different steps of our multi-oriented text line extraction algorithm are detailed in section 3. We present in section 4 some experimental results and a conclusion.

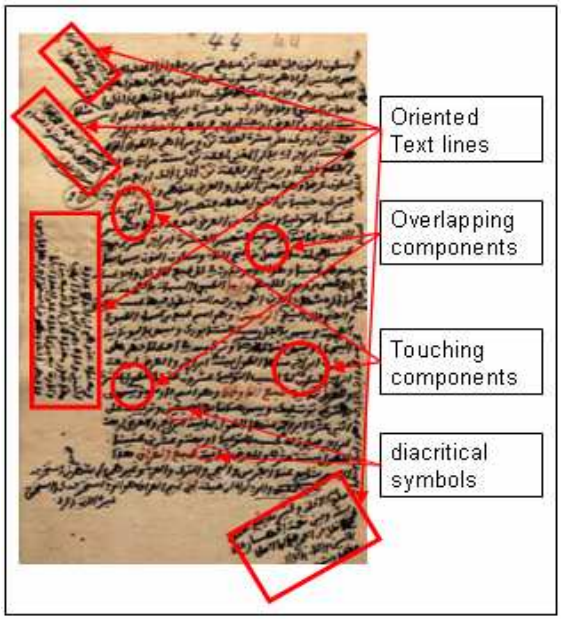


Figure 1. Characteristics of Arabic handwritten document.

2. Related Work on Text Line Extraction

In the literature, two classes of approaches exist for text line extraction: descendent and ascendant.

The descendent approaches are essentially based on the projection profile technique. In [5, 11, 16], the profile projection is applied to the whole document or a strip (horizontal or vertical) of it. Afterwards, the maxima and minima are determined and the connected components between two consecutive minima are

looked for. These connected components form the text line.

The ascendant approaches are based on the low level (pixel or connected component). In this category, we find the nearest neighbor clustering technique, the Hough transform technique, the smearing technique, and the Repulsive-Attractive technique. In the nearest neighbor clustering technique [9], first, the alignments are detected by selecting connected components elongated in specific directions. Then the text lines are extracted by grouping these alignments using three criteria: proximity, similarity and direction continuity. In the Hough transform technique [8], the voting points in the document are used to detect the text lines. Here the voting points are the centroids of the connected components. A set of aligned points in the image having a peak in the Hough transform, represents a text line. In the smearing technique [15], the Run-Length Smoothing is used. Consecutive black pixels along the horizontal direction are smeared. The bounding boxes of the connected components in the smeared image enclose text lines. In the Repulsive-Attractive technique [13], the baselines that represent the attractive forces, are constructed one by one from the top of the image to the bottom. In the image the pixels represent the repulsive forces. The attractive-repulsive forces between two consecutive lines are applied to determine the text lines.

Other approaches also use the special characteristics of the alphabet of each language. In [10], the head line of word is used to extract the multi-oriented and curved text lines from Indian documents.

All above approaches are applicable to Latin, Chinese and Indian scripts and have been developed for single skewed text lines. Few were used to the ancient handwritten Arabic scripts and multi-oriented documents. Thus, a novel approach to extract the multi-oriented text lines from the handwritten Arabic documents has to be developed. The method we present in this paper use the global paving, the multi-skew detection and the local text line extraction.

3. Multi-Oriented Text Line Extraction Algorithm

The writing in the historical handwritten Arabic documents has multi-orientation, is distorted and noisy. For this reasons, we decided to focus on a local method that removes at first the background of the image. Then, it detects the multi-skewed zones [1, 2, 3, 14]. After then, it follows the baselines to extract the text lines.

3.1. Pre-processing

To facilitate the segmentation, we need to remove the background (which takes a particular colour due to the aging of paper (see Figure 16 (a))). We have applied the "OTSU" algorithm [12] with an appropriate threshold of binarization. The diacritical symbols that have a small surface are removed in a second step, in order to lighten the image and to avoid creating local maxima in the projection histogram (see Figure 16 (b)).

3.2. Paving

In this step, the document is partitioned in windows (see Figure 16 (c)) with the same size (w×h), where w and h are calculated automatically according to the type of scripture in the document. To calculate w and h, firstly, the connected components are labelled and boxed. Then, the average width *avgw* (resp. height *avgh*) of all components is determined, as well as the average of the vertical gaps *gapv* (resp. horizontal gaps *gaph*) between each component having a width (resp. height) greater or equal to *avgw* (resp. *avgh*), and its 4 neighbours (Previous, Next, Below, Above). Thus, w=3×*avgw*+2×*gapw* and h=3×*avgh*+2×*gaph*, where 3 and 2 are thresholds representing respectively the number of components in Arabic word and gaps in 3 successive lines. They are obtained experimentally.

3.3. Insignificant Window Detection

In order to reduce the running speedup, we discard windows containing few pixels (named insignificant (empty or almost empty)) because their inclination is insignificant. When this kind of window contains connected components, we assign them to the neighbour windows sharing most of its components.

A connected component C_i, which centroïd (C_{gi}) is represented by (x_{ci}, y_{ci}), is up to a window (w) if ($x_w \le x_{ci} \le x_w + w$ & $y_w \le y_{ci} \le y_w + h$) where ($x_w, y_w$), are the upper-left coordinates of w.

For example, in the Figure 2 (a), the connected component C_1 (resp. C_2) cross the windows B and D (resp. A, B and C) but its centroïd C_{g1} (resp. C_{g2}) is up to one window A (resp. B). After the replacement of the connected components (A and B) to their appropriate windows, we obtain two empty windows C and D (see Figure 2 (b)).

In the Figure 3 (a), the connected component C_1 is replaced in the window B. Thus, the window A becomes almost empty window (it has one connected

component with few pixels). It is merged with the window B (see Figure 3 (c)).

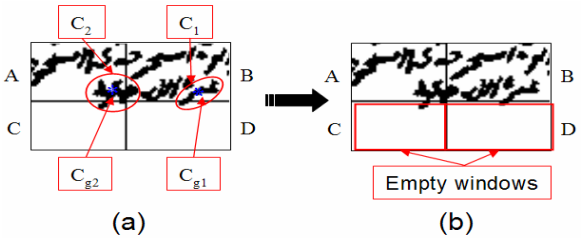


Figure 2. Example of empty windows.

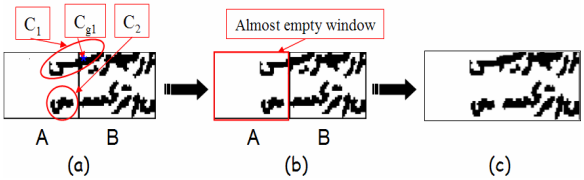


Figure 3. Example of almost empty window.

3.4. Orientation Detection

This operation needs two steps as follows:

First step:

The orientation in each window is estimated by analysing the maxima of the projection profile considering several angles varying by +15° from -75° to +90° (see Figure 16 (d)).

Second step:

The windows are enlarged to their neighbours in order to obtain homogeneous and bigger orientation areas. This extension is possible under some conditions relatively to the homogeneity of the orientation in the neighbour windows. Then, a global orientation angle is calculated for the large area. Usually, this just needs to recalculate the orientation angle for the total zone. However, we remarked that the projection profile technique becomes less efficient on large zones (see Figure 8). Thus, we proposed another method based on the energy distribution as defined by Wigner-Ville Distribution (WVD).

The energy distributions represent the energy of the signal over two description variables: time and frequency. These distributions decompose the signal on the basis of elementary signals (i.e. the atoms), which have to be well localized in both time and frequency. We know that the Cohen's class [4, 6] is the class of all the time-frequency energy distributions. The Wigner-Ville distribution is the simplest member of Cohen's class. The spectrogram of the WVD is a representation mechanism of the Cohen's class. It represents the energy distribution of the atoms (see Figure 6 where each signal corresponds to a projection profile along an angle) and it is used in different domains.

WVD is defined as:

$$W_x(t,v) = \int_{-\infty}^{+\infty} x(t+\tau/2)x^*(t-\tau/2)e^{-j2\pi v\tau}\,d\tau \qquad (1)$$

which represents the Fourier Transform of the signal:

$$x(t+\tau/2)x^*(t-\tau/2) \qquad (2)$$

where *x(t)* represents the analytical signal. The WVD is always real valued:

$$\int\int_{-\infty}^{+\infty} W_x(t,v)\,dtdv = \int_{-\infty}^{+\infty} |x(t)|^2\,dt \qquad (3)$$

It preserves time and frequency shifts and satisfies the marginal properties:

$$\int_{-\infty}^{+\infty} W_x(t,v)\,dv = |x(t)|^2$$
$$\int_{-\infty}^{+\infty} W_x(t,v)\,dt = |X(v)|^2 \qquad (4)$$

where $X|(v)|$ is the Fourier Transform of *x(t)*.

The WVD indicates the most intent alternations between peaks and dips for the projection profile corresponding to the right angle. Also, these alternations are represented in the frequency domain by big white points in the corresponding spectrogram (see Figure 6).

To estimate the skew angle of the zone, we project the zone at multiple selected angles. Then, the WVD is applied to each signal *x(t)* (projection profile (see Figure 5)) to determine the energy intensity. The angle corresponding to the projection profile with highest energy intensity is chosen as the skew angle of the zone (see Figure 7).

For example in Figure 9, we see the different values of energy for the zone in Figure 6. The angle estimated for it is +15° where the histogram of WVD has a maximum value.

In [7] E. Kavallieratou et al. used the WVD to estimate the angle of the document for Latin printed and handwritten.

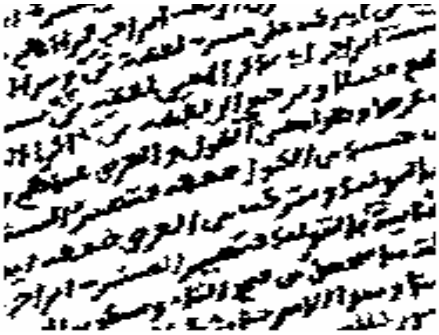


Figure 4. Skewed zone.

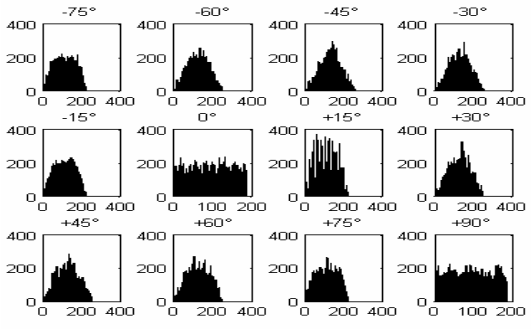


Figure 5. Projection profile (x(t)) as defined angles of the zone in Figure 4.

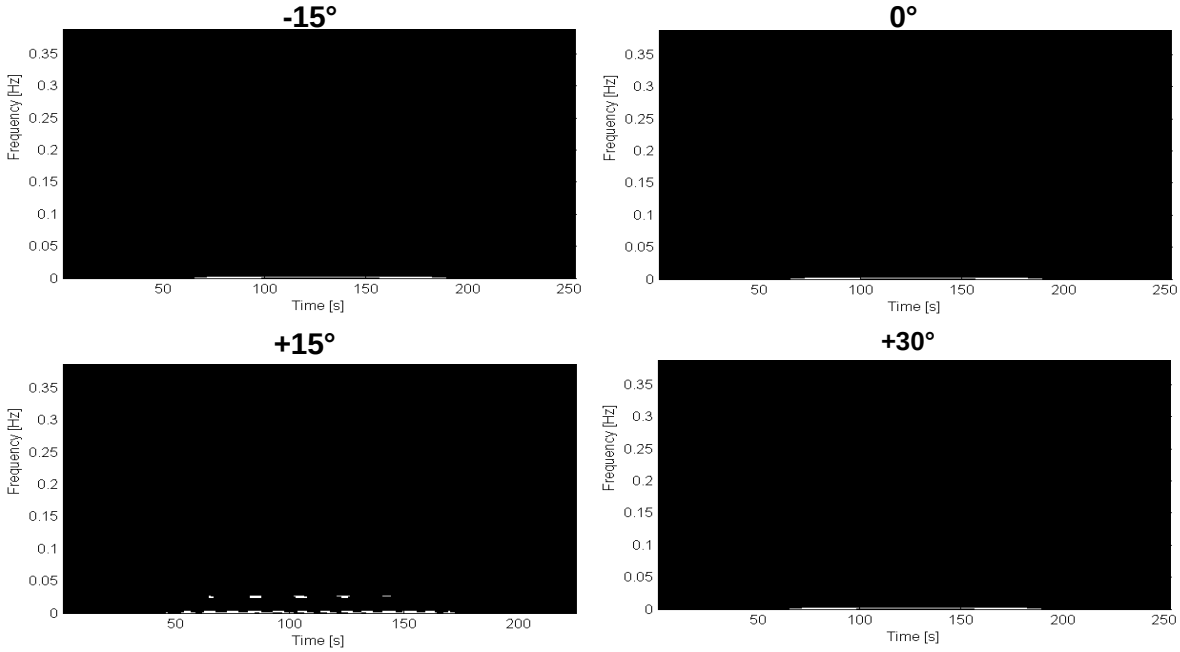


Figure 6. Spectrograms of the WVD for the zone in the Figure 4 at 4 different angles (-15°, 0°, +15°, +30°).

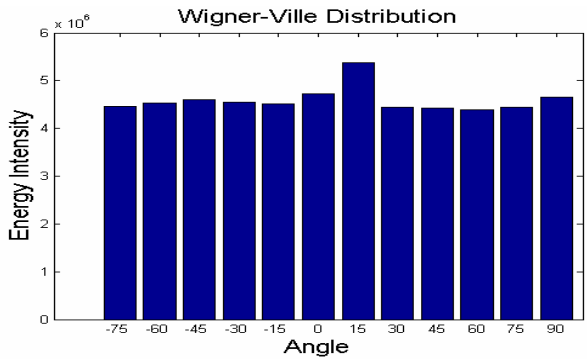


Figure 7. Distribution of the energy values of the WVD of the zone in Figure 4.

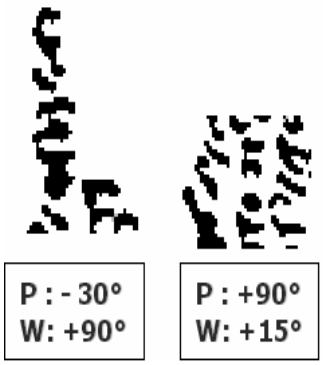


Figure 8. Difference between the projection profile (P) and the WVD (W) according to the window size.

3.5. Extension to Zone Detection

In order to detect the orientation zones, we develop an algorithm based on the nearest-neighbor principle. The fact that the Arabic is written from right to left was taken into consideration. We go from the window in the top-right until the window in the bottom-left of the document. For each window, the orientation of its neighboring windows is checked and zones constructed in accordance to 4 cases listed in Figure 9. In all cases, the departure is the dark window.

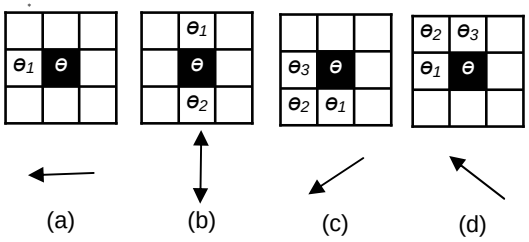


Figure 9. The 4 cases of the extension algorithm.

Case (a): Horizontal writing in Arabic

This direction is frequent for normal writing. If the departure window $w(\theta)$ has θ=0° and the neighbour window in the same direction $w(\theta_1)$ has θ_1=±15°, the two windows are merged in one window which orientation is performed by WVD. If the orientation obtained has a value equal to 0° thus θ_1=0°, else θ_1 is remained equal to ±15°.

Case (b): Vertical writing in Arabic

This direction arises for comments apposed in narrow spaces (margins). If the departure window has θ=+90° and the neighbour window in the same direction θ_1=±75° (resp. θ_2=±75°), the two windows $w(\theta)$ and $w(\theta_1)$ are merged (resp. $w(\theta)$ and $w(\theta_2)$) in one window which orientation is performed by WVD. If the orientation obtained has a value equal to +90° thus θ_1=+90° (resp. θ_2=+90°), else θ_1 and θ_2 are remained equal to their initial values.

Case (c): Negative diagonal writing in Arabic

If the departure window has +15°≤θ≤+75° and the neighbors windows in that direction have +15°≤θ_1, θ_2, θ_3≤+75°, the 4 windows $w(\theta)$, $w(\theta_1)$, $w(\theta_2)$ and $w(\theta_3)$ are merged in one window and its orientation θ_r performed. If θ_r has a value between +15° and +75° thus θ= θ_r, θ_1=θ_r, θ_2=θ_r and θ_3=θ_r, else we keep the values of θ, θ_1, θ_2 and θ_3.

Case (d): Positive diagonal writing in Arabic

In this case, if the departure window has a -75°≤θ≤-15° and the neighbour windows have -75°≤θ_1, θ_2 and θ_3≤-15°. These 4 windows $w(\theta)$, $w(\theta_1)$, $w(\theta_2)$ and $w(\theta_3)$ are merged in one window and its orientation (θ_r) is performed. If the orientation obtained θ_r has a value between -75° and -15° thus θ= θ_r, θ_1=θ_r, θ_2=θ_r and θ_3=θ_r, else we keep the values of θ, θ_1, θ_2 and θ_3.

3.6. Erroneous inclination

When a window contains several writings in different orientations, the window orientation will be erroneous (see Figure 10 and Figure 16 (f)).

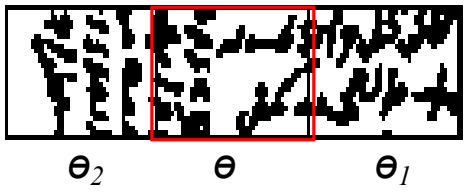


Figure 10. Erroneous inclination.

To detect this phenomenon, we observe the horizontal surrounding windows which will have angles different of θ, more precisely, when θ_1 is different to θ, θ_2 is different to θ and θ_1 different to θ_2 In Figure 10 (a), θ=60°, θ_1=0° and θ_2=90°.

Since this case arises inside the main horizontal writing, we use naturally the vertical projection profile to resolve this case. We look for the first minimum value in the projection profile from the right representing the end of the first inclination (I_m minimum index). Then we divide at I_m the window in two windows w_{left} and w_{right} (see Figure 11). Finally, we assign w_{right} to the window $w(\theta_1)$ and w_{left} to $w(\theta_2)$.

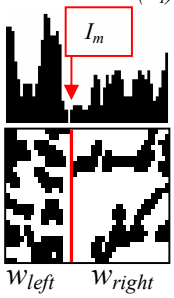


Figure 11. The erroneous inclination correction.

3.7. Paving Correction

As applied automatically, the paving edges can cross the connected components which create problem (false maxima) in text line detection (see Figure 12). The incorrect paving exists only in the horizontal and the vertical zones. We need to correct the position of these edges by proceeding a horizontal or vertical shift in order that the local paving covers the local connected components.

In the horizontal (resp. vertical) zone, the edge that divides two consecutive rows (resp. columns) is moved to the nearest position in these rows (resp. columns) when the horizontal (resp. vertical) projection vector for each of their two consecutive windows has his minimum value (see Figure 16 (h)).

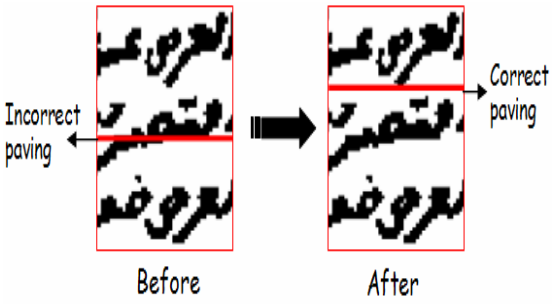


Figure 12. Example of incorrect paving in a horizontal zone.

3.8. Text Line Detection

Our idea is based on the projection profile and the follow-up of the baseline.

The text line follow-up starts in the first window on the right side of the page. It is based on the orientation recalculated after edge correction. The algorithm starts by looking for the new maxima (see Figure 13 (a)). Each peak represents the starting point (P_S) of the baseline blj. The ending point (P_e) of the baseline is calculated using the P_S , the orientation, the width and the height of each window (see Figure 13 (b)). The baseline blj is calculated basing on the two points (P_S , P_e) and the orientation of the window. The connected components that belong to a baseline are looked for construct the text line (see Figure 13 (c)).

A step of text line correction follows the text line detection to assign the non-detected components and the diacritical symbols to the appropriate text line (see Figure 13 (c) and (d)). A distance method is used to address this problem. First, the distance between the centroid of non-detected component or diacritical symbol (C_i) and the text line is calculated. C_i is assigned to the text line l_j if $d_{C_i,l_j} < d_{C_i,l_{j+1}}$ else to l_{j+1} (see Figure 14).

For each zone, the text lines are clustered to form the zone text lines. Then the relations between the text lines zones are studied to form the document text lines (see Figure 16 (i)).

For this, we see if a connected component is up to two text line between two zones. If it is the case, we merge the two text lines by one text line.

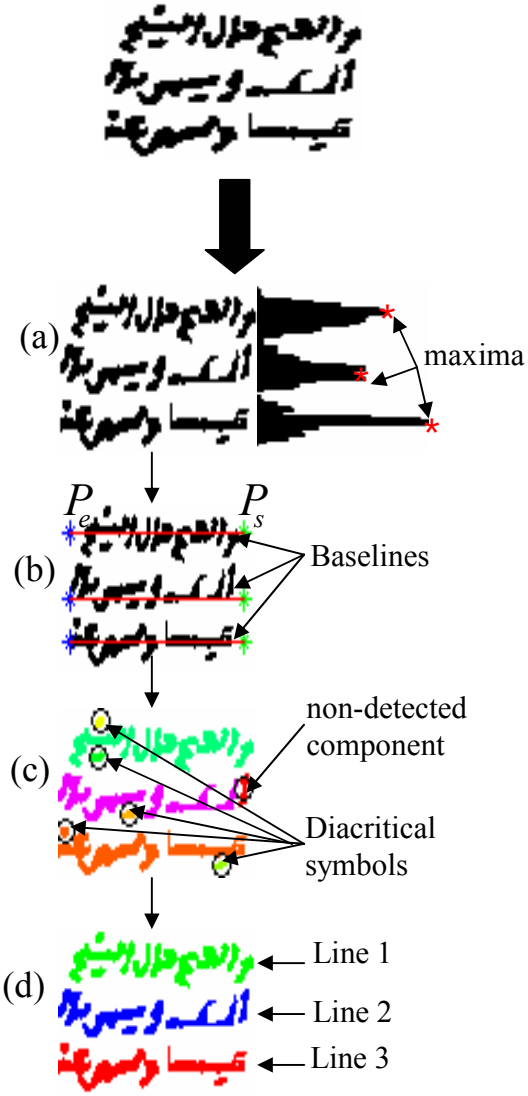


Figure 13. Text line detection steps for a window.

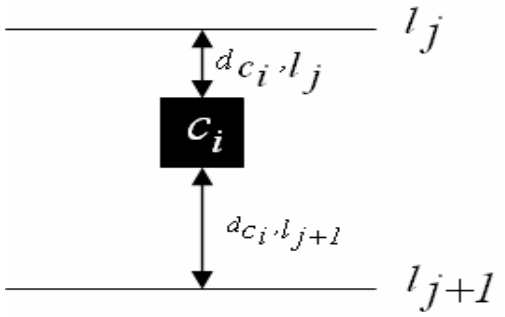


Figure 14. Assignment of the non-detected components and diacritical symbols.

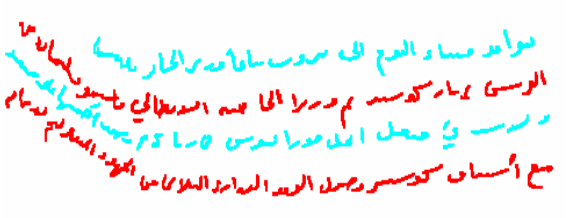
During this step, we can encounter two problems: touching and overlapping between the text lines where a connected component belongs to two consecutive text lines. These problems are detected but they will be addressed during the next step of the recognition. The methods proposed in the literature to resolve this problem are based in the dividing of the touching and the overlapping connected components. Thus, these solutions can not give us the correct connected components.

4. Experimental Results and Discussion

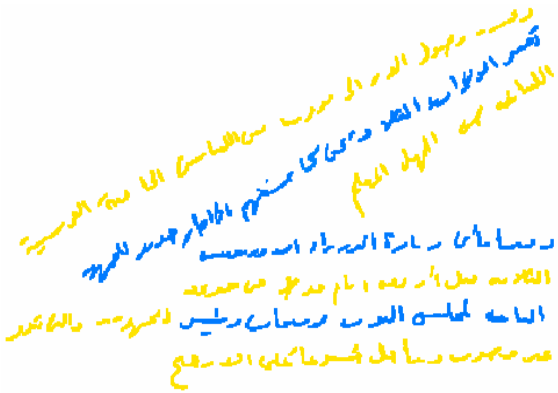
To study the effectiveness of our approach, we experimented it on 50 handwritten Arabic documents that contain 1500 text lines. They are manuscripts from the 16th century belonging to the Library National of Tunis. The tests were prepared after a manual computing of zones and text lines from each document. The orientation experimented varies between -75° and +90°. The speedup of the system covers all the steps from paving to line detection. It is depending on the document size and the window size of paving. The tests were carried out on a PC equipped with a microprocessor Pentium M 1.4 GHz and 1 GB of memory under Windows XP. The application has been developed with MATLAB R2007b. In the multi-skew detection level, we have achieved a level of accuracy of 96%, this rate increases to 98% if we don't take care with the small zones that are not detected. The 2% error rate is due to the paving and the false inclination. In the text line segmentation level, the accuracy reaches 97.6%. The 1.5% of text lines not detected is due to the multi-skew detection algorithm. The 0.9% error rate is due to the presence of diacritical symbols in the beginning of the lines that create false maxima. Figure 15 illustrates the usability of our algorithm on a sample of 2 documents chosen arbitrarily from the 50 documents processed. To identify the lines, each consecutive pair of lines is presented by two different colours. Table 1 describes the results of our method.

Table 1. Algorithms results.

	Extracted	Not Extracted	Error
Zones	96 %	2 %	2 %
Lines	97.6 %	1.5 %	0.9 %



(a)



(b)

Figure 15. Some results of the multi-oriented text line extraction approach.

5. Conclusion and Future Work

An original approach is proposed in this article, which aims to extract the text lines from the multi-oriented handwritten Arabic documents. First, the multi-skewed zones are detected using the paving, the WVD and the nearest-neighbor principles. Then, the text lines are extracted based on the orientation of each zone and the baselines. The 97.6% of extraction rate show the efficacy and the performance of our approach. The next step of this work is related to the segmentation of the lines into single words.

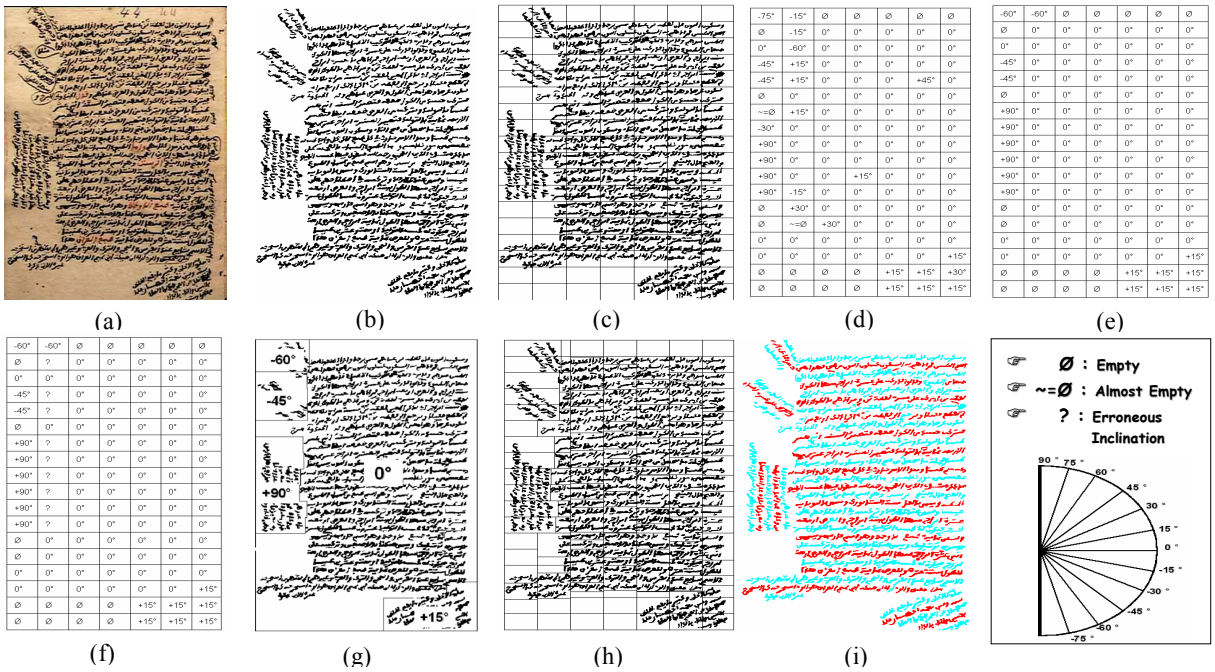


Figure 16. The results for the different steps of the multi-oriented text line extraction approach.

References

[1] A. Chaudhuri, S. Chaudhuri, "Robust detection of skew in document images", IEEE Transactions on Image Processing, Vol.6, No.2, 1997, pp.344-349.

[2] S. Chen, and R. M. Haralick, "An automatic algorithm for text skew estimation in document images using recursive morphological transforms", Proc of International Conference on Image Processing, Austin, USA, vol. 1, 1994, pp.139-143.

[3] A. K. Das, B. Chanda, "A fast algorithm for skew detection of document images using morphology", International Journal on Document Analysis and Recognition, vol.4, No.2, 2001, pp.109-114.

[4] P. Flandrin, "Temps-fréquence", HERMES, Paris 1993.

[5] A. Hashizume, P. S. Yeh, and A. Rosenfeld, "A method of detection the orientation of aligned components," Pattern Recognit. Lett., 1986, vol. 4, pp. 125–132.

[6] F. Hlawatsch, F. Auger, "Temps-fréquence", HERMES, LAVOISIER, Paris 2005.

[7] E. Kavallieratou, N. Fakotakis, G. Kokkinakis, "Skew angle estimation for printed and handwritten documents using the Wigner-Ville distribution", Elsevier, Image and Vision Computing, Volume 20, Number 11, 1 September 2002, pp. 813-824(12).

[8] L. Likforman-Sulem, A. Hanimyan, C. Faure, "A Hough Based Algorithm for Extracting Text Lines in Handwritten Document", Proc. of ICDAR'95, 1995, 774-777.

[9] L. Likforman-Sulem, C. Faure, "Extracting lines on handwritten documents by perceptual grouping, in Advances in Handwiting and drawing: a multidisciplinary approach", C. Faure, P. Keuss, G. Lorette, A. Winter (Eds), Europia, Paris, 1994, pp. 21 38.

[10] L. Likforman-Sulem, C. Faure, "Une méthode de résolution des conflits d'alignements pour la segmentation des documents manuscrits" , Traitement du Signal, 1995, 12(6) :541-549.

[11] G. Nagy, S. Seth, and M. Viswanathan, "A prototype document image analysis system for technical journals", Comput., 1992, vol. 25, pp. 10–22.

[12] N. Otsu, "A Threshold Selection Method from Gray Level Histograms", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66.

[13] E. Oztop, A. Y. Mulayim, V. Atalay, F. Yarman Vural, "Repulsive attractive network for baseline extraction on document images", Signal Processing, 1999,75:1-10.

[14] U. Pal, B.B. Chaudhuri, "Skew angle detection of digitized Indian script documents", IEEE Trans. Pattern Anal. Mach. Intell, 1996.

[15] Z. Shi, V. Govindaraju, "Line Separation for Complex Document Images Using Fuzzy Runlength", Proc. Of the Int. Workshop on Document Image Analysis for Libraries, Palo, Alto, CA, January 23-24, 2004.

[16] A. Zahour, B. Taconet, P. Mercy, and S. Ramdane, "Arabic hand-written text-line extraction", Proceedings of the 6th ICDAR, Seattle, 2001, pp. 281–285.