

# Detection of Text Lines of Handwritten Arabic Manuscripts using Markov Decision Processes

Youssef Boulid<sup>1</sup>, Abdelghani Souhar<sup>2</sup>, Mohamed Youssfi Elkettani<sup>1</sup>

University Ibn Tofail, Faculty of Sciences, <sup>1</sup>Department of Mathematics, <sup>2</sup>Department of Computer Science, Kenitra, Morocco

**Abstract** — In a character recognition systems, the segmentation phase is critical since the accuracy of the recognition depend strongly on it. In this paper we present an approach based on Markov Decision Processes to extract text lines from binary images of Arabic handwritten documents. The proposed approach detects the connected components belonging to the same line by making use of knowledge about features and arrangement of those components. The initial results show that the system is promising for extracting Arabic handwritten lines.

**Keywords** — Text Line Segmentation, Handwritten Arabic Document, Connected Component Analysis, Markov Decision Processes.

---

## I. INTRODUCTION

---

MANY documents are stored in their original state (as paper) and have yet to be exploited to produce their electronic versions. There are systems for converting the document image into text version, in order to facilitate the access and the search in the document. These systems often consist of five steps: pre-processing, segmentation, feature extraction, character recognition and post-processing.

Segmentation is the process of partitioning the document into homogeneous entities (lines, words or characters), it is a very important phase in the character recognition process since the recognition rate depends strongly on it. The segmentation of text lines is to assign the same label to units that are spatially aligned (pixels or connected components) [1].

The Arabic script is naturally cursive, which makes the extraction of lines from handwritten Arabic document a real challenge [2]. Among the challenges, we have:

- The writing styles which differ widely from one writer to another.
- The fluctuation of the base line due to the movements of the pen results in different variations within the same text line.
- The cluttered writing style and the ascender and descender introduce ligatures between parts of words which make overlap the adjacent lines.
- The diacritical points, which lie below or above the words, further complicate the task.

To cope with these problems, we propose an approach for the detection and the extraction of Arabic handwritten text lines by exploiting knowledge about the connected components arrangement using Markov Decision Processes. The components that belong to the same line are those satisfying some constraints related to the knowledge.

The rest of this paper is organized as follows: Section II examines some related works. Section III provides insight into Markov Decision Processes. Section IV describes our approach for the segmentation of

Arabic handwritten text lines. Section V gives some results and finally Section VI concludes the paper.

---

## II. RELATED WORKS

---

In general there are two approaches for the segmentation of text lines: either through searching for separating locations between lines, or by searching for physical units such as Connected Component (CC) constituting a line.

As mentioned in [3] the extraction techniques can be divided into three classes: top-down, bottom-up and hybrid methods.

In the top-down methods, the document image is partitioned into regions, often recursively based on the global characteristics of the image. These methods are influenced by the large curvatures of the lines and the lines of text that touch each other. For the bottom-up methods, the basic elements such as pixels or CC are grouped to form text line, for that a lot of calculation and heuristic analysis are needed. Hybrid approaches combine the two classes of techniques to give best results. The works in [4,1], present a revision for existing segmentation methods of handwritten text lines. There are also competitions like ICDAR [5] and ICFHR [6] that participate in advancing this field of research.

Table I presents an overview of some related methods.

---

## III. MARKOV DECISION PROCESSES

---

Markov decision processes (MDP) are defined as controlled stochastic processes satisfying the Markov property, assigning rewards to state transitions [7,8] They are defined by:

- $S$ , is the state space within which the process evolve;
- $A$ , is the space of actions that control the dynamics of the state;
- $P()$  are the transition probabilities between states;
- $r()$  is the reward function on transitions between states.

The transition probabilities characterize the dynamics of the system state.  $P(s'/s, a)$  represents the probability that the system spends in state  $s'$  after executing action  $a$  in state  $s$ . Conventionally requires that

$$\forall s, a, \sum_{s'} P(s'/s, a) = 1.$$

As a result of the choice of action  $a$  in state  $s$ , we receive a reward. Positive values of  $r(s, a)$  may be regarded as gains and negative values as costs. This award may depend on the end state  $s'$ ;  $r(s, a, s')$ .

A function  $\Pi : S \rightarrow A$  assigning at each time step an action to every state is called a (stationary) policy.

The value function of a policy  $v_{\Pi} : S \rightarrow \mathcal{R}$  is defined so that  $v_{\Pi}(s)$  represents the infinite horizon, discounted cumulative reward attached to a policy  $\Pi$  applied to a MDP, with initial state  $s$ .

TABLE I  
AN OVERVIEW OF SOME METHODS OF SEGMENTATION OF HANDWRITTEN TEXT LINES

Ref	Methods	Description	Rate of segmentation
[12]	Projection profile and K-means	The document is split vertically into several strips, and text is detected based on the histogram of the PP. Text blocks are clustered using K-means in three classes; the large blocks are split vertically into a number of average text blocks according to the analysis of the neighborhood and interline spacing.	96%
[13]	Connected Component	The algorithm starts by removing outlier components using a threshold value and then characters belonging to tow lines are detected and divided horizontally at the half distance. For line detection, a rectangular neighborhood is centered on a current component and increases to include those that satisfy certain conditions. The filtered components at the beginning are reallocated to the corresponding lines with respect to distance from their bounding box.	93.35%
[11]	Projection Profile	After removing small components and linking broken characters, Fourier curve fitting within the horizontal PP is used to locate the point of separation. The contour is used to extract the base line of the CC which allows locating the cut point between different adjacent lines. The components are assigned according to the closest line that is approximated by a polynomial curve that fits the pixels in the baselines. Finally, lifted small size components are reassigned to their line according to the closest CCs nearest neighbour in four directions.	93% and 94% for MS 95% and 90% respectively
[14]	Projection profile and morphological dilatation	The horizontal PP is used to estimate the skew line. The slop in each zone is used for smearing it, using dilation with adaptive structuring element that changes the size and the slop according to the zone. In the second stage the big blobs are detected with a recursive function that search for the cut point in order to separate the components and match them with their lines following attraction and repulsion criterion. Touching components are detected and separated using erosion recursively in order to detect the thinness part, which corresponds to the cut point.	96% and 97% for MS 95% and 90% respectively
[15]	Adaptive Local Connectivity map (ALCM)	The grey-level image produced by the ALCM algorithm is binarized using an adaptive thresholding algorithm to detect the location of the text lines (masks). In the second step, CCs are grouped into location masks for each text line. Finally, the text lines are extracted by superimposing the components with text line pattern mask. For touching characters, the contour is divided into pieces of segments and reallocated according to the distance from their center of mass with respect to the nearest text line.	99.5%
[3]	Wigner-Ville distribution and Projection profile	The image is divided into cells and the orientation within each cell is calculated using the Wigner-Ville distribution on the PP histogram. The cells having similar orientation are joined. The method exploits the projection peaks and the orientation within each area to follow the connected components forming text lines. The diacritical components are reassigned to the nearest text line. To separate connected lines, for each touching character and from a starting point, the flow continues beyond the intersection and the strokes of the same component are recovered by analyzing the angular variation corresponding to the curvature of the descending character.	98.6%.
[16]	BreadthFirst Search	The algorithm begins by removing diacritics and then the sparse similarity graph is built based on the local orientation of the component. Text lines are represented by a disjoint set using BreadthFirst Search (BFS). The affinity propagation clustering method is used to assign the blobs to text lines.	95,6%
[17]	Breadth-first-search and affinity-propagation clustering	After removing small components, the direction of text line is detected at each component by locating the region having maximum neighboring components, and then the local orientation is estimated by the least square line passing through the centroid of the components. After that, a graph is constructed where the nodes correspond to components and weights on edges correspond to distance to the estimated orientation line. The shortest path-algorithm is used to complete the graph. Two estimations of text lines are obtained: the Breadth-first-search and the affinity-propagation clustering method. The splitting error are corrected using affinity propagation while merging error are corrected using Expectation Maximization. Touching components are localized by finding the common tangent of the convex-hull of successive components, and then the component is split near the centroid. Finally, the diacritics and accent components are reassigned to their closest component.	98,76% of F1-score for 90% MS
[5]	Divide image into blocks	The Gaussian filter is used and the image is divide into blocks which are then binarized using an adaptive threshold with respect to the skew angle estimated in each block. Then, the blocks are concatenated to get the path of text lines, and finally text lines are extracted by thinning the background of the path image.	94%

This value can be computed as:

$$v_{\Pi}(s) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t r(s^t, \Pi(s^t)) \mid s^0 = s \right] \quad (1)$$

The expectation is taken over all possible trajectories where  $\langle s^0, a^0, s^1, a^1, \dots, s^t, a^t, \dots \rangle$  from the initial state  $s$ , the policy  $\Pi$  is applied. The factor which ensures that the above infinite sum converges is the discount factor  $0 \leq \gamma \leq 1$ .

The problem of finding an optimal policy with respect to the discounted criterion (1) can be written as:  $\Pi^*: S \rightarrow A$ , so that:

$$v_{\Pi^*}(s) \geq v_{\Pi}(s) \quad \forall \Pi \in A^S.$$

Let's consider the Bellman operator  $\beta$  operating on any value function  $v$  as:

$$h_i^j = f \left( \sum_{k=1}^{n_{i-1}} w_{k,j}^0 x_k \right) \quad j=1, \dots, n_i \quad (2)$$

Let  $v^*$  denote the value function attached to an optimal policy  $\Pi^*(v^* = v_{\Pi^*})$ . It has been shown (see, e.g. [8]) that  $v^*$  can be computed as the unique solution of the system of non-linear equations  $v^* = \beta(v^*)$ :

$$v^* = \max_{a \in A} \left\{ r(s, a) + \gamma \sum_{s' \in S} P(s'/s, a) v^*(s') \right\}, \forall s \in S \quad (3)$$

The optimal policy  $\Pi^*$  can then be obtained from  $v^*$ :

$$\Pi^*(s) = \arg \max_{a \in A} \left\{ r(s, a) + \gamma \sum_{s' \in S} P(s'/s, a) v^*(s') \right\}, \forall s \in S \quad (4)$$

There exist several methods for computing the optimal policy  $v^*$ . In this paper we have used the value iteration algorithm, shown in the figure below:

```

1: Procedure Value_Iteration( $S, A, P, R, \theta$ )
2:   Inputs
3:      $S$  is the set of all states
4:      $A$  is the set of all actions
5:      $P$  is state transition function specifying  $P(s'/s, a)$ 
6:      $R$  is a reward function  $R(s, a, s')$ 
7:      $\theta$  a threshold,  $\theta > 0$ 
8:   Output
9:      $\pi[S]$  approximately optimal policy
10:     $V[S]$  value function
11:   Local
12:    real array  $V_k[S]$  is a sequence of value functions
13:    action array  $\pi[S]$ 
14:    assign  $V_0[S]$  arbitrarily
15:     $k \leftarrow 0$ 
16:   repeat
17:      $k \leftarrow k+1$ 
18:     for each state  $s$  do
19:        $V_k[s] = \max_a \sum_{s'} P(s'/s, a) (R(s, a, s') + \gamma V_{k-1}[s'])$ 
20:     until  $\forall s |V_k[s] - V_{k-1}[s]| < \theta$ 
21:     for each state  $s$  do
22:        $\pi[s] = \arg \max_a \sum_{s'} P(s'/s, a) (R(s, a, s') + \gamma V_k[s'])$ 
23:   return  $\pi, V_k$ 
    
```

Fig. 1. Pseudo code of value iteration algorithm.

## IV. PROPOSED APPROACH

The methods based on CC analysis use geometric information such as shape, orientation, position, and size of CC for grouping them in rows. These methods are more suitable for complex documents compared to methods based on Projection Profile (PP) [9,10], even if they could be sensitive to changes in the size and component structures [11].

Firstly the diacritical components are removed from the document image using an estimated threshold. At the end of the process, they will be reallocated to their corresponding lines. The diagram of the proposed system is shown in Fig.2.

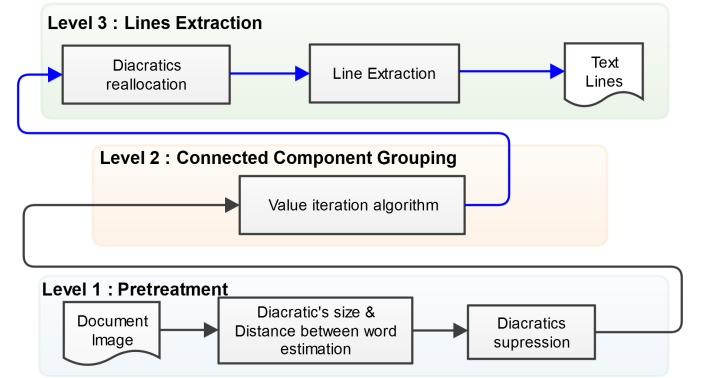


Fig. 2. Block diagram illustrating the sequence of treatments in the proposed approach.

We represent the environment by a set of CCs which are connected to each other, in such way that allows us to link only those which probably belong to the same line.

For this purpose, a circle with an estimated radius is centered on each CC, and from all the components that are located inside the circle, we take only those that are located between  $145^\circ$  and  $-145^\circ$  relative to the current component (Fig. 3).

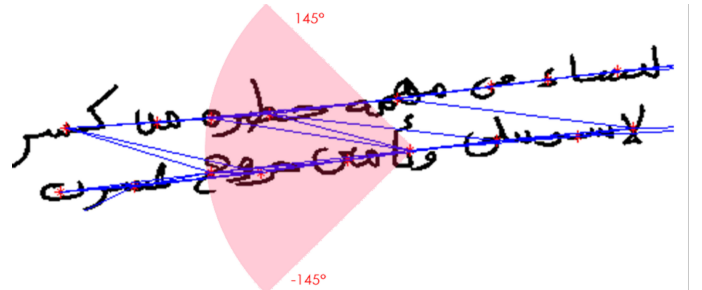


Fig. 3. Example of links (in blue) between components of two lines

The radius of the circle is estimated taking into consideration the average distance between words which is estimated by:

$$D_w = \frac{\sum_{i=1}^m W_i}{m} * 2 \quad (5)$$

$W_i$  represents the width of the  $i^{\text{th}}$  component and  $m$  represents the total number of components in the image after removing diacritics.

We found that there are some components that are connected with others while these latter do not participate in the same line. To solve this problem we introduce in the next paragraph the concept of field of vision.

### A. Field of Vision Function

To detect the components that are in the left side of another (which probably belong to the same line), the algorithm loops through pixels in rows and columns from the left side of the current component, and for each line, the index of the first black pixel is added to a list.

Thus, we denote  $FV(C)$  the function of field of vision for the component  $C$ , such as for the example in Fig. 4, we have:

$$FV(ح) = \{د, ر, ص\}$$

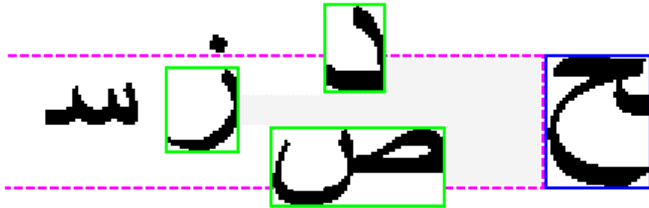


Fig. 4. Example of field vision function.

To improve the previous results we add the criterion of field of vision to keep only the components that are directly visible from the current component. The Fig. 5, illustrates the result of this improvement.

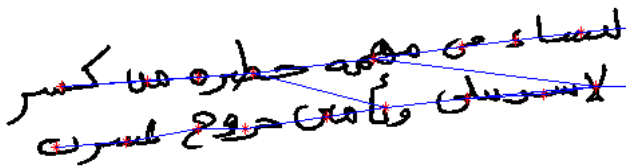


Fig. 5. Example of links (in blue) between components of two lines after using field of vision criterion

The field of vision function improves the environment, which can be used to extract the lines.

The problem now is to choose for each component the one that most likely belong to the same line as the previous component.

In the next paragraph, we will show that this problem could be resolved using MDP, where the environment can be considered as an oriented graph, in which the connected components represent the nodes and the edges represent some measures of alignment between the nodes.

### B. Proposed Solution

So, from this point of view, we consider the following:

1. The **States** correspond to connected components:

$$\{S_i\} = \{C_i\}, i = 1 \dots m$$

2. The **Actions** correspond to transitions from a given state source  $S_i$  to other possible states  $S_j$  :

$$A(S_i) = \{S_j\}$$

3. The **Transition Probabilities**; For the example in Fig. 6, the transition probabilities are computed as follows:

$$P(S'_{1,1}, S_1) = 0.8 * \nu(S'_{1,1}) + 0.1 * \nu(S'_{1,2}) + 0.1 * \nu(S'_{1,3})$$

$$P(S'_{1,2}, S_1) = 0.8 * \nu(S'_{1,2}) + 0.1 * \nu(S'_{1,1}) + 0.1 * \nu(S'_{1,3})$$

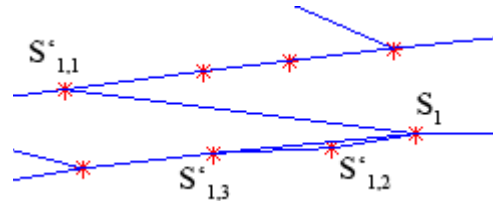


Fig. 6. Example of links (in blue) between components (red dots) of two lines

4. The **Reward Function**, is estimated as a combination of the Euclidean distance between the state  $S$  and the next state  $S'$ , and the percentage of participation of  $S'$  related to  $S$ .

If we consider that:

- $P_{S'/S}$ , The percentage representing the portion of component  $S'$  which is included in the field of vision of component  $S$  (Fig. 7).
- $P'_{S/S'}$ , The percentage representing the portion of component  $S'$  which is included in the field of vision of component  $S$ , relatively to  $S'$  (Fig. 7).



Fig. 7. The values of  $P_{S'/S}$  and  $P'_{S/S'}$  respectively for a (55%, 0.48%), for b (0.38%, 0.38%), for c (0.96%, 0.68%) and for d (0.61%, 100%).  $S'$  represents the green rectangle and  $S$  represents the blue rectangle, the gray part represents the portion in  $S'$  included in  $S$ .

So the Reward Function  $r(s', s)$ , can be estimated by:

$$port = (2 * P_{S'/S} + P'_{S/S'}) / 3 \tag{6}$$

$$dist = \frac{(3 * D_w - D(s, s'))}{3 * D_w} \tag{7}$$

$$r(s', s) = \frac{port + dist}{2} \tag{8}$$

After several experimental tests, we found that the value 0.95 for the **Discount Factor** gives the best results.

At the end of the execution of the value iteration algorithm, we get a list that contains the states and their successors (the next state having maximum utility value). So from this list we recover all the linked components participating to the same line.

Fig. 8, illustrate the result of the value iteration algorithm for a given image.

In order to extract the components with their diacritics, we first search for the polyline that goes through all the pixels of the lower contours of the components. This polyline is then shifted down vertically in order to include diacritical points that are below the words. Secondly we define a polygon which is bounded by the polyline and the two top corners of the image. Finally we extract and suppress the portion of the image represented by this polygon (as shown in Fig. 8.d).

This treatment is executed iteratively for each line, until there are no more rows to be extracted.

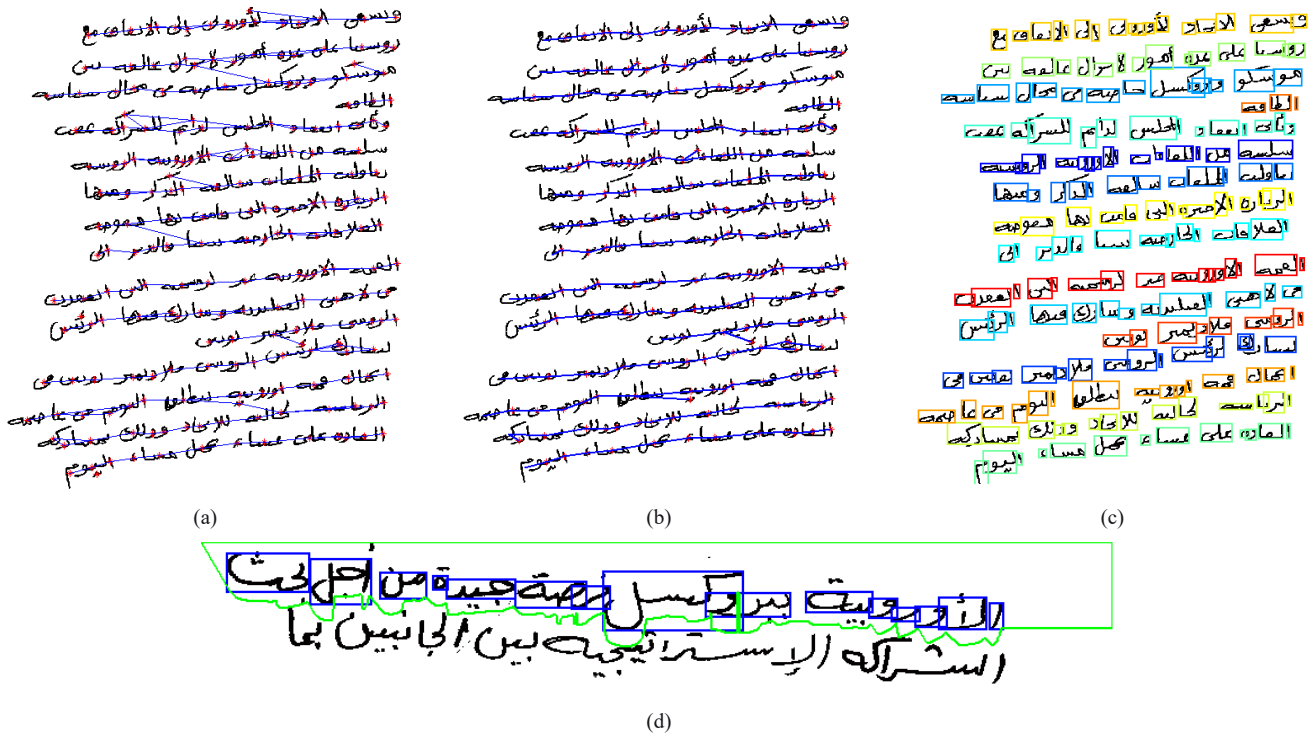


Fig. 8. Example of result of value iteration algorithm, (a) the initial state of the environment, (b) the result of value iteration algorithm, (c) connected component grouping. (d) Example of components detection, diacritics assignment and line extraction

V. EXPERIMENTATION

For experimentation, we have used a subset of the publicly available data set “Handwritten Arabic Proximity Datasets”, which contains 125 images of Arabic handwritten document [18].

To evaluate the performance of the system, we used the criterion Matching Score (MS) [19] which is calculated as follows:

$$MS(r_i, g_j) = \frac{T(P(r_i) \cap P(g_j))}{T(P(r_i) \cup P(g_j))} \tag{9}$$

With MS(ri,gi) is a real number between 0 and 1 which represents the matching score between the zone ri resulted from the algorithm and the zone gj in the ground truth. P corresponds to pixels that represent the foreground (text) and T is an operator that counts the number of pixels in each zone.

If the MS for a zone is found above a threshold, this is counted as true positive (TP). The resulted zones that do not match any zone in the ground truth are counted as false positive (FP) and the zones in the data set that are left unmatched are considered false negative (FN). Therefore, we count precision, recall and F1-score as follows:

$$precision = \frac{TP}{TP + FP} \tag{10}$$

$$\sum_{i=1}^N u_i \geq 1 \tag{11}$$

$$u_{i-1} \times \prod_{j=1}^{i-1} \prod_{k=1}^{i-1} (1 - v_{k,j}^{-1}) = 0 \quad \forall i = 2, \dots, N \tag{12}$$

According to the tests we obtained a F1-Score respectively of 95.8 % and 90.5 % for the MS-threshold values of 90% and 95%. Table II

compares the score of the proposed method with other methods tested on the same data set.

TABLE II  
RESULTS OBTAINED ON THE ORIGINAL DATA IN [18]

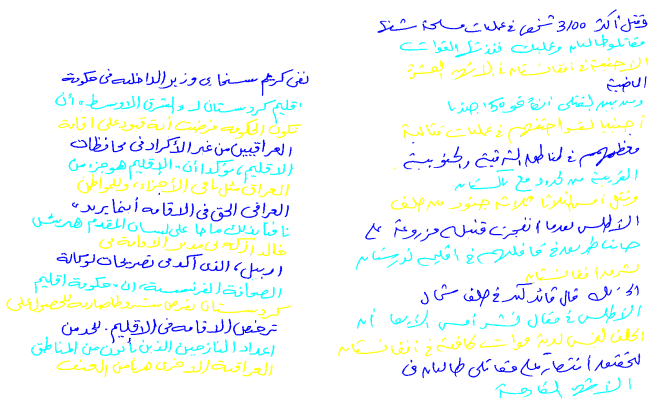
Methods	F1-Mesure	
	MS=90	MS=95
The method in [16]	95.6	90.9
The method in [17]	98.8	Not reported
The proposed method	95.8	90.5

Fig. 9 illustrates some visual results of our approach for segmentation of Arabic handwritten documents text line.

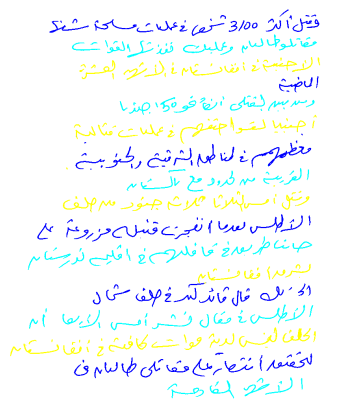
As can be seen in Fig.9.e, even though this proposed approach has been adapted here for Arabic script it could be used in the extraction of text lines for Latin script, by just changing the angle parameter and the field of vision direction.







(c)



(d)

George Washington was one of the Founding Fathers of the United States serving as the commander in chief of the Continental Army during the American Revolutionary War. He also presided over the convention that drafted the Constitution which replaced the Articles of Confederation. The Constitution established the position of President of the republic which Washington was the first to hold. Washington was elected President at the unanimous choice of the 69 electors in 1789 and he served two terms in office. He oversaw the creation of a strong well-financed national government that maintained neutrality in the wars raging in Europe suppressed rebellion and won acceptance among Americans of all types. His leadership style established many forms and rituals of government that have been used since such as using a cabinet system and delivering an inaugural address. Further the peaceful transition from his presidency to the presidency of John Adams established a tradition that continues into the 21<sup>st</sup> century. Historically Washington has been widely regarded as the father of his country.

(e)

VI. CONCLUSION

In this paper we presented an approach based on connected components analysis for extracting text lines from the binary Arabic handwritten documents. We formulate the problem in such a way that allows us to use Markov Decision Processes to solve it, which has given promising results.

In the future work we estimate to improve the diacritical assignment in the process of line extraction and to incorporate a preprocessing phase concerning the segmentation of touching and overlapping lines to accurately separate the different strokes belonging to different characters.

REFERENCES

[1] Likforman-Sulem, L., Zahour, A., & Taconet, B. (2007). Text line segmentation of historical documents: a survey. International Journal of Document Analysis and Recognition (IJ DAR), 9(2-4), 123-138.

[2] Li, Y., Zheng, Y., Doermann, D., & Jaeger, S. (2008). Script-independent text line segmentation in freestyle handwritten documents. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 30(8), 1313-1329

[3] Ouwayed, N., & Belaïd, A. (2012). A general approach for multi-oriented text line extraction of handwritten documents. International Journal on Document Analysis and Recognition (IJ DAR), 15(4), 297-314

[4] Razak, Z., Zulkiflee, K., Idris, M. Y. I., Tamil, E. M., Noor, M. N. M., Salleh, R., ... & Yaacob, M. (2008). Off-line handwriting text line segmentation: A review. International journal of computer science and network security, 8(7), 12-20.

[5] Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., & Alaei, A. (2013, August). Icdar 2013 handwriting segmentation contest. In Document Analysis and Recognition (ICDAR), 2013 12th International Conference on (pp. 1402-1406). IEEE

[6] Gatos, B., Stamatopoulos, N., & Louloudis, G. (2010, November). Icfhr 2010 handwriting segmentation contest. In Frontiers in handwriting

recognition (icfhr), 2010 international conference on (pp. 737-742). IEEE.

[7] Bertsekas D. P., Dynamic Programming : Deterministic and Stochastic Models, Prentice-Hall, 1987.

[8] Puterman M., Markov Decision Processes : Discrete Stochastic Dynamic Programming, John Wiley & Sons, Inc., New York, USA, 1994.

[9] Bennisri, A., Zahour, A., & Taconet, B. (1999). Extraction des lignes d'un texte manuscrit arabe. In Vision interface (Vol. 99, pp. 42-48).

[10] Nicolaou, A., & Gatos, B. (2009, July). Handwritten text line segmentation by shredding text into its lines. In Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on (pp. 626-630). IEEE

[11] Adiguzel, H.; Sahin, E.; Duygulu, P., "A Hybrid for Line Segmentation in Handwritten Documents," Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on , vol., no., pp.503,508, 18-20 Sept. 2012

[12] Zahour, A., Likforman-Sulem, L., Boussalaa, W., & Taconet, B. (2007, September). Text line segmentation of historical arabic documents. In Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on (Vol. 1, pp. 138-142). IEEE.

[13] Khandelwal, A., Choudhury, P., Sarkar, R., Basu, S., Nasipuri, M., & Das, N. (2009). Text line segmentation for unconstrained handwritten document images using neighborhood connected component analysis. In Pattern Recognition and Machine Intelligence (pp. 369-374). Springer Berlin Heidelberg

[14] Khayyat, M., Lam, L., Suen, C. Y., Yin, F., & Liu, C. L. (2012, March). Arabic handwritten text line extraction by applying an adaptive mask to morphological dilation. In Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on (pp. 100-104). IEEE

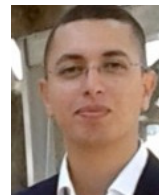
[15] Shi, Z., Setlur, S., & Govindaraju, V. (2009, July). A steerable directional local profile technique for extraction of handwritten arabic text lines. In Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on(pp. 176-180). IEEE.

[16] Kumar, J., Abd-Almageed, W., Kang, L., & Doermann, D. (2010, June). Handwritten Arabic text line segmentation using affinity propagation. In Proceedings of the 9th IAPR International Workshop on Document Analysis Systems (pp. 135-142). ACM

[17] Kumar, J., Kang, L., Doermann, D., & Abd-Almageed, W. (2011, September). Segmentation of handwritten text lines in presence of touching components. In Document Analysis and Recognition (ICDAR), 2011 International Conference on (pp. 109-113). IEEE

[18] Handwritten Arabic Proximity Datasets. Language and Media Processing Laboratory. <http://lamprsv02.umiacs.umd.edu/projdb/project.php?id=65>

[19] Phillips, I. T., & Chhabra, A. K. (1999). Empirical performance evaluation of graphics recognition systems. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 21(9), 849-87



**Youssef Boulid** received his M.S. degree in Decision Support Systems and Project Management in 2012 from university Ibn Tofail, Faculty of science, Kenitra- Morocco. Currently he is preparing a PhD in the Computer Science Department at the same faculty. His research interests include image processing, handwritten document analysis, Arabic handwritten recognition and Artificial intelligence.



**Abdelghani Souhar** received M.S. degree in applied Mathematics in 1992, PhD degree in computer science in 1997 from University Mohammed 5 - faculty of science in Rabat - Morocco. Now he is a Professor at university Ibn Tofail - faculty of science in Kenitra - Morocco. His research interests include Computer Aided Engineering, Computer Aided Design and Artificial Intelligence.



**Mohamed Elyoussfi Elkettani** received M.S. degree in applied mathematics in 1980 and PhD degree in Statistics in 1984 from Orsay Faculty of Science, University of Paris XI. Now he is a Professor at university Ibn Tofail - faculty of science in Kenitra - Morocco. His research interests include Multivariate statistics and Image recognition algorithms.