Generalized Hough Transform for Arabic Printed Optical Character Recognition

Sofien Touj[1], Najoua Ben Amara[2], and Hamid Amiri[1]
[1]National Engineers School of Tunis, Tunisia
[2] National Engineers School of Monastir, Tunisia

Abstract: *The Hough Transform (HT) is a technique commonly used in image processing. It is known for its capacity to detect objects in a given image. In the present paper, we propose to explore the properties of the HT and the use of the Generalized HT (GHT) in Arabic Optical Character Recognition (AOCR). Hence, we first present a GHT based approach for the recognition of Arabic printed characters in their different shapes depending on their position in the word. Accordingly character models are stored in a structure called dictionary which is used further for text recognition. In fact, we have proposed two segmentation-by-recognition techniques for cursive printed writing recognition. The first one uses a technique by a dynamic sliding window. The second one is based on the identification and the localisation of the characters within a word or a part of a word called also sub word. Some outcomes of this study are also assessed in this paper.*

Keywords: *Generalized Hough transform, Arabic printed optical character recognition, printed cursive writing, segmentation by recognition techniques.*

Received July 6, 2004; accepted November 1, 2004

1. Introduction

Arabic is written from right to left. Its alphabet is composed of twenty-nine letters. Each character may have different shapes depending on its contextual position (at the beginning, in the middle, at the end or isolated), which increases the allographic shapes to be described (Table 1). Moreover, Arabic script is rather semi-cursive in the sense that a word can be composed of one or several separated components which we shall call sub-words. The discontinuities between the different sub-words are due to some characters which cannot be connected with the succeeding character from the left side. Consequently, Arabic Optical Character Recognition (AOCR) continues to pose a major challenge to researchers. This is also attributed to the peculiarities of Arabic fonts, the complexity of the vertical and/or horizontal ligatures in Arabic script (Figure 1) and the lack of supporting utilities like Arabic text databases and dictionaries [1].

Table 1. Arabic characters shapes 37 principal shapes without diacritics (35 shapes and لا and ء), I: Isolated, B: Beginning, M: Median, and E: End.

I	B	M	E	I	B	M	E	I	B	M	E
ا	-	-	ـا	ٮ	ٮـ	ـٮـ	ـٮ	ح	حـ	ـحـ	ـح
د	-	-	ـد	ر	-	-	ـر	س	سـ	ـسـ	ـس
ص	صـ	ـصـ	ـص	ط	-	-	ـط	ع	عـ	ـعـ	ـع
ڡ	ڡـ	ـڡـ	ـڡ	ٯ	ڡـ	ـڡـ	ـٯ	ك	كـ	ـكـ	ـك
ل	لـ	ـلـ	ـل	م	مـ	ـمـ	ـم	ں	ٮـ	ـٮـ	ـں
ه	هـ	ـهـ	ـه	و	-	-	ـو	ى	ٮـ	ـٮـ	ـى

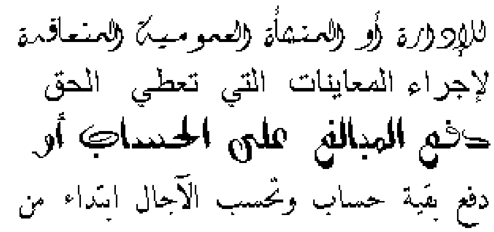


Figure 1. Illustration of the multi-font aspect.

Substantial research efforts have been devoted during last years to AOCR and many approaches have been developed (structural, geometric, statistics, stochastic…). However, certain problems remain open and deserve more attention in order to achieve results equivalent to those obtained for other scripts such as Latin. Besides, other methods must be explored and various sources of information have also to be used [2].

In our study, we focused on the Hough Transform (HT) which is a standard technique in computer vision having many applications for detecting lines, circles and other features in images. In 1981, Ballard generalized the HT to detect arbitrary shapes. He considered the edge orientation into account; which made the algorithm faster and also greatly improved its accuracy [6]. The HT has also been used efficiently in the OCR field for the detection of ascenders and descenders in Arabic writing [3, 4], the estimation of the skew angle in digitized texts [13], the extraction of directional features in Latin writing [8] and the global recognition of printed Arabic sub words [15]. In this paper, we present the adopted approach to use the

Generalized Hough Transform (GHT) to recognize printed Arabic characters in their different shapes. The tested method leans on the storing of characters models in a structure called dictionary. Consequently and in order to exploit the outcomes of our survey, we present two GHT based segmentation-by-recognition techniques for cursive printed writing recognition. The first one uses a technique by a dynamic sliding window. The second one is based on the identification and the localisation of the characters within sub words.

The remainder of this paper is organized as follows. In section 2, we present a brief background of HT. In section 3, we survey the use of GHT for Arabic printed characters recognition. We describe in section 4, the two proposed GHT based architectures for cursive Arabic printed writing recognition. Experimental results and conclusion are addressed in section 5 and 6.

2. The Hough Transform

The Hough Transform was first introduced in the area of computer vision to detect lines in images. It was then extended to other geometrical features like circles and ellipses [6]. In the beginning of eighties it was improved in order to apply it for the detection of general shapes yielding the GHT. The basic idea of HT is to map edge points from the image to the parameter space; which represents all instances of possible features present in the image. Each edge point polls for the instance to which it belongs, the instance with maximum polling defines the feature present in the image. The success of HT can be explained by its global aspect, no a priori knowledge on point distribution is needed, but voting process of each point leads to the emergence of a peak in the accumulator. The voting process gives to the HT robustness toward missing edge points, each point taken individually is not important but all the points will poll for a particular shape.

2.1. Hough Transform for Lines Detection

In the HT space, a line is represented in its normal form by:

$$\rho = xi \cos\theta + yi \sin\theta \qquad (1)$$

Each feature point (xi, yi) votes then for a sinusoid of points in the parameter space (accumulator). Where these sinusoids cross, there are higher accumulator values [5]. Localizing maxima in the accumulator is equivalent to finding the existing lines (Figure 2).

We can extend the HT to other shapes of arbitrary complexity that can be expressed parametrically. The number of parameters required to describe the shape, dictates directly the dimension of the accumulator.

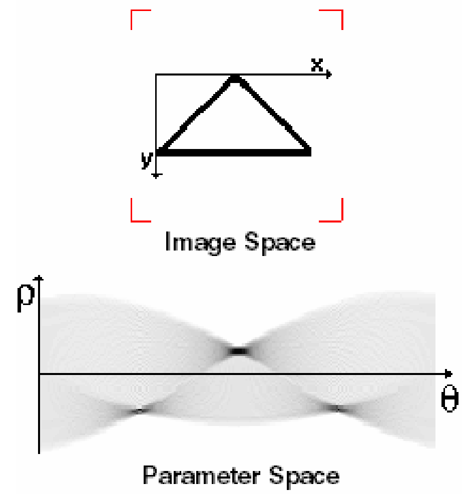


Figure 2. Grey levelled accumulator array (three visible peaks).

2.2. The Generalized Hough Transform

The GHT is an extension of the standard HT to detect any arbitrary objects in an image. In fact, all shapes can not be easily expressed using a small set of parameters. The solution proposed is then to create a table for storing all the edge pixels (feature points) of our target shape. For each pixel, we can store its position relative to a chosen reference point of the shape. The GHT can be made more efficient by forbidding feature points to vote for all possible pixels of the target boundary. Instead feature points vote only for pixels whose representation is coherent with the edge orientation θ. We build then a table that we shall call R_Table, in which we record, for each point with edge tangent orientation θ, its offset relative to the reference point. Thus, in the recognition phase, when we find a point with edge tangent orientation θ, we have to vote only according to all the offsets stored for that specific orientation. Local maxima in the Hough space correspond then to the plausible reference points of the considered shape [5, 12].

3. GHT Based Approach for Arabic Printed Characters Recognition

In this section, we describe the adopted technique to use the GHT for Arabic printed characters recognition. We begin first by extracting the different feature points then we build the characters' models and finally we describe the recognition process.

3.1. Feature Points Extraction

Feature points extraction in Figure 3 is done by applying a gradient operator to the template image and storing the corresponding tangent orientation θ. The gradient operator makes use of two filtering masks: A horizontal mask and a vertical one as shown in Figure 4. By performing convolution on the template image with those two masks independently, we obtain two gradient difference maps. Local gradient orientation at each point is then given by:

$$\theta = \arctan\left(\frac{\Delta v(\mathrm{x}, \mathrm{y})}{\Delta h(x, y)}\right) \quad (2)$$

Origin Character Image (x, y) | Horizontal Gradient map Δh (x, y) | Vertical Gradient map Δv (x, y) | Feature points

Figure 3. Feature points extraction.

$$\begin{bmatrix} -1 & -2 & -3 & -2 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -3 & 0 & 3 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Vertical mask Horizontal mask

Figure 4. Vertical and horizontal masks.

3.2. Models Building

After choosing the reference point, we build the R_table corresponding to each model of character. An indexed dictionary is then created in order to be used in the recognition phase.

- *Reference point choosing*: We choose as reference point, the center of the bounding box of the character model. Such choice reduces the problem of the influence of the little shapes' variability in the localization of the reference point O (x_o, y_o).
- *R_Table building*: The R_Table is obtained by combining the local gradient orientation θ, corresponding to each feature point P_i (x_i, y_i), and the offset Δ (x_i, y_i) between this point and the reference point O (x_o, y_o). Hence, the R_Table is a set of dynamic lists of pair of offsets referenced by the orientation of local gradient θ. Figure 5 illustrates the procedure of building the R_table in the case of the character "ط".

$$\Delta(xi, yi) = O\ (xo, yo) - Pi\ (xi, yi) \quad (3)$$

- *Dictionary building*: After creating the different R_Tables relative to all shapes of Arabic printed characters, we form an indexed dictionary which will be used in the recognition step as shown in Figure 6. The dictionary comprises a set of 37 different models representing Arabic characters shapes previously deprived of their diacritics as shown in Table 1. Each character model includes an R_table and several structural data relative to the presence of diacritics, the height, the width and the pixels density.

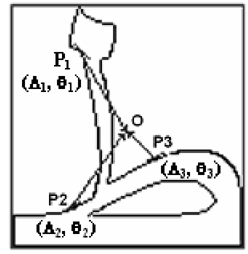


Figure 5. R_Table building.

Model 0

Structural information + **R_Table**

loops, diacritics, height, width, pixels density

θ in degree	List of offsets pairs
0	$\Delta(x_1,y_1)$, $\Delta(x_2,y_2)$, $\Delta(x_3,y_3)$,
1	$\Delta(x_5,y_5)$, $\Delta(x_6,y_6)$
2	$\Delta(x_7,y_7)$
. .	
358	null
359	$\Delta(x_n,y_n)$ …

.
.
.
.
.

Model n

Structural information + **R_Table**

loops, diacritics, height, width, pixels density

θ in degree	List of offsets pairs
0	$\Delta(x_1,y_1)$, $\Delta(x_2,y_2)$, $\Delta(x_3,y_3)$,
1	$\Delta(x_5,y_5)$, $\Delta(x_6,y_6)$
2	$\Delta(x_7,y_7)$
. .	
358	null
359	$\Delta(x_n,y_n)$ …

Figure 6. Dictionary architecture.

3.3. Characters Recognition

Given a target character image, we calculate the local gradient orientations θi according to the formula (2). Using θi to index into the R_Tables, we utilize the offsets Δ (xi, yi) to estimate the position of the reference points of different character models. We accumulate the obtained votes in the different models accumulators. Undergoing a peak seeking process among the entire accumulator set, we retain the more plausible character as well as the position of its reference point in the image (Figure 7) [11].

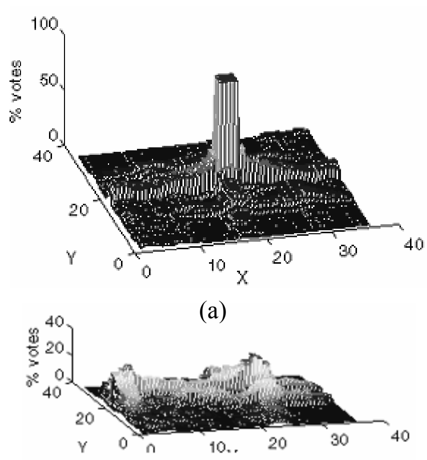
3.4. Various Scales Shapes Detection

The proposed recognition approach supposes that the characters to be recognized have the same size as the models stocked in the dictionary. Accordingly, the accumulators called also parameter spaces have the same size of the target image. To adapt the GHT algorithm to possible scale's variation, an extension toward 4D-parameter spaces proves to be necessary.

Therefore we define two scaling factors S_x and S_y. The following relation can be setup to compute votes in the different models 4D-parameter spaces:

$$O\ (x_o, y_o) = P_i\ (x_i, y_i) + (S_x, S_y)\ .\ \Delta\ (x_i, y_i) \quad (4)$$

The offsets Δ (x_i, y_i) are obtained from the previous dictionary's R_Tables indexed by the gradient orientation θ_i. We note that the reference point has to be chosen as being the origin of the bounding box of the template image [10] as shown in Figure 8.



(a)

(b)

Figure 7. A 3D representation of accumulators corresponding to a sample of character "ط" obtained using (a) the model "ط" and (b) the model "ن".

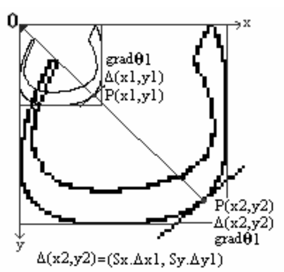


Figure 8. Estimation of the reference point in case of scale change.

Accumulator array corresponding to model "ن" (size 12) obtained using a sample of the character "ن" (size 20): A visible peak in the reference point located in the origin of the image.

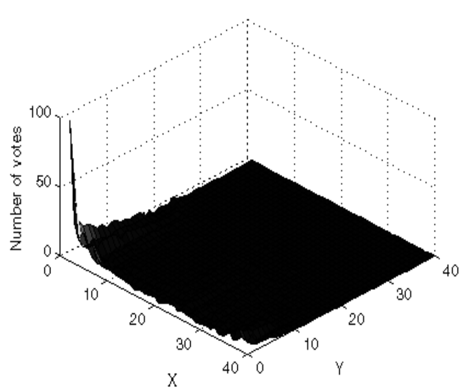


Figure 9. Example of a projection of a 4D-parameter space.

4. Arabic Printed Cursive Writing Recognition

In this section we discuss two segmentation-by-recognition techniques for cursive printed Arabic recognition. The first method is based on the use of a dynamic sliding window; the second one is based on an identification and localization process. Both methods use the GHT based approach for character recognition presented in section 3. A preliminary pre-processing stage is necessary, where the different sub-words are delimited and deprived of their diacritics.

4.1. Pre-Processing

From a printed Arabic text digitized at 300 dpi, we first proceed to the extraction of the different sub words contained in the text. Each detected sub word is deprived of its diacritics then processed to extract feature points as explained in section 3.1.

4.2. A Technique by a Dynamic Sliding Window

The proposed approach is based on a serial-by-character recognition process using a dynamic sliding window as shown in Figure 10. The principle of the technique is based first on the recognition of the beginning and ending characters of the sub word, and then we identify the possibly remaining middle characters of the chain that composes the sub word. Thus for each sub word target image, we do the following treatments:

- For each model of a beginning character stored in the dictionary, we compute the width of a dynamic window which is taken at the right side of the image. Then we compute for each windowed image portion, the corresponding accumulator array. Finally, the recognition of the beginning character is done according to the recognition procedure detailed in section 3.3.
- For each model of an end character stored in the dictionary, the same dynamic windowing technique is repeated from the left side of the sub word image to identify the end character.
- For each model of a median character stored in the dictionary, the same dynamic windowing technique is repeated from the left limit of the beginning character to identify the first middle one. Then we slide progressively the dynamic window toward the left and according to the left limit of the last detected middle character until we reach the right limit of the end character.

At the end of the windowing procedure, all the chain of characters composing the sub word is identified [9].

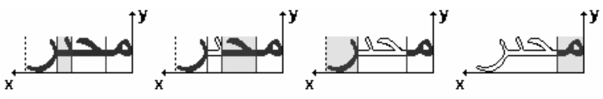


Figure 10. Use of the dynamic sliding window technique.

4.3. Identification and Localization Technique

As already explained in the previous section, the GHT can be used in OCR not only to recognize characters but also to localize their position in the image. Our aim is then to explore this specific property for the recognition of printed Arabic cursive writing without having resort to a segmentation stage. Thus, we propose to apply the GHT to the whole sub word image instead of windowed portions of the image. The obtained GHT accumulators corresponding to the different dictionary models will finally allow us to extract the characters composing the sub word and estimate their relative positions.

The proposed approach includes then a stage of identification and localization of the characters in the sub word and a post processing stage to make decision.

4.3.1. Characters Identification and Localisation in Sub Words

Given a sub word, we construct for each model of character stored in the dictionary, the corresponding accumulator array according to the method explained in section 3.3. Each obtained accumulator is merely a re-transcription of the sub word image where are consigned the votes percentages for each pixel. We define the percentage of votes Pv_{ki}, for a pixel Pi as follows:

$$Pv_{ki} = \frac{Number\ of\ feature\ points\ that\ voted\ for\ Pi}{Number\ of\ feature\ points\ of\ the\ considered\ model\ K} \quad (5)$$

The votes permit to recover the possible location of the reference point(s) of the considered character model (Figure 11). When a percentage is over then a preset threshold, the model of the character is retained as well as the position of its reference point PR_k in the image. Thereafter, we do an adjustment of the percentages of votes for the recorded reference point like follows:

$$P_{PR_k} = \frac{Number\ of\ feature\ points\ that\ voted\ for\ PR_k}{Number\ of\ feature\ points\ present\ in\ F_k} \quad (6)$$

where F_k is a window centered in the reference point PR_k of the presumed character and having the width of its corresponding model k.

Finally, we obtain for each sub word, a list of propositions of characters, their presence rates and their positions. We show through Figure 11 the possibility to detect and localize the reference points of the characters composing the target image: on the one hand, Figure 11-a illustrates the presence of character “ـحـ” twice at the beginning and in the middle of the sub word “جحد” and on the other hand the Figure 11-b demonstrates the presence of character “ـد” in the end of the same sub word [14].

4.3.2. Post Processing

In this stage we verify a set of contextual rules which permits to eliminate every contradictory proposition according to the shape of the character and its corresponding position in the sub word. If multiple propositions are candidates for the same position, we retain the one that presents the highest percentage of votes.

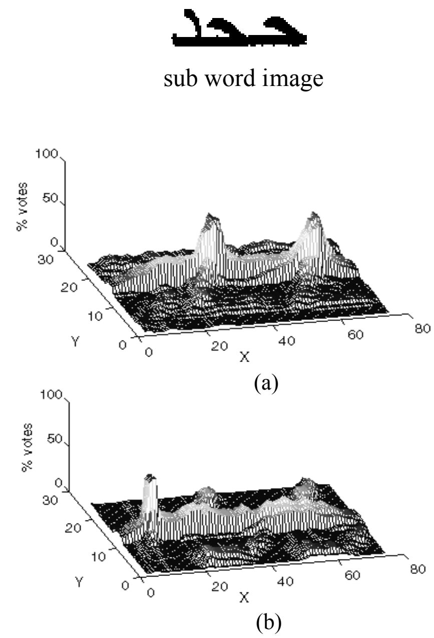


Figure 11. Accumulators array of the sub word «جحد » obtained: (a) using the model « - ـحـ» (two visible peaks indicating the sites of the two characters) and (b) using the model «ـد ».

5. Experiments

The tests concern on the one hand, Arabic printed characters recognition in different fonts and sizes using the technique described in section 3 and on the other hand, cursive printed writing recognition using the two different techniques described in the previous section.

5.1. Characters Recognition

- *Experiment 1:*

Tests have been conducted on a set of 166 873 samples of Arabic characters in Arabic Transparent font scanned at 300 dpi. Diacritics are eliminated automatically and characters are taken in their different shapes (at the beginning, in the middle, at the end and isolated). A recognition success average rate of 93% is obtained [11] as shown in Figure 12.

The survey of the different errors showed that they are generally due to the scattering of votes in accumulators around the reference point. This dispersion is a normal consequence of characters distortions. To deal with this problem, we decided to accumulate for every point in the picture, the number of votes that it got in its neighbourhood. Hence, every characteristic point must vote for a cell or one of its eight neighbours only once. This modification permitted us to optimize the evaluation of the reference point position and then to improve the recognition average rate that reached 99% [12].

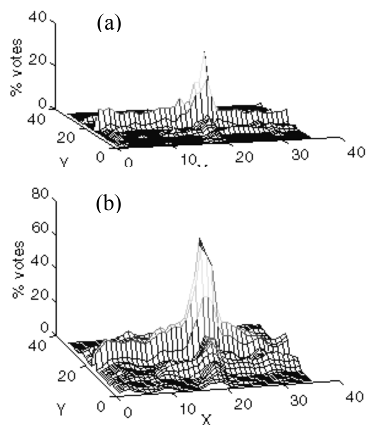


Figure 12. Accumulator array corresponding to model "ط" obtained by using a sample of the same character : (a) before and (b) after optimization.

- *Experiment 2:*

The Generalized Hough Transform is also able to detect characters in different fonts by extending the dictionary's R_Table set. We have tested on a set of 234868 characters in their isolated form in three different fonts: Arabic Transparent, Andalus and Traditional Arabic; the recognition success average rates obtained are around 97%. But we noticed that enlarging the dictionary by adding characters from other fonts, increased computational costs. Therefore a preliminary detection of fonts Proves to be necessary [11].

- *Experiment 3:*

Tests have also been made in a set of about 1000 samples of multiple sizes mixed isolated characters in Arabic Transparent font. The tested sizes vary from 12 pt to 20 pt. Recognition success rate obtained is 86% [10].

5.2. Cursive Printed Writing Recognition

The following tests have been made on a set of Arabic texts in Arabic Transparent font including about 6400 characters with variable sub words lengths going until 10 successive characters. Documents were scanned at 300 dpi and diacritics were eliminated automatically.

First experiments used the serial windowing technique. The obtained recognition success rate was 93%. The main registered mistakes were generally due to the touching that can take place between successive sub words (Figure 13). In fact, in these cases, end characters met in the middle of the studied sub word, disrupt the middle characters searching process.

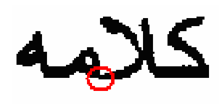


Figure 13. Example of touching between successive sub words.

Experiments have also been conducted to recognize cursive printed writing using the identification and the localisation technique. We have obtained a recognition success rate of 97% which shows the capacity of the technique to overcome the constraints of serial treatments [14].

6. Conclusion

The ultimate objective of this study was to use the HT in order to improve the recognition rates of the Arabic printed OCR and in particular the recognition of the cursive printed writing. This study has led us to explore the specific properties of the GHT. In fact, the GHT is able to detect arbitrary shapes with accuracy. In addition, it localizes their position in the target image.

In the first step, we used the edge orientation of the different Arabic characters taken in their different positions (at the beginning, in the middle, at the end and isolated) to build a structure called dictionary. Hence, one ore several labelled dictionaries could be built according to the chosen font. In the recognition step, we showed that the GHT could be adapted to the recognition of characters having either the same size or different sizes. Several tests have been conducted to show the efficiency of the method in the recognition of Arabic printed characters in different fonts and sizes.

In the second step, we have presented two approaches for cursive printed writing recognition using the GHT. The first one used a technique by a dynamic sliding window. The second one was based on the identification and the localisation of the characters in the sub word. In spite of the fact that the two approaches didn't have resort to a segmentation stage, they can be applied to a non limited vocabulary.

However the first method was more sensitive to any mistake during the serial-by-character recognition process. Meanwhile, the second method appeared more efficient. On the one hand, it remedies to the cases of touching between sub words in the text and on the other hand, it is less sensitive to mistakes because the recognition process is independent of the characters position in the sub word. Finally, this study can be considered as a step forward towards the AOCR field.

In addition, the first obtained results are encouraging and incite to deepen this study and enlarge tests.

References

[1] Ben Amara N., "On the Problematic and Orientations in Recognition of the Arabic Writing," *in Proceedings of French-Speaking International Symposium on the Writing and the Document (CIFED'02)*, Hammamet, Tunisie, pp. 1-10, 2002.

[2] Ben Amara N. and Bouslama F., "Classification of Arabic Script Using Multiple Sources of Information: State of the Art and Perspectives," *Special Issue on Multiple Classifiers, International Journal on Document Analysis and Recognition (IJDAR'03)*, pp. 195-212, 2003.

[3] Fakir M. and Sodeyama C., "Recognition of Arabic Printed Scripts by Dynamic Programming Matching Method," *IEICE Transactions* on Information Systems, vol. E76-D, no. 2, February, 1993.

[4] Fakir M. and Sodeyama C., "On the Recognition of Arabic Scripts Using Features Selected in Hough Transform Space," *Advanced Session of IIEEJ on Document Processing and Digital Imaging*, pp. 231-234, 1992.

[5] Goto H. and Azo H., "Designing Efficient Hough Transform by Noise Level Schaping," *IEICE Transactions Computer Science and Systems*, vol. E83-D, no. 2, 2000.

[6] Maitre H., "A Panorama of the Hough Transform," *Signal Processing*, vol. 2, pp. 305-317, 1986.

[7] Maharatna K. and Banerjee S., "A VLSI Array Architecture for Hough Transform," *Pattern Recognition*, vol. 34, pp. 1503-1512, 2001.

[8] Ruiz-Pinales J. and Lecolinet E., "Cursive Handwriting Recognition Using the Hough Transform and a Neural Network," *in Proceedings of ICPR'00*, pp. 231-234, 2000.

[9] Touj S., Ben Amara N., and Amiri H., "Recognition of Arabic Printed Writing by a Segmentation by Recognition Approach Based on the Hough Transform," *Sciences, Electronic, Technology of Information and Telecommunications (SETIT'2003)*, Sousse, Tunisie, 2003.

[10] Touj S., Ben Amara N., and Amiri H., "Use of the Generalized Hough Transform for Printed Arabic Characters Recognition," *Signals, Systems, Decision & Information Technology (SSD'2003)*, Sousse, Tunisie, 2003.

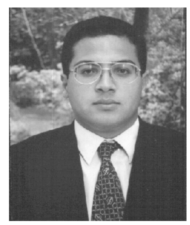
[11] Touj S., Ben Amara N., and Amiri H., "Generalized Hough Transform for Arabic Optical Character Recognition," *International Conference on Document Analysis and Recognition (ICDAR'2003)*, pp. 1242-1246, Edinburgh, Scotland, August 2003.

[12] Touj S., Ben Amara N., and Amiri H., " Two Approaches Based on the Hough Transform for the Recognition of Arabic Printed Characters," *Sciences, Electronic, Technology of Information and Telecommunications*, Sousse, Tunisia, 2004.

[13] Touj S., " A Method of Skew Correction of Digitalized Texts," *Postgraduate Diploma*, National Engineers School of Tunis, November 1999.

[14] Touj S., Ben Amara N., and Amiri H., "Printed Arabic Writing Recognition Using Generalized Hough Transform," *in Proceedings of French-Speaking International Symposium on the Writing and the Document (CIFED'04)*, la Rochelle, France, June 2004.

[15] Zarrouk S., "General Recognition of Arabic Written PAWs ," *Postgraduate Diploma*, National Engineers School of Tunis, October 1998.



Sofien Touj received his BSc and MSc degrees in electrical engineering, signal processing and systems analysis from the National Engineers School of Tunis, Tunisia in 1997 and 1999, respectively. Since 2001, he has been studying toward a PhD degree at the laboratory LSTS of the National Engineers School of Tunis, Tunisia. In February 2000, he joined the Electrical Engineering Department of the Superior institute of Technological Studies of Gabes, Tunisia, as an assistant. In September 2004, he joined the Superior Institute of Computer Science. His research interests include mainly optical character recognition applied to Arabic printed texts, and the use of the Hough transform.

Najoua Ben Amara received her BSc, MSc, and PhD degrees in electrical engineering, signal processing and system analysis, from the National School of Engineer of Tunis, Tunisia, in 1985, 1986 and 1999, respectively. From 1985 to 1989, she was a researcher at the Research Institute on Informatics and Telecommunication IRSIT, Tunis, Tunisia. In September 1989, she joined the Electrical Engineering Department of the National School of Engineer of Monastir, Tunisia, as an assistant professor. In 2004, she obtained the PhD supervision degree and currently she is an associate professor at the same institution. Her research interests include mainly optical character recognition applied to Arabic documents, image processing, compression, the use of stochastic models, wavelets, and hybrid approaches in the above applications.

Hamid Amiri obtained his German electrical engneering degree in 1978, his German PhD Engineer in 1983, and his State Doctor in 1993. He was a full professor at the ENIT (Tunisia) 1987-2001. He has been a professor at the Riyadh College of Telecom and Information (KSA) since 2001. His research interests include image processing and pattern recognition.