Asian Journal of Information Technology 5(6): 609-613, 2006

New Preprocessing Methods for Hand-Written Arabic Word

Faouzi Bouchareb, Mouldi Bedda and Salim Ouchetati
Department of Electronics Laboratory of Automatic and Signal Annaba
Faculty of Engineering, Baji Mokhtar University Annaba BP 12,
ANNABA 23000 Algeria

Abstract: This study describes new preprocessing methods for hand-written Arabic word using Hough Transform and geometrical processing applied on the skeleton chain list, using characteristic points (end points, intersection points and coin points) of the word in order to estimate and correct a baseline and slant of the word. We use these algorithms in order to obtain an aligned word which can be well segmented and recognized. The results show that these methods are very powerful for most word images on the Algerian cities name database. The result image is useful for the holistic approach and segmentation approach.

Key words: Arabic word recognition, baseline correction, slant correction, smoothing image, Hough Transform, skeletonization, end points, intersection points, coin point

INTRODUCTION

The off line cursive script word recognition has got an increasing attention during the last years. Much work has been done inversed on the recognition of hand-written in Latin, Chinese and Japanese word. Although little research has interested on the Arabic hand-written word. However Arabic characters are used in several languages like Farisi, Curdu and Urdu, this area still needs thorough investigations[1].

Extensive research has reveled that the recognition of the hand-written Arabic word is a complex task, this complexity is due to several factors:

Cursive nature of the Arabic word, present an important deformation which make a segmentation a difficult task.

Since each Arabic character has two to four different forms which vary according to the character's position in the word or subword, this extend the classes to be recognized from 28 to 100.

That's why the preprocessing of the Arabic word is an important task in optical character recognition, so the preprocessing methods are used for increasing the accuracy of recognition in diverse application of OCR like address recognition in letters, or check bank.

The principal idea in this work is to provide a regular word that resembles to the printed word which can be used as an input word in OCR systems for printed word or we use the same features of the printed word when we develop our OCR system for hand-writing word .

The principal methods developed in this work are some preprocessing treatments of the word based on the Hough Transform[2] (HT) and geometrical processing applied on the skeleton of the word.

خ	ح	ج	ث	ت	ب	أ
ص	ش	س	ذ	ر	ذ	د
ق	ف	غ	ع	ظ	ط	ض
ي	و	ه	ن	م	ل	ك

Fig. 1: Arabic alphabet

Using the word skeleton is using a most information with minimum data and minimizing a calculate in our treatment.

We proceed to compute the HT of the skeleton in the neighbor of the horizontal line and we estimate the angle of the baseline as maximum of the HT function[3,4].

After correction of the baseline we use a horizontal projection to detect any reversed word

For slant correction we use characteristics points (end points, intersection points and coin points) for extraction sketeton chain list and we analyze the end point of vertical segments[5,6] of skeleton and we make some analyses on he slanted vertical segments.

HOUGH TRANSFORM

The Hough Transform is a parametric transform, transforming the set of points of the shape into a small set of parameters characteristic of the shape. Practically, it converts a difficult global pattern detection problem in the image space into a relatively easier local peak detection problem in the parameter space or Hough space. It was

Corresponding Author: Faouzi Bouchareb, Department of Electronics Laboratory of Automatic and Signal Annaba Faculty of Engineering, Baji Mokhtar University Annaba BP 12, ANNABA 23000 Algeria

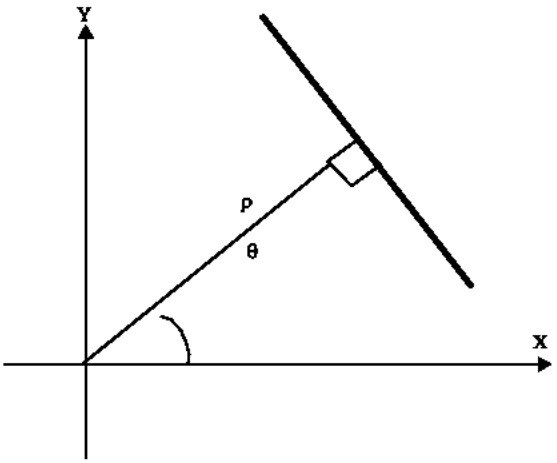

Fig. 2: (x,y) space

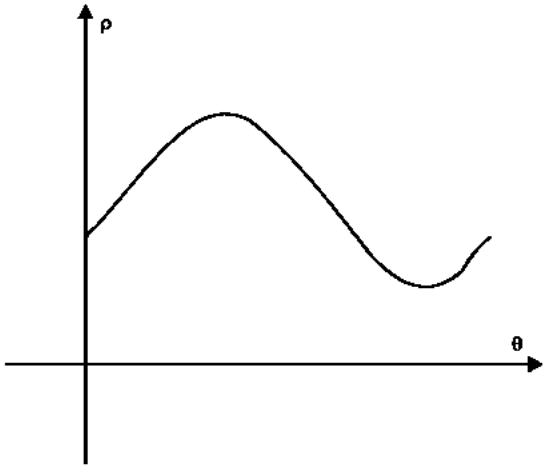

Fig. 3: (ρ,θ) space

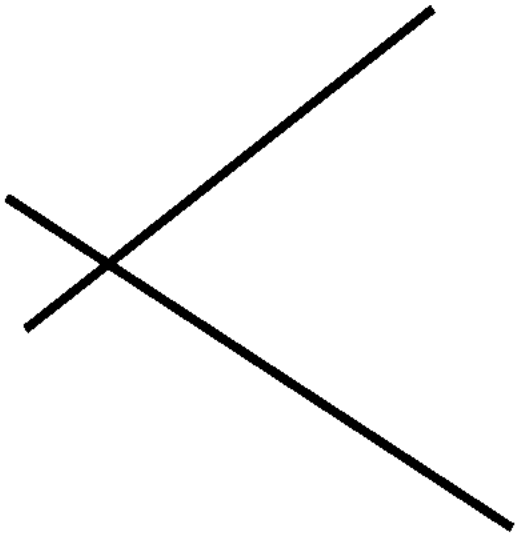

Fig. 4: image with two lines

originally used to detect straight lines in a digital image, but has been expanded to detect analytically defined curves and generalized curves.

In this work the Hough transform was used for straight lines detection.

Any straight line in the image can be represented by its parametric equation:

$$\rho = x.\ \mathrm{Cos}(\theta) + y.\ \mathrm{Sin}(\theta) \qquad (1)$$

where ρ is the normal distance to the line from the origin and θ is the angle of the normal axis.

Using this parametric equation each point (x,y) Fig. 2 is transformed into the sinusoid Fig. 3, which represent the (ρ,θ) parameters of all straight lines passing through it.

Each point on a straight line maps to different sinusoidal curve. All these curves intersect at a common point in the transform space and the (ρ,θ) parameters at the intersection are the parameters of the line they all reside on. Thus each point in the parameter space maps to a different straight line on the image space.

The Hough space, like the image, is a two-dimensional array, quantified in both,θ and ρ directions. To perform the transform ,θ is sampled in the range [0,180] for each non-zero image point the sinusoid is calculated for the sampled θ values and the resulting (ρ,θ) entries are incremented[7].

The straight line information is then extracted from the transform space by identifying the parameters at entries with high value Fig. 4 show HT of two straight lines of Fig. 5.

SKELETONIZATION

Skeletonisation is one of the important areas in image processing. It is most often ,although not exclusively,

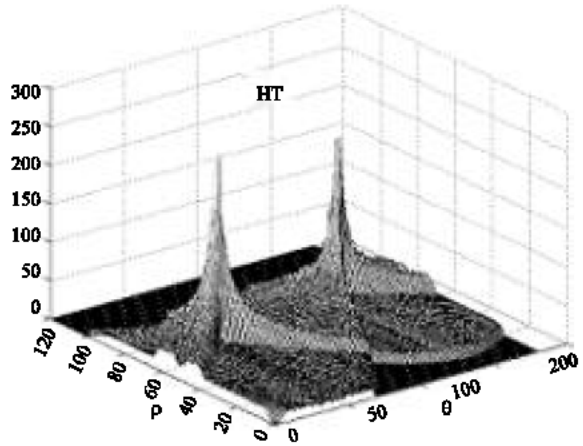

Fig. 5: HT of the image two peaks

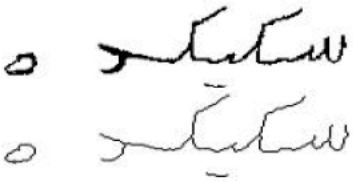

Fig. 6: the word and its skeleton

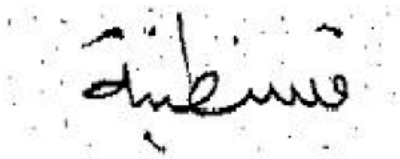

Fig. 7: original image

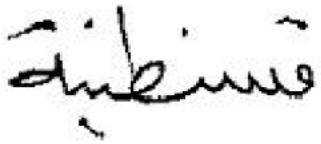

Fig. 8: smoothed image

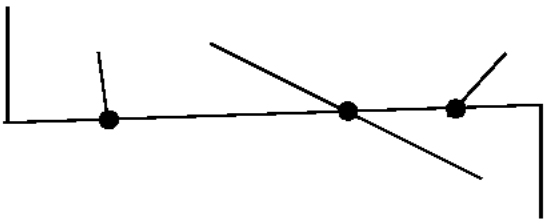

Fig. 9: Intersection points

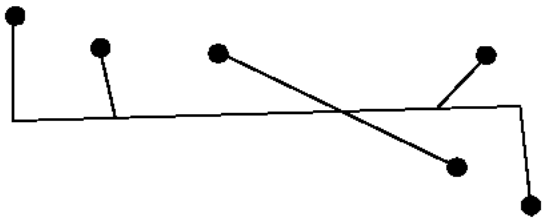

Fig. 10: end points

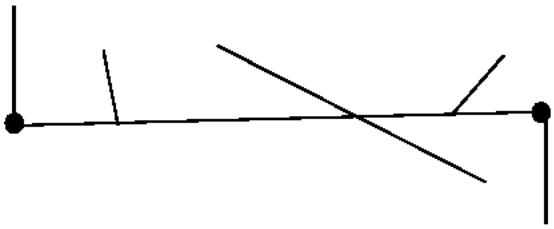

Fig. 11: coin points

used for image of hand-written or printed characters so we describe it here in this context in order to reduce data storage and increase processing speed,

Fig. 12: Input image

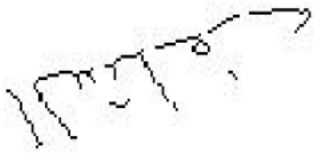

Fig. 13: Skeleton image

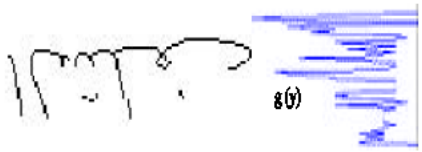

Fig. 14: Baseline correction and vertical projection

Fig. 15: Reversed Word correction

The objective of skeletonisation is to find the medial axis of a character. Most skeletonisation algorithms approximate the medial axis a unit-width binary image obtained from the original character by iteratively peeling its contour pixels until there remains no more removable pixel. The process is called Thinning and the result is the skeleton of the character, which keep most information and maintained a morphology and topology of initial word. In our work we have described a skeleton as chained list the Fig. shown an example of skeletonisation.

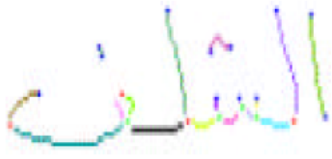

Fig. 16: skeleton chain list and characteristic points

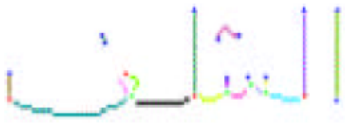

Fig. 17: Slant correction of the word

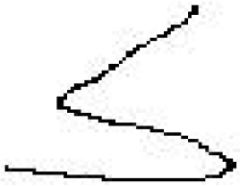

Fig. 18: kaâf character

Fig. 19: Laâmalif word

SMOOTHING IMAGE

The smoothing method used in this work based the skeleton. In the first we estimate the writing width, which depend of the pen,

Let im be initial image, sk be skeleton of the image, maxx be width of the image and maxy be length of the image. The writing width wd can be estimated by (Eq. 1)

$$wd = \frac{\sum_{x=1}^{maxx}\sum_{y=1}^{maxy} im(x,y)}{\sum_{x=1}^{maxx}\sum_{y=1}^{maxy} sk(x,y)} \quad (2)$$

The second step is to calculate the area of the every connect component in the image.

The last step is to keep only components which have area more than.

This processing was applied in order do avoid elimination of significant shape of the word like dots Fig.8.

CHARACTERISTIC POINTS

The characteristic points is used in order to extract the chain list of skeleton the skeleton in slant correction stage.

There are three types of characteristic point (intersection points, end points and coin points).

Intersection point. The point on a segment which has more than two connected branches.

End point. The point on a segment that has only one neighbor.

Coin point: The point on a segment where the pint isn't an intersection point and the curvature of the segment changes sharply.

BASELINE AND REVERSED WORD CORRECTION

The baseline correction[3] is based on HT which estimate global baseline orientation of the word by searching the maximum of the HT in the neighborhood of the horizontal. After that we rotate the word using these equations:

$$x' = x\cos(\theta) + y\sin(\theta) \quad (3)$$

$$y' = -x\sin(\theta) + y\cos(\theta) \quad (4)$$

(x,y)is the coordinate of the pixel in the initial
(x',y') is the coordinate of the pixel in corrected image
θ is the estimate base line orientation

After baseline correction we compute horizontal projection of the word and we apply statistical analyses Formulated by (eq 6) which permit us to know if the word is reversed. We rotate the word by 180°if it 's reversed.

$$g(y_0) = \max 9g(y)) \quad (5)$$

$$\sum_{y=y_0-2\cdot wd}^{y0-1} g(y) > \sum_{y=y_0}^{y0+2\cdot wd} g(y) \quad (6)$$

g(y) is the horizontal projection of the word y_0 is the coordinate of maximum of g, wd is the width of writing, estimated in smoothing image section.

SLANT CORRECTION

The aim of this stage is to detect and correct any slanted vertical strokes, in order to decrease the irregularities due of human writing. After extraction the skeleton chain list of the word limited by the characteristic points we analyze the ending points located on the upper zone of the baseline and we replace all of the points that belong to the same list of considered end point by vertical straight line and

in order to avoid undesirable deformation of characters. we use a prior knowledge about Arabic characters

Rule 1: we don't apply slant correction algorithm if the vertical angle of the segment is more than 40°. The Fig. 5 show an example of Arabic character which describe this case

Rule 2: We keep all segments which has a same intersection point. The Fig. 18 and Fig. 19 show an example of Arabic word which describe this

case. We follow this treatment by normalizing the end points in the upper zone in the same level.

EXPERIMENTAL RESULTS

Our algorithms are tested on the Algerian cities name database which belongs more than 14000 words and sub words.

The example of all algorithms described in this work is shown in Fig. 5.

The results show that these algorithms provide an regular word which can be compared with a printed word so they facilitate the segmentation and recognition task.

CONCLUSION

In this study we have described a new algorithms of preprocessing for Handwriting Arabic word .we have use Hough Transform to estimate and correct the baseline and we have used horizontal projection in order to detect any reversed word. In our work we have described a skeleton as chained list using characteristic points (end points, intersection points and coin points) used for slant correction. this preprocessing methods perform segmentation task which can be use some information extracted in preprocessing stage like characteristic points and baseline the slant correction decrease a shape variability of the same character written by different person. which facilitate training stage of recognition task

REFERENCES

1. Bozinovic, R.M. and S.N. Srihari, 1989. Off-line Cursive Script Word, IEEE Trans.PAMI, pp: 68-83.
2. Hough, P.V.C., 1962. Methods and Means for Recognizing Complex Patterns U.S. Patent 069654.
3. Tsuruoka, S., N. Watanabe, N. Minamide, F. Kimura, Y. Miyake and M. Shridhar, 1995. Base line correction for handwritten word recognition IEEE Trans ICDAR.
4. Changming Sun and Deyi Si, 1997. Skew and slant correction for document images using gradient direction Document Analysis and Recognition,., Proceedings of the Fourth International Conference on pp: 142-146.
5. Narima, Z., R. Messaoud and B Mouldi, 2003. Neuro-Markovian hybrid system for handwritten Arabic word recognition IEEE Trans ICECS, pp: 14-17.
6. Freeman Shape, H., 1978. Description Via The Use of Critical Points. *Pattern Recognition*, 10, pp: 159-166.
7. Duda, R.O. and P.E. Hart, 1972. Use of the Hough Transform to Detect Lines and Curves in Pictures, Comm. ACM, pp: 11-15,().