A new Approach for Online Arabic Handwriting Recognition

Monji Kherallah Lobna Haddad Adel M. Alimi

University of Sfax.
REGIM: Research Group on Intelligent Machines
{monji.kherallah, lobna.haddad, adel.alimi} @ieee.org

Abstract

One of the most promising methods of interacting with small portable computing devices, such as personal digital assistants, is the use of handwriting. In order to make this communication method more natural, we proposed to visually observe the writing process on ordinary paper and to automatically recover the pen trajectory from numerical tablet sequences. On the basis of this work we developed handwriting recognition system based on Freeman codes similarity. The modelling system is based on beta-elliptical representation. Our experimentations have been released using ADAB dataset. In this paper we will present the different steps of the handwriting recognition system. The results obtained are promising.

1- Introduction

In most handwriting recognition systems, the processing steps of segmentation, recognition, decision-making and post processing are serially taken. These systems usually use the resources exhaustively in each stage of the serial engine. Every stage is tuned to maximize global performance. General processing steps of handwriting document recognition are pre-processing segmentation, recognition, decision making and post processing. Pre-processing is primary related to word processing operations such as normalization to remove irregularities of handwriting. Recognition is the application of classification algorithms. Independent contextual information is used to reorganize and enhance as a post processing step.

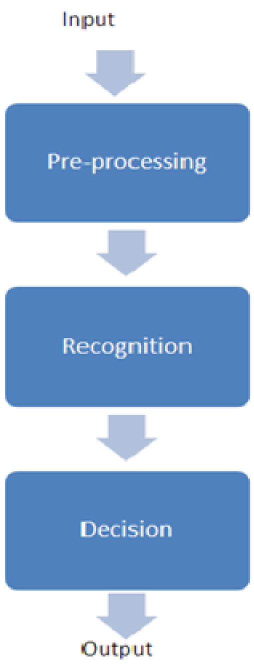


Figure 1: Different steps of the recognition system

Most architectures of handwriting recognition system are monotonically linear and based on stochastic models like Hidden Markov models [HAF 04]. These models are absolutely probabilistic and need powerful calculators and a considerable calculation time. However, in the structural method, a set of basic strokes are usually selected as primitives [ALI 02, ROK 05, CHA 07, KHE 08, KHE 09], and stroke recognition is based on the use of certain geometrical features like line segment directions, stroke length, stroke order, stroke number, stroke relation, etc.

In our work, the features extraction method was based on the Beta-elliptical model consisting of a combination between geometry and kinematics in handwriting generation movements [KHE 08]. The next step is consisting of trajectory segmentation by a sequence of strokes. We attribute every stroke to one Freeman direction. The last step of the recognition system is based on the similarity measurement between two word strings.

The next section presents the pre-processing and the encoding system. Section 3 gives the details of the recognition system. Experiments and results are given in section4.

2- Pre-processing and encoding system

A smoothing operation is first applied to the data provided by the tablet to eliminate the hardware imperfections, the trembles in writing, etc. A filtering step is necessary to eliminate duplicated date points by forcing a minimum distance between consecutive points. We use the Chebyshev filter because it has an acceptable stability in pass-band and the band of transition is large (figure 2).

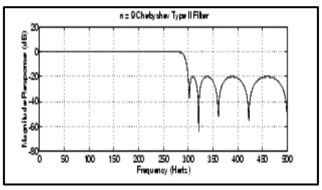


Figure 2: Pass-low filter

Figure 3 presents an example of an extracted word from the ADAB database. Note that the coordinates of all the trajectory points are considered in our work. The handwritten words are pre-classified by their number of strokes.

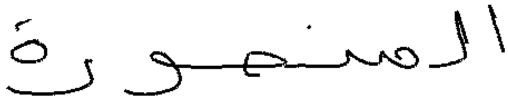


Figure 3: Tunisien region nom “ELMANSOURA”

The number of strokes was determined automatically from the curvilinear velocity representation of handwritten character (figure 4).

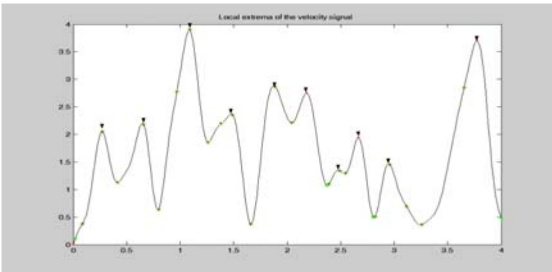


Figure 4: Velocity signal of the previous word

Note that the velocity is calculated by this equation:

$$V_{\sigma}(t)=\sqrt{\left(\frac{dx(t)}{dt}\right)^{2}+\left(\frac{dy(t)}{dt}\right)^{2}} \qquad (1)$$

Handwritten scripts are segmented into simple movements, as already mentioned, called strokes, and are the result of a supreme-position of time-overlapped velocity profiles.

The curvilinear velocity of each individual stroke obeys the Beta approach. So, the generation of a complex trajectory pattern is the result of an algebraic addition of stroke velocity terms (see equation 2).

$$V(t)=\sum_{i=1}^{n} V_i(t-t_i) \qquad (2)$$

In our case, we locate the extreme of the velocity signal of the Arabic word trajectory. The example given in figure 4 shows the velocity signal of the word “ELMANSOURA”. The inflection point trajectory with maximum and minimum of velocity signals are located. These points are considered of significance (see figure 4). Based on the inflection points given by the acceleration of the pen movement, we attributed the overlapped form of Beta to the velocity extremum profile. Therefore, after being calculated, the velocity profile of handwriting will be modulated by the Beta signal.

Consequently, the complete velocity profile of the neuromuscular system will be described by a Beta model [KHE 08] as follows:

$$\beta(t,q,p,t_0,t_1)=\begin{cases}\left(\frac{t-t_0}{t_c-t_0}\right)^{p}\left(\frac{t_1-t}{t_1-t_c}\right)^{q} & \text{if } t\in[t_0,t_1]\\ 0 & \text{if not}\end{cases}$$

(3)

Where p, q are intermediate parameters which have an influence on the symmetry and the width of Beta shape (see figure 5b).

t_0 is the starting time of Beta function,

t_c is the instant when the curvilinear velocity reaches the amplitude of the inflexion point,

t_1 is the ending time of Beta function,

$t_0 < t_1 \in$ IR and

$$t_c=\frac{p\times t_1+q\times t_0}{p+q} \qquad (4)$$

One Beta signal can be represented as shown in figure 5a.

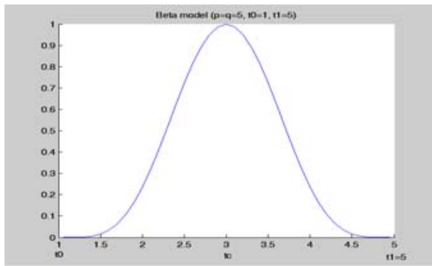


Figure 5a: One Beta signal

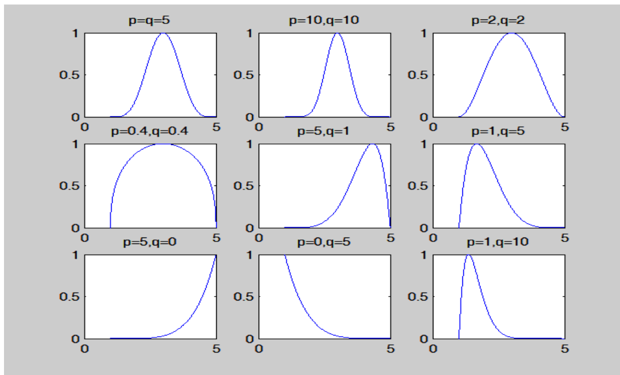


Figure 5b: Different shapes of the mono-dimensional Beta function

The parameter k is the amplitude of the beta signal (k=1 in this case). The different shape of Beta function represented in figure 5b depends on p and q values.

Dynamic data contains the information about how the shapes were written. Static data conveys the result of the writing process i.e. what has been written.

In fact, a stroke executed from an arbitrary starting position is characterized by three parameters. These parameters are collected from the Beta function. Each elementary component called “stroke” is also characterized in the space domain by three statistical parameters. These parameters globally reflect the geometric properties of the set of muscles and joints used in a particular handwriting movement. The parameters a and b are respectively the half dimensions of the large and small axes of the elliptic shape. X0 and Y0 are the Cartesian coordinates of the elliptic center relative to the orthogonal reference (A, X, Y).

As shown in figure 11, the angle θ defines the deviation of the elliptic portion according to the orthogonal reference (A, x, y).
From two points (A, B) of one stroke, we calculate the (θ, a, b) parameters. These points A and B correspond respectively to the minimum and the maximum values of the velocity profile. C is rightly before the point B, we join a tangent line crossing B and C. By an orthogonal projection on B, we get the center O and the different axes a and b of the ellipsis (see figure 6).

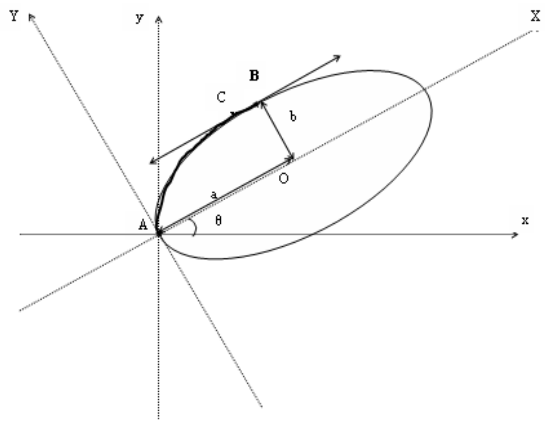


Figure 6: Elliptic arc representation of one stroke shapes

Consequently, a single movement which is called stroke is represented in the space and velocity domains by a curvilinear velocity starting at time t_0 at an initial point, and moving along an elliptic path.
Elliptic model is a static model. In the spatial state, the trajectory is represented by a sequence of elliptic arcs [10, 33].
The elliptic equation is written as follows:

$$x^2/a^2 + y^2/b^2 = 1 \qquad (5)$$

Each elliptic arc is drawn by the calculation of (θ, a, b) parameters.

The second step of the features extraction system is to extract the visual codes that correspond to the elliptic strokes obtained through the Beta-Elliptic modeling of handwritten script [KHE 09]. According to the Freeman code method, we attribute for every stroke one direction and one visual code.

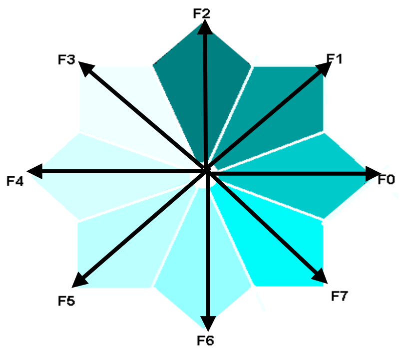
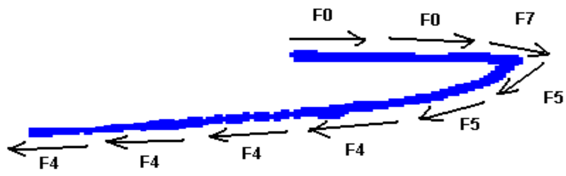


Figure 7: Freeman codes

The succession of Freeman's codes generates easily a string of codes describing the different entities. These entities form the Arabic word [JOU 03] . The visual representation is a sequence of Freeman's codes from which we quote for example the Arabic letter "Ja" and it is coded as follows (F0, F0, F7, F5, F5, F4, F4, F4, F4), that gives an approximate form of the letter (see figure 7 and figure 8).



F0	**F0**	**F7**	**F5**	**F5**	**F4**	**F4**	**F4**	**F4**

Figure 8: Example of Freeman codes of letter "JIM"

The last step of the encoding system is based on Freeman codes inspection, we obtain a new string containing the number of every Freeman code (see figure 9).

2	**0**	**0**	**0**	**4**	**2**	**0**	**1**

Figure 9: Freeman code inspection of the letter "JIM"

3- Recognition system

Many developed recognition systems are presented in literature. These systems are manly based on Neural Networks, Hidden Markov Models, Genetic Algorithms, Fuzzy approaches, etc. Inspired from the Graph matching algorithm and the Euclidian distance calculation [ROK 05], we developed a new method which consists of similarity calculation between 2 strings. This similarity is calculated by the following equation.

$$S = \sum_{i=1,j=1}^{8}(B_i - B_j) \qquad (6)$$

Bi and Bj are respectively the waits of every bit of string i and j. We proceed to calculate the similarity rate between two Arabic words.
The similarity function is the outcome of the comparison between two handwritten Arabic words (tested word and either a set of words). This process is carried out through the calculation of the similarity degree between the Freeman code strings of the concerned words. According to equation 6, this similarity is measured by paralleling and comparing the different bits. The calculation of the

fitness value; therefore, is based on the sum of the similarity measurement between two Freeman code strings.
In order to decrease the executed time and the calculation complexity, we proceed to divide the learning dataset of ADAB in to 5 sets. Every set is specialized on stroke number interval (see figure10).

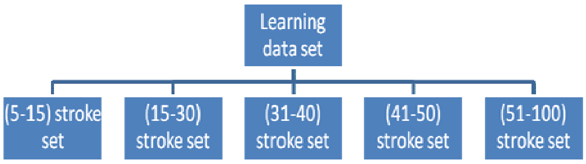


Figure 10: Hierarchical representation of the ADAB dataset

4- Experimental Results

In order to evaluate the performance of our system, 24 participants are invited to contribute to the handwriting data construction for our recognition experiment. The data set of words of each participant is stored in one data file. The Arabic words of ADAB dataset are chosen identically to IfN/ENIT dataset words. The difference between them is that our dataset contains both the on and the off-line information. When producing the data file, each participant is asked to write some Arabic words. We collected 34500 words written by different writers. 20000 words are used as data prototypes, the others are used for testing our system.
The execution time is variable and the average of the recognition rate is about 91%.

5- Conclusion

In the first stage of our work, we proceed to use the beta elliptical modelling of the handwritten trajectory. This segmentation of the handwritten word is considered as a low level segmentation. Every stroke is represented by one Freeman direction. Our recognition system is based on similarity score. The recognition rate obtained is about 91%. Some other confusions between Arabic words have been noticed when the Freeman code strings of the words present a big similarity. Compared to other systems as HMM or Graph matching, our system performs better. However the weakness of our system resides in execution time.

Acknowledgements

The authors thank Prof Volker Margner and Dr ElAbed Haikal for their assistance and invaluable help. In addition, they acknowledge the financial support of this work by grants from the General Direction of Scientific Research and Technological Renovation (DGRST), Tunisia, under the ARUB program 01/UR/11/02.

References

Alimi, M. A. (2002). “Evolutionary Computation for the Recognition of On-Line Cursive Handwriting”. IETE Journal of Research, Special Issue on "Evolutionary Computation in Engineering Sciences” edited by S.K. Pal et al., vol. 48, no. 5, pp. 385-396.

Charfi M, Kherallah M., El Baati A and Alimi A.M. (2007),. « Système de reconnaissance automatique des adresses manuscrites par une representation Beta-elliptique » , Journal of Computer Science and Engineering in Arabic (JCSEA), Volume 1- N°1, pp 56-78, See errata, Volume 1- N°2.

Kherallah M., Hadded L., Mitiche A., Alimi A. M. (2008). “On-Line Recognition Of Handwritten Digits Based On Trajectory And Velocity Modelling”. International journal of Pattern Recognition Letter. Vol. 29. pp. 580-594.

Kherallah M., Bouri F., And Alimi M. A. (2009). “On-Line Arabic Handwriting Recognition System Based On Visual Encoding And Genetic Algorithm.” Engineering Applications of Artificial Intelligence. Vol. 22 . pp 153–170.

Kherallah, M and Alimi M.A, (2008). “A new Lecture Support Based on On-line Arabic Handriting Recognition”, in the International Conference on Innovations in Information Technology. Innovation’08.

Rokbani N., Kherallah M., and Alimi M. A. (2005) “ On-Line Handwriting Recognition System based on Graph matching and Genetic Algorithm”. Proc.Int. ACIDCA-ICMI’2005, pp. 959-963. Tozeur, Tunisie.

Jouini B., Kherallah M., Alimi A.M. (2003). “ A new approach for on-line visual encoding and recognition of handwriting script by using neural network system”. Book chapter, David W. Person. Ed. Artificial Neural Nets and Genetic Algorithms. Spring, Wien, pp.161-167. Rouane, France.

Hafsa, W., Kherallah, M., Jemaa, M. B., and Amara,N. E. B. (2004). “Une approche hybride neuro-markovienne pour la reconnaissance en ligne des digits Arabes,”. In Proc. IEEE Int. Conf SCS’04. pp.137-141. Monastir, Tunisie.