UNIVERSITI SAINS MALAYSIA

Laporan Akhir Projek Penyelidikan Jangka Pendek

Investigation on The Usefulness of Bioinformatics Application Techniques For Arabic Character Recognition

by
Assoc. Prof. Umi Kalthum Ngah
Mr. Moayad Yousif Potrus

2013

F0484

BORANG FRGS – P3(R)

Kod Projek :	FRGS/FASA1-2009/(BIDANG)/(NAMA IPT)/(NO.RUJ. KPT)

FINAL REPORT
FUNDAMENTAL RESEARCH GRANT SCHEME (FRGS)
Laporan Akhir Skim Geran Penyelidikan Asas (FRGS) IPT
Pindaan 1/2009

A

RESEARCH TITLE : Investigation on the Usefulness of Bioinformatics Application Techniques for Arabic Character Recognition
Tajuk Penyelidikan

PROJECT LEADER : Assoc. Prof. Umi Kalthum bt. Ngah
Ketua Projek

PROJECT MEMBERS (including GRA) : 1.Moayad Yousif Potrus
Ahli Projek

PROJECT ACHIEVEMENT *(Prestasi Projek)*

B

ACHIEVEMENT PERCENTAGE			
Project progress according to milestones achieved up to this period	**0 - 50%**	**51 - 75%**	**76 - 100%**
Percentage			100%

RESEARCH FINDINGS

Number of articles/ manuscripts/ books	**Indexed Journal**	***Non-Indexed Journal***
Paper presentations	**International**	***National***
	2	1
Others (Please specify)	IEEE International Conference, included in proceeding but could not attend to present	

HUMAN CAPITAL DEVELOPMENT

Human Capital	Number		Others (Please specify):
	On-going	Graduated	
PhD Student		1	
Masters Student			
Undergraduate Students		1	
Temporary Research Officer			
Temporary Research Assistant			
Total		2	

EXPENDITURE *(Perbelanjaan)*

C

Budget Approved *(Peruntukan diluluskan)* : **RM40000.00**
Amount Spent *(Jumlah Perbelanjaan)* : **RM37682.06**

Balance *(Baki)* : **RM2,317.94**
Percentage of Amount Spent : 94.2%
(Peratusan Belanja)

ADDITIONAL RESEARCH ACTIVITIES THAT CONTRIBUTE TOWARDS DEVELOPING SOFT AND HARD SKILLS
(Aktiviti Penyelidikan Sampingan yang menyumbang kepada pembangunan kemahiran insaniah)

D

International

Activity	Date (Month, Year)	Organizer
(e.g : Course/ Seminar/ Symposium/ Conference/ Workshop/ Site Visit)	-	

National

Activity	Date (Month, Year)	Organizer
(e.g : Course/ Seminar/ Symposium/ Conference/ Workshop/ Site Visit)	-	

PROBLEMS / CONSTRAINTS IF ANY *(Masalah/ Kekangan sekiranya ada)*

-

-

RECOMMENDATION *(Cadangan Penambahbaikan)*

F

-

RESEARCH ABSTRACT – Not More Than 200 Words *(Abstrak Penyelidikan – Tidak Melebihi 200 patah perkataan)*

G

Rapid advancement in mobile and tablet devices technology has enabled the use of text-based applications such as chatting, messaging and text editing applications. However, hardware or software need to be hardwired in their keypad which limits their capabilities. Arabic language which is used by a great portion of people on social and public internet applications varies in its writing style varies making it difficult for automated recognition. An online Arabic handwritten text recognition system based on Hidden Markov Model (HMM), Genetic Algorithm (GA) and Harmony Search (HS) concentrates on the use of heuristic search technique to deal with problems related to segmentations and writer independency. For the segmentation problem, the system uses threshold-based clustering, dominant point detection and dominant direction detection for the pre-segmentation strategies. The segmentation process is tested using two different approaches: segmentation-based recognition using HMM-HS and recognition-based segmentation using GA-HS. The HMM-HS segmentation-based recognition uses character classification with HMM to reduce the character numbers which are passed to the final character recognition phase. HS is then applied to find the best matched character based on the character regional matching together with the probability of the matched feature in its position for the partial character class. On the other hand, the GA-HS recognition-based segmentation uses binary GA representation to assign the segmentation points. The produced characters from each chromosome are passed one by one to the HS to determine the best-matched character with minimum score. The best-matched text is determined by the GA minimum text score obtained from the sum of the score for each of the HS character matching. For the recognition-based segmentation, three methods are used for text recognition i.e. GA-HS-All (consider all line segments for segmentation), GA-HS-FW (consider foreword line segments for segmentation) and GA-HS-DP (segmentation using dominant point detection). The average accuracy is found to be very high and time is very much shortened.

Date : 19 July 2013
Tarikh

Project Leader's Signature:
Tandatangan Ketua Projek

COMMENTS, IF ANY/ ENDORSEMENT BY RESEARCH MANAGEMENT CENTER (RMC)
(Komen, sekiranya ada/ Pengesahan oleh Pusat Pengurusan Penyelidikan)

H

Penyelidik perlu cuba menerbitkan dp dapatan penyelidikan dalam jurnal berwasit.

Name:
Nama:

PROF. MADYA LEE KEAT TEONG
Pengarah
Pejabat Pengurusan & Kreativiti Penyelidikan
Universiti Sains Malaysia
11800 USM, Pulau Pinang.

Signature:
Tandatangan:

Date:
Tarikh: 1/8/13

Purchase Requisition ▸ Purchase Order ▸ Suppliers ▸ Maintenance ▸ Financials ▸ Coda Info ▸ Reports ▸ Admin ▸

UserCode: AZURINA / USMKCTLIVE / PELECT **Program Code: Votebook9100** **Current Program : Votebook (Header)**

Update Coda Currency

Update Supplier to Coda

Elements ▸

List of Asset

Current Date : 01/07/2013 4:29:34 PM **Version: 15.03, Last Updated at 03/12/2012** **DB: 13.00, 09/18/2010 VB: 13.01, 03/14/2011** **Switch Language : English /Malay**

Wildcard : eg. Like 100%, Like 10%1, Like %1

Element 1: 203 Element 2: % Element 4: PELECT

Element 5: 6071189 Year: 2013

Detail	Excel	Budget Rule	Budget Control	Account Description	Budget Account Code	Roll over	Budget	Cash Received	Advanced	Commit	Actual	Available	Percentage
Detail	Excel	117	T	Penyelidikan Fundamentals (FGRS)	203.221.0.PELECT.6071189	4,746.82	0.00	0.00	0.00	0.00	0.00	4,746.82	0.00%
Detail	Excel	117	T	Penyelidikan Fundamentals (FGRS)	203.222.0.PELECT.6071189	-14.30	0.00	0.00	0.00	0.00	0.00	-14.30	0.00%
Detail	Excel	117	T	Penyelidikan Fundamentals (FGRS)	203.223.0.PELECT.6071189	3,500.00	0.00	0.00	0.00	0.00	0.00	3,500.00	0.00%
Detail	Excel	117	T	Penyelidikan Fundamentals (FGRS)	203.226.0.PELECT.6071189	1,500.00	0.00	0.00	0.00	0.00	0.00	1,500.00	0.00%
Detail	Excel	117	T	Penyelidikan Fundamentals (FGRS)	203.227.0.PELECT.6071189	-2,553.50	0.00	0.00	0.00	0.00	0.00	-2,553.50	0.00%
Detail	Excel	117	T	Penyelidikan Fundamentals (FGRS)	203.229.0.PELECT.6071189	-4,863.08	0.00	0.00	0.00	0.00	0.00	-4,863.08	0.00%
		117	T	SubTotal		2,315.94	0.00	0.00	0.00	0.00	0.00	2,315.94	0.00%
Detail	Excel	118	T	Penyelidikan Fundamentals (FGRS)	203.335.0.PELECT.6071189	2.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00%
		118	T	SubTotal		2.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00%
		9999		GrandTotal		2,317.94	0.00	0.00	0.00	0.00	0.00	2,317.94	0.00%

AN EVOLUTIONARY BASED ONLINE RECOGNITION SYSTEM FOR HANDWRITTEN ARABIC TEXT

by

MOAYAD YOUSIF POTRUS

Thesis submitted in fulfilment of the requirements
for the degree of
Doctor of Philosphy

December 2012

ON-LINE CHARACTER RECOGNITION USING ARTIFICIAL INTELLIGENT AND BIOINFORMATICS TECHNIQUES

LIM YEE WAN

28/04/2010

ASSOC. PROF. DR NOR ASHIDI MAT ISA (*B. Eng; PhD (USM)*)
School of Electrical and Electronic Engineering
Universiti Sains Malaysia
Engineering Campus
14300 Nibong Tebal, Penang, Malaysia.

Dengan ini disahkan bahawa segala pindaan telah dilakukan oleh pelajar berkenaan.

Tandatangan Penyelia

Nama Penyelia: DR. ICALTHUM NADZIR

UNIVERSITI SAINS MALAYSIA
2010

LIST OF PUBLICATIONS

1. Potrus, M.Y.; Ngah, U.K.; Sakim, H.A.M.; AbdulRahman, S.A.(2010). "Normalization and Rectification Method for Online Hindi Digit Recognition with Partial Alignment Algorithm," *IEEE International Conference On Electronics and Information Engineering* (ICEIE 2010), Kyoto, Japan. Vol.1, no., pp.V1-223-V1-227, 1-3 Aug. 2010.
(URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5559886&isnumber=5559664)

2 Potrus, M.Y.; Ngah, U.K.; Sakim, H.A.M. (2010). "An effective segmentation method for single stroke online cursive Arabic words,"*IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE)*, Kuala Lumpur. 2010, Vol., no., pp.217-221, 5-8 Dec. 2010.

3 Potrus, M.Y.; Ngah, U.K.; Sakim, H.A.M.(2011) "Hidden Markov Model-Harmony Search Combination for Online Arabic Character Recognition". *Electrical & Electronic Postgraduate Colloquium* (EEPC 2011). Janda Baik, Pahang.

4. Potrus, M.Y.; Ngah, U.K.(2012); , "A Harmony Search Algorithm for Recognition-Based Segmentation of Online Arabic Text," *International Conference on Engineering and Information Technology*, (ICEIT 2012) Sep. 17-18, 2012, Toronto, Canada.
(URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5735078&isnumber=5735003)

CONTENTS

ICEIE 2010 Session 5

ICEIE 2010 Session 6

classifications are isolated form, start form, middle form or end form (as shown in figure 1) [15].

These characters can be classified into 18 main groups for isolated character form, 11for start form, 9 for middle form and 16 for end as shown in table 1. This classification is very important and helpful for recognition since these groups will control the flow of recognition, as well as, minimizing the recognition error.

The problem of cursive Arabic word can be represented as follows: Consider an Arabic word W having N of (x, y) coordinate points and M characters C. The number of segments S is always less by 1 than the number of characters C. Thus, we have M-1 segments. Then,

$$(x, y)_i \in S_j \Leftrightarrow (x, y)_i \notin C_k \forall (x, y)_i \in W$$

$$\text{With } i = 1,\ldots,N,\ j = 1\ldots M-1 \quad \text{and} \quad k = 1\ldots M\ .$$

It is required to find which coordinate point belongs to the segment and which belong to the character.

Figure 2 below shows the full diagram for segmentation and recognition system. The components for our system are described in the following sections.

A. *Preprocessing*

The preprocessing phase includes the following two steps:

A.1. Smoothing

The Arabic written word is smoothed to modify and remove rough edges to obtain more enhanced writing with less noisy shape. The output will be more stable and recognizable as it gives a straight form of the character and ligature.

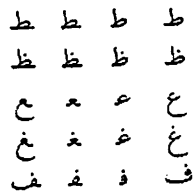


Figure 1. Some Arabic characters and their pattern based on position

TABLE I. GROUPS OF ARABIC CHARACTERS

Form	End Form(EF)	Middle Form(MF)	Start Form(SF)	Isolated Form(IF)
Characters	(ب) (ـبـ ـتـ ـثـ ـنـ) (ي) (ـجـ ـحـ ـخـ) (ـسـ ـشـ) (ـصـ ـضـ) (ـطـ ـظـ) (ـعـ ـغـ) (ـفـ ـقـ) (ـكـ) (ـمـ) (ـهـ) (ـلـ) (ـو) (ـلا ـلأ ـلإ ـلآ)	(ـبـ ـتـ ـثـ ـنـ ـيـ) (ـجـ ـحـ ـخـ) (ـسـ ـشـ) (ـصـ ـضـ) (ـطـ ـظـ) (ـعـ ـغـ) (ـفـ ـقـ) (ـلـ) (ـهـ)	(ـبـ ـتـ ـثـ ـنـ ـيـ) (ـجـ ـحـ ـخـ) (ـسـ ـشـ) (ـصـ ـضـ) (ـطـ ـظـ) (ـعـ ـغـ) (ـفـ ـقـ) (ـكـ) (ـلـ) (ـمـ) (ـهـ)	(ا) (ب ت ث ن) (ي ى) (ج ح خ) (د ذ) (ر ز) (س ش) (ص ض) (ط ظ) (ع غ) (ف ق) (ك) (ل) (م) (ه) (و) (لا لأ لإ لآ)(ء)
No. of Groups	16	9	11	18

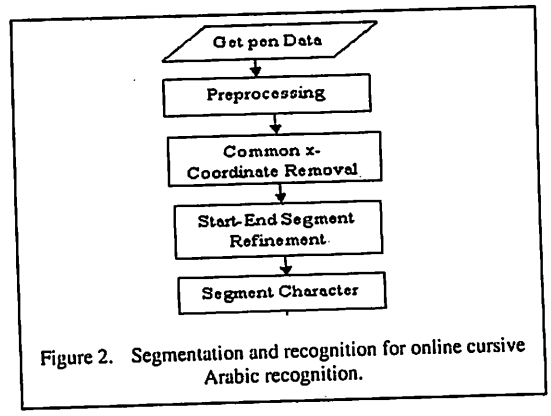


Figure 2. Segmentation and recognition for online cursive Arabic recognition.

A.2. Normalization and Rectification

The complete word will be normalized and rectified for removing errors and construct segment line based writing [16]. The following formula is used to normalize the complete word.

$$S = S - S_j \quad \forall P_j < \lambda = \frac{1}{10}\log\frac{N}{2} \tag{1}$$

Where S is the direction vector for the word, S_j is the direction vector in direction j, P_j is the probability of existence for direction j, λ is the normalization factor and N is the number of points for the word. Then, the output of this process is further rectified to remove errors.

$$S = S - S_{ij} \quad \forall S_{ij} < \varepsilon \tag{2}$$

Where S_{ij} is the partial vector i in the direction j, ε is the rectification factor (selected as 2 empirically). The output of these two steps represents a number of joint lines.

B. *Coordinate Transformation and Clustering*

The coordinate of the system is changed from (x, y) coordinate into four direction vector of Up, Down, right and Left. Then each movement group is clustered to transfer the shape into more rectangular shape.

Consider (x, y) as point coordinate belong to Arabic word W and β is angle between start and end points after y coordinate is changing direction as shown in figure 4. Then, $\forall (x, y) \in W$ we obtain

$$g \in W_{(U,D,L,R)} : g = \begin{cases} U & \Delta x = 0, \Delta y > 0 \\ D & \Delta x = 0, \Delta y < 0 \\ L & \Delta x > 0, \Delta y = 0 \\ R & \Delta x < 0, \Delta y = 0 \\ RU & \Delta x < 0, \Delta y > 0 \\ RD & \Delta x < 0, \Delta y < 0 \\ LU & \Delta x > 0, \Delta y > 0 \\ LD & \Delta x > 0, \Delta y < 0 \end{cases} \tag{3}$$

Where $W_{(U,D,L,R)}$ is the word movement vector of combined direction Up, Down, Left and Right. The result does not include diagonal movement which is transferred into equal RU, RD, LU or LD movement vector as shown in figure 3.

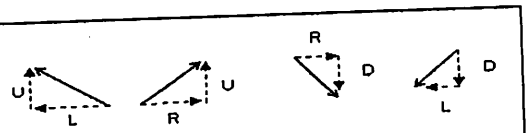


Figure 3. Changing Diagonal Movement into (U, D, L, R) vector.

In some case, the characters are very difficult to segment due to overlapping between consecutive characters' coordinates (Figure 4 (a)). Previous researches [11][13] didn't solve this particular problem due to its difficulty. We propose a method for modifying equation (3) above to exclude movements from left to right (RU and RD) to become (URU and DRD) since these movements forces overlap to occur. This transformation will reduce the length of the character movement towards R vector and increase the movement towards U or D vectors as shown in figure 4 (b).

The clustering operation intended to remove single or double movement in specific direction between two dominant different direction vectors. Let's consider g_i to be the count of consecutive movement for specific direction of U, D, L or R. Then, $G_{W(U,D,L,R)}$ consist of a set of g_{ij} i=1,2,....n and jϵ{U,D,L,R}. Then clustering is performed as;

$$\forall\, g_{ij} < 2 \text{ (empirically)}, g_{ij} = \begin{cases} g_{i-1,j} & \text{if } g_{i-1,j} \geq g_{i-1,j} \\ g_{i-1,j} & \text{if } g_{i-1,j} < g_{i-1,j} \end{cases} \quad (4)$$

Figure 5 shows the Arabic word عمل as written by the user. It has inconsistent writing shape to be segmented efficiently. Figure 5 (b) shows the character after smoothing where jittered movements are removed and the shape is more consistant. Then, after applying equation (1) through (5) the shape becomes a flat shape with no diagonal movement as shown in figure 5 (c). At this stage the character is more recognizable as character and ligature.

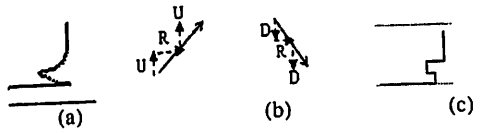


Figure 4. Changing Diagonal Movement into (U, D, L, R) vector. (a) Original overlapped characters, (b) Diagonal movement transformation, (c) Characters after preprocessing, transformation and diagonal movement transformation.

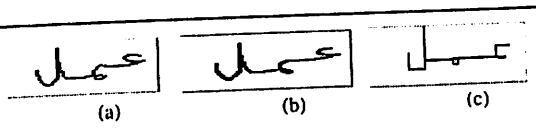


Figure 5. Preprocessing steps performed on Arabic word Arabic word عمل. (a) Original written word. (b) Word after smoothing. (c) Word after preprocessing, coordinate transformation and clustering.

C. *x-Coordinate Scan and Segmentation*

The output shape of figure 4(c) consists of line segments belong either to character body or to line joining these characters. All Arabic characters share one common feature. Character body writing style has the same x-coordinate passed through twice or more. Consider (x^e, y^e) to be the point coordinate along the Arabic word after applying equation (2) above. Then, coordinates belong to line joint segment (x^s, y^s) is obtained as:

$$(x_i^s, y_i^s) = \begin{cases} 0 & \forall (x_i^e, y_i^e) = (x_j^e, y_j^e) \\ (x_i^e, y_i^e) & otherwise \end{cases} \quad (5)$$

Where as, points belong to character body (x^c, y^c) is obtained as:

$$(x_i^c, y_i^c) = \begin{cases} (x_i^e, y_i^e) & \forall (x_i^e, y_i^e) = (x_j^e, y_j^e) \\ 0 & otherwise \end{cases} \quad (6)$$

Where i=1....N and j=1.....N.

Figure 6 (a) shows the result of extracting character body (red color) from joint lines (black color). As noted the segmentation processes so far is working very well and the lines in the black color represents the ligatures between the characters in the word. But the only error represents the selection of first character starting movement and the last character horizontal movement as ligatures instead of character body. In next section these two problems are overcome.

The segmentation points between characters are calculated as follows:

$$P_S(x_k, y_k) = \left(\frac{x_{sk}^S + x_{ek}^S}{2}, y_k^S \right) \quad k = 1...n \quad (7)$$

Where Ps(xk, yk) is the kth segment point, n is number of segments, x_{sk}^S is the starting x coordinate for kth segment, x_{ek}^S is the end x coordinate for kth segment, y_k^S is the y coordinate for kth segment. Figure 6 (b) shows the segmented characters after applying equation (5).

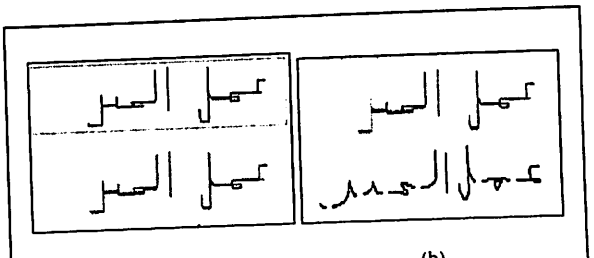


Figure 6. Results of applying (a)x-coordinate scan. Red color highlights the character body while black highlight the joint line. (b) Character segments at segment points.

D. *First and Last Segment Filtering*

Some of the character such as ع or ك in the first segment or characters such as بفلرو in the end

segment may cause segmentation error (as shown in figures 6 and 7). These segments must be filtered to remove ambiguity regarding the segment location. Two filters are used to distinguish between start or end character movement from actual segment locations. These two filters are described as follows:

D.1. Start Segment Filtering

Consider L_s to represent a straight line from segment start point (x_s, y_s) to its end and having (x_{ls}, y_{ls}) coordinate. Then, removing the segment is subject to the condition:

$$\forall x_{ls} \in L_s \exists | y_{ls} - y_s | > 0 \qquad (8)$$

Equation 8 implies that if the variant in y direction, related to first character, is 0 then the movement is in straight line. Thus, this movement is a part of the first character. Otherwise, if there is a rapid change in the movement then Δy will have a value greater or less than zero representing the ligature following the first character.

D.2. End Segment Filtering

Consider y_s to be the y coordinates of the end ligature point. x_c and y_c is the character (x, y) coordinate. then,

$$\Delta y_i = (y_c - y_s) \qquad (9)$$

The filter removes the ligature if it is subject to one of the following conditions:

$\Delta y = \Delta y_i - \Delta y_{i-1} < 0$ and $\theta < 70$ for all i=1,...N.
Or,
$\Delta y = \Delta yi - \Delta yi\text{-}1 > 0$ and $\theta < 70$ for all i=1,...N.

Where N is the number of ligature points and θ is the angle from the ligature point to the last point in the current segment as shown in figure 7. This filter will ignore final characters such as (ـه or ـل) since the first has $\Delta y > 0$ and $\Delta y < 0$ along its movement, while the second has $\Delta y > 0$ and $\theta > 70o$ along its movement. Figure 8 shows the final refined result after applying the filter to the word. It can be seen that the segmentation is done exactly at the positions that it must be.

III. Results and Discussion

The segmentation system was tested with 5 different sentence templates to insure word variation and increasing the possibilities of error to take place (figure 9). These sentences contain a wide combination of difficult segmentations such as overlap, first character error, end character error, jitter in writing.

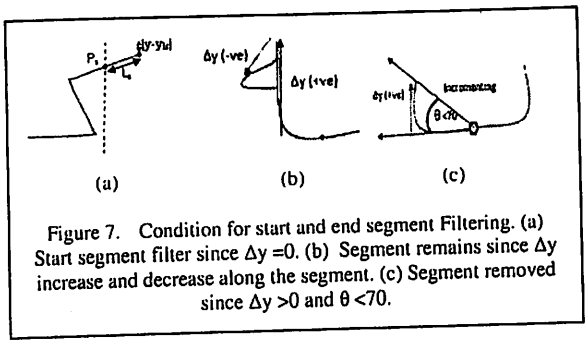


Figure 7. Condition for start and end segment Filtering. (a) Start segment filter since $\Delta y = 0$. (b) Segment remains since Δy increase and decrease along the segment. (c) Segment removed since $\Delta y > 0$ and $\theta < 70$.

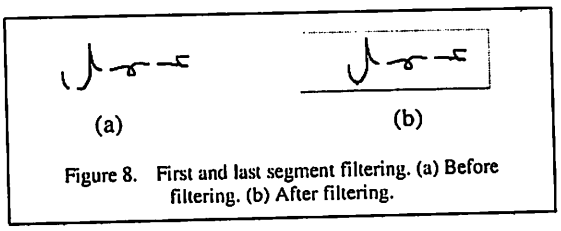


Figure 8. First and last segment filtering. (a) Before filtering. (b) After filtering.

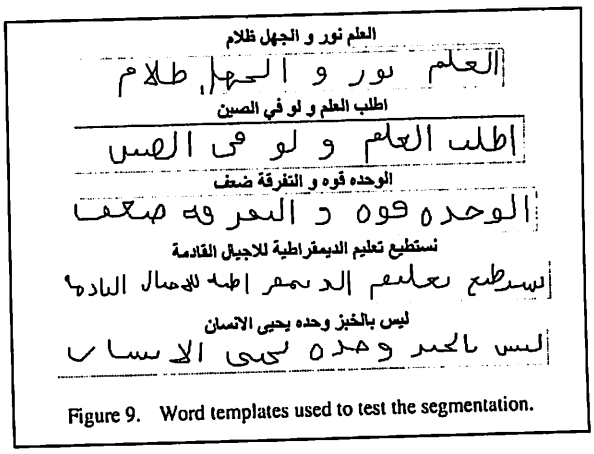


Figure 9. Word templates used to test the segmentation.

The system was tested with 20 different persons to obtain the success percentage of the segmentation depending on how many characters were segmented successfully. These users were asked to perform normal style writing with natural technique. They were also asked to write with speedy and inconsistent writing technique to test the limit of the system. Table 3 shows the result of the segmentation applied to the sentence templates. The results represent the percentage of successfully segmenting and filtering 580, 600, 200 and 1580 of first, last, overlap and normal characters included in these cursive words out of total 2300 characters.

Table 2 shows the result of segmentation using word template of Fig 9. It can be noted that normal character has scored the highest percentage (98.03%) with only characters such as س-ص that contain "rogza" will affect the recognition rate due to ambiguity of separating ligature as it match the character body. The first character filter obtained 94.82% errors occurs due to overlapped segment or miss segment of س-ص. The Last character filter obtained 97.33% with the same reason. The overlapped characters scored 90% with errors mostly occurred due to high overlapping cases as illustrated in figure 10.

The system initially has achieved a very promising result to be used as a first step in online Arabic words recognition system.

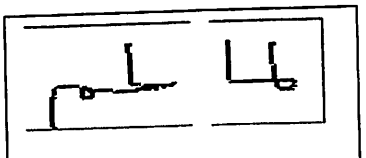


Figure 10. Difficult overlapping cases

TABLE II. TEST RESULT FOR THE TEMPLATES IN FIGURE 10

Group	Total successful segments	Total miss segments	Percentage
Normal	1549	41	98.03%
First Character filter	550	10	94.82%
Last Character filter	584	16	97.33%
Overlapped	180	20	90%
All Characters	2213	87	96.2%

IV. CONCLUSION AND FUTURE WORK

We presented a technique for character segmentation from online Arabic cursive writings. The method is based on statistic and transformation of character movement vector. The segmentation depends on flattening the word and extract movement having uncommon x coordinate that represent possible ligature positions. Then, a combination of two filters is applied to refine the start and last segment of the word. The system were tested with varies Arabic texts to examine the performance of the system. The system scored 98.03%, 94.82, 97.33 and 90% for normal segmentation, starting segment, last segment and overlapped segment success rate. The errors presented in the test are common problem for Arabic as high overlap and rogza problem always exists.

The proposal for future work is to apply a recognition system for testing the system performance under recognition. In addition, it is required to solve high overlap and Arabic rogza segmentation errors.

V. ACKNOWLEDGEMENT

We would like to thank Ministry of Higher Education/Malaysia grant number 203/PELELT/6071189 for their assistant and contribution in finishing this work.

REFERENCE

[1] T. J. Klassen and M. I. Heywood, "Towards the on-line recognition of Arabic characters," in *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*, 2002, pp. 1900-1905.

[2] A. M. Namboodiri and A. K. Jain, "Online script recognition," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, 2002, pp. 736-739 vol.3.

[3] Z. Osman, L. Hamandi, R. Zantout, and F. N. Sibai, "Automatic processing of Arabic text," in *Innovations in Information Technology, 2009. IIT '09. International Conference on*, 2009, pp. 140-144.

[4] T. S. El-Sheikh and S. G. El-Taweel, "Real-time Arabic handwritten character recognition," in *Image Processing and its Applications, 1989., Third International Conference on*, 1989, pp. 212-216.

[5] S. Al-Emami and M. Usher, "On-line recognition of handwritten Arabic characters," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, pp. 704-710, 1990.

[6] A. M. Alimi, "A neuro-fuzzy approach to recognize Arabic handwritten characters," in *Neural Networks,1997., International Conference on*, 1997, pp. 1397-1400 vol.3.

[7] G. Al-Habian and K. Assaleh, "Online Arabic handwriting recognition using continuous Gaussian mixture HMMS," in *Intelligent and Advanced Systems, 2007. ICIAS 2007. International Conference on*, 2007, pp. 1183-1186.

[8] Randa I. Elanwar, Mohsen A. Rashwan, and Samia A. Mashali, Simultaneous Segmentation and Recognition of Arabic Characters in an Unconstrained On-Line Cursive Handwritten Document, World Academy of Science, Engineering and Technology 29, 2007, pp. 288-291.

[9] M. Hussain and M. N. Khan, "Online Urdu Ligature Recognition using Spatial Temporal Neural Processing," in *9th International Multitopic Conference, IEEE INMIC 2005*, 2005, pp. 1-5.

[10] R. Saabni and J. El-Sana, "Hierarchical On-line Arabic Handwriting Recognition," in Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on, 2009, pp. 867-871.

[11] K. Daifallah, N. Zarka, and H. Jamous, "Recognition-Based Segmentation Algorithm for On-Line Arabic Handwriting," in *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, 2009, pp. 886-890.

[12] M. Kherallah, L. Hadad, and A. M. Alimi, "A new Approach for Online Arabic Handwriting Recognition," in *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, 2009, pp. 22-25.

[13] B. Alsallakh and H. Safadi, "AraPen: An Arabic Online Handwriting Recognition System," in *Information and Communication Technologies, 2006. ICTTA '06. 2nd*, 2006, pp. 1844-1849.

[14] A. Elbaati, M. Kherallah, H. El Abed, A. Ennaji, and A. M. Alimi, "Arabic handwriting recognition using restored stroke chronology," in *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, 2009, pp. 411-415.

[15] Http://en.wikipedia.org/wiki/Arabic_alphabet

[16] M. Y. Potrus, U. Ngah, H. A. mat Sakem and S. A. Abdulrahman, "Normalization and Rectification Method for Online Hindi Digit Recognition with Partial Alignment Algorthm", Accepted in The 2010 International Conference on Electronics and Information Engineering (ICEIE 2010).

Proceedings of International Conference on
Engineering and Information Technology
Sep. 17-19, 2012, Toronto, Canada

A Harmony Search Algorithm for Recognition-Based Segmentation of Online Arabic Text

Moayad Yousif Potrus[1], Umi Kalthum Ngah[2*]

Imaging & Computational Intelligence Group
School of Electrical and Electronic Engineering
University Sains Malaysia, Malaysia
myp.ld09@student.usm.my[1], eeumi@eng.usm.my[2], umikalth@yahoo.co.uk[2]

Abstract

In this paper a Harmony Search algorithm (HS) is used for online Arabic text recognition. The algorithm is divided into two phases: text segmentation using dominant point detection and character recognition using HS. The segmentation algorithm uses dominant point detection to mark minimal number of points which could form the text skeleton. Then, the generated text skeleton is expressed as a directional model with 6 directions. This directional model minimizes the directions opposite to the writing direction. As a result, the new text directional expression will exploit all the possible segmentation points. Finally, HS is used to match the best database character to the target character generated from the segmentation process by minimizing the total score obtained from the overall text matching. The system is tested using a database of 4500 words forming 21234 characters in different positions or forms (isolated, start, middle and end). The data set is divided into a set of 3000 words for training and 1500 words for testing. The algorithm scored a 93.4% successful word recognition rate with an execution time of 4.3 sec.

Keywords: Character Segmentation, Dominant Point Detection, Handwritten Arabic Text Recognition, Harmony Search.

1. Introduction

Arabic text recognition is one example of complex character recognition problem. It has gained wide research interest in the past decade. Many competitions were being held to assess the success rate of various recognition systems and their performances in this area [19, 20]. The research motivation in Arabic character recognition originates from its special and complicated style of different writing forms. Complications in Arabic recognition comes from the possibility of writing a single word with one stroke or multiple strokes, many writing styles of a single character and generating different character when diacritics are added (i.e. dots, hamza,...etc).

The problem of Arabic text segmentation has led onto two different strategies of text recognition: segmentation-free and segmentation based strategy. In the segmentation free strategy, no segmentation operation is present and thus the whole text is recognized as one piece. Most of these systems depend on Hidden Markov Models (HMM) classifier. For example, HMM was combined with soft computing methods such as neural network [20] or fuzzy logic [22] or with other methods for segmentation-free Arabic text recognition. In most of these systems, the segmentation-free can generate numerous errors associated with the high similarities between the Arabic characters.

The segmentation based systems were initially used for Arabic text recognition in the 1980's and 1990's [1, 2]. The problem in the segmentation-based strategy comes from the problem of over segmentation or miss segmentation of characters. This problem led to two different approaches to tackle this problem: segmentation-based recognition and recognition-based segmentation. In the first strategy, the segmentation process extracts all the possible segmentation points and filters them according to the character skeleton knowledge for final recognition process [3]. Meanwhile, the second strategy tries to find the best segmentation point combination using a feedback process between segmentation points and recognition [4]. Although, it is slower than the segmentation-based recognition, the recognition-based segmentation strategy generates more accurate results for the recognition system. However, there are limited number of researches incorporating this method with regards to Arabic text recognition [5,7,9]. This work focuses on the use of this strategy for online Arabic text recognition.

Evolutionary computation algorithms provide an efficient calculation mechanism for hard NP problems [8].

* Corresponding author. Tel.: 0060134222600; Email: eeumi@eng.usm.my; umikalth@yahoo.co.uk

Although, these algorithms provide near optimum solutions, a longer time is spent to find the solution. This makes it inefficient for real time applications. As a result, Genetic Algorithm (GA) is initially used for offline character recognition systems in optimizing the feature selection problem. The generated population is used to find the smallest subset of features from a wider feature range which optimizes the separability between the different classes. This method was tested successfully in different languages such as Persian [24] Latin [6] and Chinese [23]. In addition to feature selection, it was also integrated for offline character recognition [15]. For online Arabic text, GA is used to find the best individual character from the matches based on the process of minimization of the bit comparison score. This score is found between two statistical based directional chromosomes: the written character and the full character dataset [16]. In addition to GA, other heuristic methods are also used in character recognition such as the swarm optimization. Particle swarm optimization (PSO) is combined with back propagation neural network and bee colony algorithm for digit recognition based on momentum features [21].

HS algorithm has proven to be a fast and efficient algorithm when compared to genetic algorithm [12]. It is used successfully to solve various optimization problems [10, 11]. However, it has never been tested in the field of text and character recognition. The way HS builds its new improvisations may involve all the entities in harmony memory. This may trigger ventures for new solutions or may lead to the optimum final solution. This feature elevates it as a useful tool in matching algorithms. In this paper, HS algorithm is used for online handwritten Arabic text recognition. The proposed segmentation algorithm generates a number of possible segmentation points using dominant point detection and vertical projection on the modified directional vector. Then, HS algorithm is applied to select the best match to the target character compared to those stored in a database. The recognition algorithm tries to determine the best set of character combined directional feature from the data set which may form a minimal matching score. Finally, it determines the best segmentation combination which generates the total text minimal score.

2. Arabic Character System

Arabic language and characters is used by more than 600 million peoples and its usage is especially extensive in large parts of Asia. Its character set are basically used for other language's character sets such as Persian, Urdu and Jawi. It consists of 28 letters in its basic isolated form. The normal form of writing is cursive directed from right to left. The cursive form changes the shape of the character when located at specific positions and this can be categorized into four groups: isolated, initial, medial or final form as shown in Fig.1. The number of characters is not fixed in each of the forms and does not include all the characters. It is specified as 28 in the isolated form, 23 in the start form, 23 in the middle form and 28 in the end form. Moreover, the main structure of some of these letters are the same (e.g. (ب,ت,ث) or (ى,ي,ئ)), but adding the punctuation (dot or hamza ء) will make these letter differ from one another.

Most of the common errors obtained from Arabic character recognition come from the close similarity between the character's common features. In addition, the cursive nature of the Arabic writings lead to a plethora of letter shapes and writing styles which are dependent upon each individual writer's style. Moreover, the standard writing movement for any letter could be easily changed, subject to the person's educational background or preferences. Other factors, which may affect the quality or clarity of writings, are the instability of the writing devices (e.g. writing pen and tablet device) which may cause inconsistency in the writing patterns, discontinuous connected patterns and character hooks which might occur at the start and end of the writing process. Fig. 2 illustrates some of the common characteristics and difficulties which may be implicated in the Arabic character system.

Isolated	Initial	Medial	Final	Isolated	Initial	Medial	Final	Isolated	Initial	Medial	Final
ا	ا	ـا	ـا	ر	ر	----	ـر	ف	فـ	ـفـ	ـف
ب	بـ	ـبـ	ـب	س	سـ	ـسـ	ـس	ك	كـ	ـكـ	ـك
ت	تـ	ـتـ	ـت	ش	شـ	ـشـ	ـش	ل	لـ	ـلـ	ـل
ث	ثـ	ـثـ	ـث	ص	صـ	ـصـ	ـص	م	مـ	ـمـ	ـم
ج	جـ	ـجـ	ـج	ض	ضـ	ـضـ	ـض	ن	نـ	ـنـ	ـن
ح	حـ	ـحـ	ـح	ط	طـ	ـطـ	ـط	ه ـه	هـ	ـهـ	ـه
خ	خـ	ـخـ	ـخ	ظ	ظـ	ـظـ	ـظ	و	و	----	ـو
د	د	----	ـد	ع	عـ	ـعـ	ـع	ى	ى	ـى	ـى
ذ	ذ	----	ـذ	غ	غـ	ـغـ	ـغ				
ر	ر	----	ـر	ف	فـ	ـفـ	ـف				

Figure 1: Arabic letters data set with the different letter forms.

3. Arabic Text Recognition System

The proposed recognition system considers that the characters are already segmented and ready to be recognized. It consists of three steps: preprocessing, classification and recognition. Fig. 3 shows the description of the system used in online Arabic character recognition.

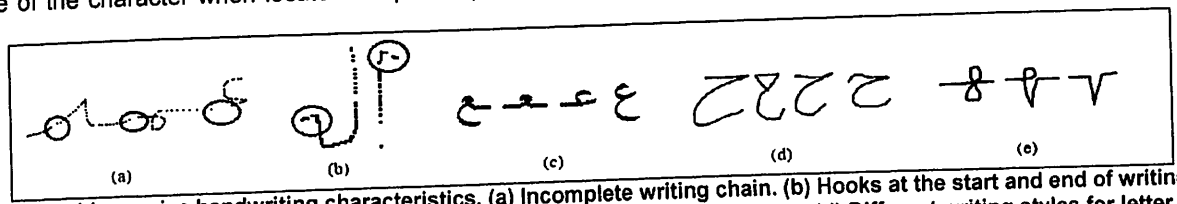


Figure 2. Arabic cursive handwriting characteristics. (a) Incomplete writing chain. (b) Hooks at the start and end of writings. (c) Changes in the letter ع shape at different positions (isolated, start, middle and end). (d) Different writing styles for letter ح. (e) Different movement patterns for Arabic letter(ـ) at the middle position.

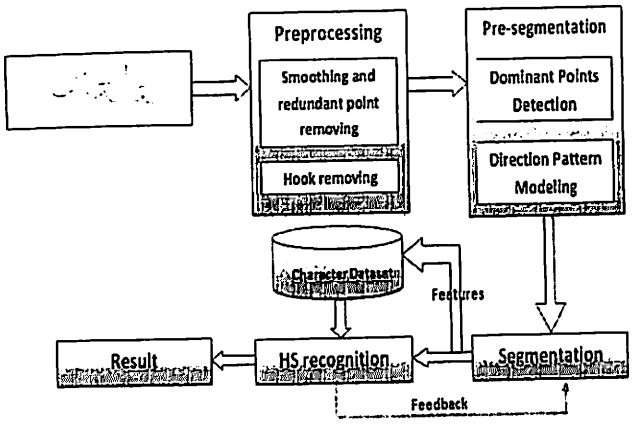


Figure 3. The proposed system for online Arabic Text recognition.

3.1. Preprocessing

In the preprocessing step, the characters are enhanced and smoothened to remove writing errors and jitters, in addition to the written character problems described in the previous section and shown in Fig.2 (a and b). In this study, Bezier cubic curve approximation algorithm [13] is used to fill the gaps between the characters points, as well as smoothing the character shape. The algorithm splits the character into groups of four points. Then the curve approximation is applied for each group using the following equation,

$$BZ(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t)P_3 + t^3 P_4 \qquad (1)$$

Where P_x represents the character points, and t is a control parameter varying between [0, 1]. The result of this approximation is that, all the gaps will be filled with substitution points. However, redundant points will be produced along the curve. The application of the algorithm thereafter will remove all the redundant points to produce a consistent set of character features. In addition, a hook removing algorithm is used to remove hooks from the start and end of the characters by detecting the existence of sudden changes in the angle direction [14].

3.2. Pre-Segmentation Algorithm

The pre-segmentation algorithm for Arabic text recognition consists of two main steps: dominant point detection and directional modeling. The following two sections explain each of these steps.

3.2.1. Dominant Point Detection

Dominant point detection is used to identify a set of point which defines the whole text patterns. These points are located in places where the writing direction is changed. The proposed original work by Marji *et. al.*[19] which is used to identify dominant points within polygons for images, is modified in this work to suit the case of online written text. The following are the outline of the modified algorithm.

Let $C = \{p_i = (x_i, y_i), i = 1,\ldots,n\}$ be the coordinate points which describe the handwriting. Let p_j be the point where the region of support to be determined and F is the function to be minimized with F_{old} is the old function value. Then,

Algorithm Dominant_Point_Detection

1 Input N (number of text points), P_N =coordinate points set of the text
2 Output P_D set of dominant points, M number of dominant point
3 Set F_{old}=0, M=0.
4 For j=1 To N
5 set $p_j=(x_j, y_j)\ \epsilon P_N$;
6 Initialize k=j+2 and set $p_k=(x_k, y_k)\ \epsilon P_N$;
7 Calculate $L_{jk}=sqrt((x_k-x_j)^2+(y_k-y_j)^2)$
8 For g=j+1 To k-1
9 Find ||n||= perpendicular distance of point p_g between p_j and p_k
10 Calculate $E_g = E_g + |Ax_g + By_g + C| / ||n||$
11 End For
12 Calculate $F_{new}=L_{jk} - E_{ik}$.
13 IF $F_{new}<F_{old}$, then set $p_{k-1}\ \epsilon P_D^*$ and M=M+1;
14 Else
15 set $F_{old} = F_{new}$.
16 Increment k and go to step 2.
17 End else
18 End For
19 For j=2 to M*
20 Find the directional DR_j of line p_jp_{j-1} using Fig. 4(a).
21 End For
22 For j=2 to M*-1
23 If $DR_j = DR_{j-1}$ Then
24 Push p_{j-1} and p_{j+1} to P_D
25 set M=M*-1;
26 End If
27 End For
28 End Procedure

This algorithm will transform the original hand writing into a set of straight lines as shown in Fig. 4(b). It is clear from the figure that the algorithm has reduced the amount of points which defines the whole text. This approach will make it easier for feature extraction process. Moreover, the final text structure will neither contain any curves nor does it smoothen the polygon shapes. This process will make it easier to locate candidate segmentation points after applying the next step.

3.2.2. Directional Pattern Modeling

The recognition process is highly dependent on locating a possible segmentation point (candidates). Many previous studies have tried to obtain an increase in the chance of finding these point by either using vertical-horizontal projection or baseline detection [2,3]. However, for the cases shown in Fig 5 (a), these methods may greatly suffer from missing segmentation points. In order to discover all of the possible segmentation points, the model shown in Fig 4 (a) is used to describe the handwritten text output obtained from the dominant point detection. Then, the resultant direction model is transformed into 6 directions by omitting the direction vectors 2 and 4. This process transforms the directions which are opposite to the writing direction (2, 3 and 4) into a back movement (3) and an adjacent direction (in most

cases into 1 or 5) according to the direction of the adjacent line segments.

The output of this process will expose all hidden segmentation points (with respect to vertical projection) by increasing the segment lines between the characters, while decreasing the effective area of the character body. As a result, the vertical projection can easily locate all candidate segmentation points even in the cases of Fig 5(a) as shown in Fig 5 (b).

3.3. Segmentation-Based Algorithm

3.3.1. Harmony Search Recognizer

Harmony search algorithm uses the same concept as in all evolutionary algorithms i.e. which is based on random population generation. It consists of a harmony memory having a size (HMS) which contains the generated population and their corresponding solution [13]. This memory can be described in (2) as,

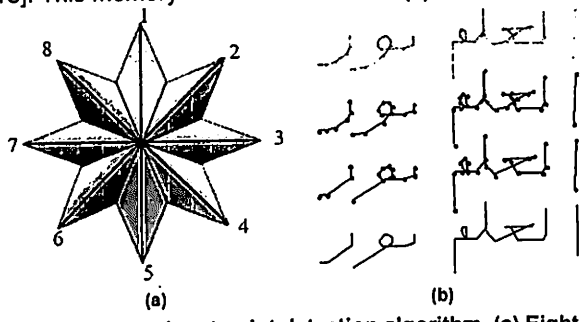


Figure 4. Dominant point detection algorithm. (a) Eight directional model for line slope merging. (b) Arabic text which result from the application of the algorithm.

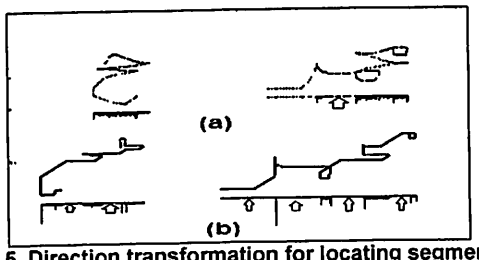


Figure 5. Direction transformation for locating segmentation point. (a) Hard segmentation point discovery. (b) Direction transformation effect on exploring all candidate segmentation points.

$$HM = \begin{bmatrix} x_1^1 & \cdots & x_n^1 & f(x^1) \\ \vdots & \ddots & \vdots & \vdots \\ x_1^{HMS} & \cdots & x_n^{HMS} & f(x^{HMS}) \end{bmatrix} \quad (2)$$

Where n is the number of variables used to find the function f. The solution is constructed by selecting random values from HMS or from a vector bounded by values x_{min} and x_{max} depending on the value generated from a random variable called harmony memory consideration rate ($0 \le HMCR \le 1$). In this work, the minimum and maximum limits are the directional vectors 1 and 8. Thus,

$$x_i^{new} = \begin{cases} \in HM & probability = HMCR \\ \in [1,8] & probability = HMCR \end{cases} \quad (3)$$

When harmony memory value is selected, another parameter is used to decide whether this value is picked or to be tweaked to another value called the pitch adjustment rate (PAR). In discrete harmony search, the PAR adopts the neighbor value according to the shift parameter.

$$x_i^{new}(k) = \begin{cases} x_i^{new}(k) & probability = 1 - PAR \\ x_i^{new}(k+m) & probability = PAR \end{cases} \quad (4)$$

where m ϵ [-1,1] is the shifting value. To consider the shift direction, a shifting parameter (SP) is used and defined by:

$$x_i^{new}(k) = \begin{cases} x_i^{new}(k-1) & propability = SP \\ x_i^{new}(k+1) & propability = 1 - SP \end{cases} \quad (5)$$

The most important part in metaheuristic optimization problems is to define the objective function. To define such functions, let us consider the characters ع, ح and ه which has similar structures in the end part for (ح،ع) and similar starting structures for (ه،ع). The matching function can be defined based on the movement structure by partitioning all the characters into a similar number of parts N. This representation can be separated into 3 parts: start, middle and end movement parts which can be compared separately. For each part, a weight is given to separate the matching according to the character forms, Thus, the matching function can be written as,

$$f_{match} = w_s C_s + w_m C_m + w_e C_e \quad (6)$$

with,

$w_s = w_e > w_m$ *Isolated*

$w_s > w_m, w_e = 0$ *Initial*

$w_m > w_s > w_e$ *Medial*

$w_e > w_s > w_m$ *Final*

where w is the weight value assigned to the character part. C is the matching function defined as,

$$C = \sum_{i=1}^{N} V_i \rightarrow V_i = \begin{cases} -1 & v_t = v_s \\ 1 & otherwise \end{cases}$$

where, v_t *and* v_s are the target and source directions. From (5), it can be noted that the weight value is considered differently for each form. For the initial form, only the start and middle part is considered in the matching process because the end part represents the connector between the characters and is similar in all of them. For the medial and final form, the start and middle part is mostly considered since most characters are similar at the end part. The matching function cannot be considered as the only scoring function because in some cases, the matching process fails to select the right character. To enhance the scoring function, parameters p_i^d which represent the probability of a direction d at position i for each character in the training dataset are used. They are used to reduce the possibility of incorrect directions or misplaced directions to occur in the new improvised harmony vector. For the chosen direction model, each character in the training dataset has N by 8 probability value array to reflect the direction probability per position as,

$$\begin{bmatrix} p_1^1 & \cdots & p_N^1 \\ \vdots & \ddots & \vdots \\ p_1^8 & \cdots & p_N^8 \end{bmatrix}$$

The total direction vector probability is found to be,

$$P = \sum_{i=1}^{N} p_i^d \quad (7)$$

Thus, the final matching objective function to be minimized is defined as,

$$Min \quad F_{score} = f_{match} \times \frac{1}{P} \tag{8}$$

The parameter 1/P minimizes the scoring function whenever the improvised direction is correctly located in the right position between the source and target direction vectors. At the same time, the matching function is minimized when the direction vectors of the dataset and the target are closely matched according to (5).

3.3.2. Character Segmentation and Extraction

In order to find the best segment combination, a different approach to the one used in [7] is used. The approach proposed in [7] may take a long time to find the right combination when the number of segmentation points exceeds 5 (5!=120 combinations). Instead, an approach based on score and shift is considered. This approach tries to minimize the following function,

$$F_{Total} = Min \quad F_{score} = \frac{\sum_{i=1}^{C(n)} F_{char}}{C(n)} \tag{9}$$

where F_{char} is the character matching score obtained from HS matching. $C(n)$ is the number of characters obtained from n segmentation points. The algorithm works as follows:

1. Consider the text as isolated and calculate the score from (8) and consider it as minimum.
2. Take only one segmentation point starting from right to left and calculate the minimum of (9) with C(1)=2. If at any point, the minimum generated by any point is less than that obtained from step 1, then omit the possibility of an isolated character and proceed to 3. Else display the matched isolated character as the final result.
3. Find the minimum of each segmentation point from 2 using (8) and set it as the first segmentation point.
4. Take the next points to the left of the one found in (3) and find the minimum from these points and set as the second segmentation.
5. proceed with 4 until there are no more candidate segmentation points.

This procedure will try to affirm the selection for each of the candidate segmentation points. The algorithm will try to locate the minimum HS score for each of the generated characters from the point shifting starting from right to left until the score of (9) is the minimum.

Fig. 6 shows the working of this algorithm. It can be noted that each time the window frames the correct characters, the scores of (8) and (9) are minimized. This algorithm will cost (n^2) at most to locate the right combination compared to (n!) used in [7].

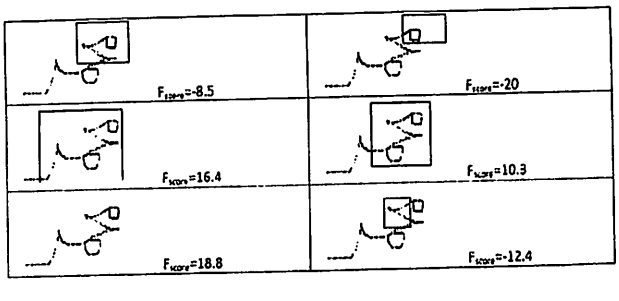


Figure 6. Different matching scores for different character segmentation point consideration.

4. Results and Discussion

In order to test the system, a data set of 4500 Arabic collected words during the study is used. The total numbers of characters in these words are 24960 characters. The data set is divided into two parts: 14500 characters are obtained from 3000 words which are used in the HS matching process, while 1500 words with 14460 characters are used for testing the HS recognizer. The stored 14500 characters are stored as 250 samples per character with 15,10,10,15 characters per form; some of these forms have more than one class based on their similarities. Table 1 shows the results of the recognition success rates in addition to the errors inflicted at different stages. The final recognition rate scored an overall accuracy of 93.6%. It can be noted that the highest number of recognition errors occurred during the segmentation process whereby some of the words and character features are not possible to be segmented. Moreover, some recognition errors are obtained during the test since many of the written samples may interfere with other groups due to the similarity in the writing styles.

Table 2 shows the comparison of the test result with previous recognition-based segmentation systems. The system over-scores when compared with the previous systems because of the mechanism used in the character matching. However, it must be noted that this system depends on the probability of score as well as the one-to-one match. This feature is proposed to support the idea of writer independency. The system fails to identify some of the 100% one-to-one matched characters because the probability score is low. This low probability is obtained because the writing style of the character is different than the normal writing style as shown in some of the errors obtained from the test in Fig. 7. The style of character (عـ) is closely similar to character (قـ) and therefore, the system recognized it as (قـ) rather than (عـ).

Table 1: The system recognition results for the test data.

Sample	No of Samples	Correctly recognized	Seg. Errors	Rec. Error	Accurcy
Training	3000	2825	132	43	94.17%
Testing	1500	1396	50	54	93.03%
Overall Accuracy					93.6%

Table 2: Comparison with other recognition-based segmentation systems.

System	Method	Database Size	Accurcy
Razzak *et. al.* [7]	HMM	150	83.03%
Touj et. al. [25]	Hough Trans. and	Words with 6400 characters.	91%
HS with DPT.	HS	4500	93.6%

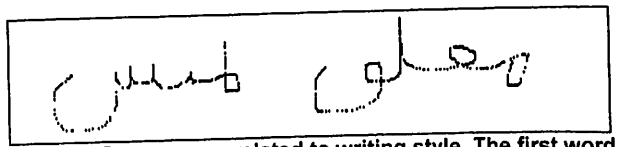


Figure 7. Some errors related to writing style. The first word is (معلق) recognized as (مقلق), and the second is (ليس) recognized as(ليس).

5. Conclusion

In this study, a recognition-based segmentation is proposed for online handwritten Arabic text. The system utilizes the dominant point detection algorithm with HS to find the best combination of segmentation points for the best recognition result. The system incorporates and modifies the dominant point detection to extract all possible segmentation points. Then, it locates the best segmentation points by using a feedback system between HS and total word score. The system is tested using 4500 words which scored a success rate of 93.6%.

In conclusion, the use of dominant point detection and HS in handwriting is promising. The obtained results can be further modified and enhanced by incorporating the system with more features. This can increase the separability between the characters as well as increase the efficiency of the segmentation process. Then, solutions to the problems related to character overlapping and merging can be facilitated.

AKNOWLEDGEMENT

We would like to thank the Ministry of Higher Education/Malaysia grant number 203/PELELT/6071189 and Universiti Sains Malaysia for their assistance and contribution to finish this work.

REFERENCES

[1] Al-Emami, S. & Usher, M., On-line recognition of handwritten Arabic characters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12, 704-710, 1990.

[2] Amin, A.: "OCR of Arabic Texts". Pattern Recognition, (616-625), 1988.

[3] Amin, A.; Al-Fedaghi, S.; " Machine recognition of printed Arabic test utilizing natural language morphology", International journal of Man-Machine studies, 35(6): 769-788, 1991.

[4] Casey, R. G., Lecolinet, E., A survey of methods and strategies in character segmentation, IEEE Trans. Pattern Analysis and Machine Intelligence, 18,7, pp. 690-706, 1996.

[5] Cheung, A., M. Bennamoun, et al. . "An Arabic optical character recognition system using recognition-based segmentation." Pattern Recognition 34(2): 215-233, 2001.

[6] Cordella, L.; De Stefano, C.; Fontanella, F.; Marrocco, C.; , "A feature selection algorithm for handwritten character recognition," *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* , (1-4), 2008.

[7] Daifallah, K.; Zarka, N.; Jamous, H., "Recognition-Based Segmentation Algorithm for On-Line Arabic Handwriting," in Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on, (886-890), 2009.

[8] Eiben, A.E.; Smith, J.E., Introduction to Evolutionary Computing, Springer, Natural Computing Series,1st edition, 2003.

[9] Elnagar, A. and R. Bentrcia. "Recognition-based Segmentation of Arabic Handwriting." Pattern Recognition in Information Systems, Proceedings. (83-92), 2009.

[10] Frosolini, M., M. Braglia, et al.; "A modified harmony search algorithm for the multi-objective flowshop scheduling problem with due dates." International Journal of Production Research 49(20): 5957-5985, 2011.

[11] Gao, K. Z., Q. K. Pan, et al.; "Discrete harmony search algorithm for the no-wait flow shop scheduling problem with total flow time criterion." International Journal of Advanced Manufacturing Technology 56(5-8): 683-692, 2011.

[12] Geem, Z. W., "Music-Inspired Harmony Search Algorithm, " Springer, 2009.

[13] Hain, T.F.; Racherla, S.V.R.; Langan, D.D.; , "Fast, precise flattening of cubic Bezier segment offset curves," *Computer Graphics and Image Processing, 2004. Proceedings. 17th Brazilian Symposium on* , vol., no.. pp. 244- 249, 2004.

[14] Huang, B.; Zhang, Y. B.; Kechadi,M., Preprocessing Techniques for Online Handwriting Recognition, Intelligent Text Categorization and Clustering, pp. 25-45, 2009.

[15] Kala, R.; Vazirani, H.; Shukla, A. ; Tiwari, R., Offline Handwriting Recognition using Genetic Algorithm, International Journal of Computer Science Issues, 7(2), No 1, pp. 16-25, 2010.

[16] Kherallah, S., Bouri, F.; Alimi, A.M. , "On-line Arabic handwriting recognition system based on visual encoding and genetic algorithm, " Engineering Applications of Artificial Intelligence. 22 (153-170), 2009.

[17] Margner, V.; El Abed, H.; , "ICDAR 2009 Arabic Handwriting Recognition Competition," Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on , (1383-1387), 2009.

[18] Margner, V.; Pechwitz, M.; Abed, H.E.; , "ICDAR 2005 Arabic handwriting recognition competition," Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on , 1 (70- 74), 2005.

[19] Marji, M.; Siy, P. : " Polygonal representation of digital planar curves through dominant point detection—a nonparametric algorithm", Pattern Recognition 37, (2113 – 2130), 2004.

[20] Narima, Z.; Messaoud, R.; Mouldi, B.; , "Neuro-Markovian hybrid system for handwritten Arabic word recognition," Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003 10th IEEE International Conference on , 2 (878- 881), 2003.

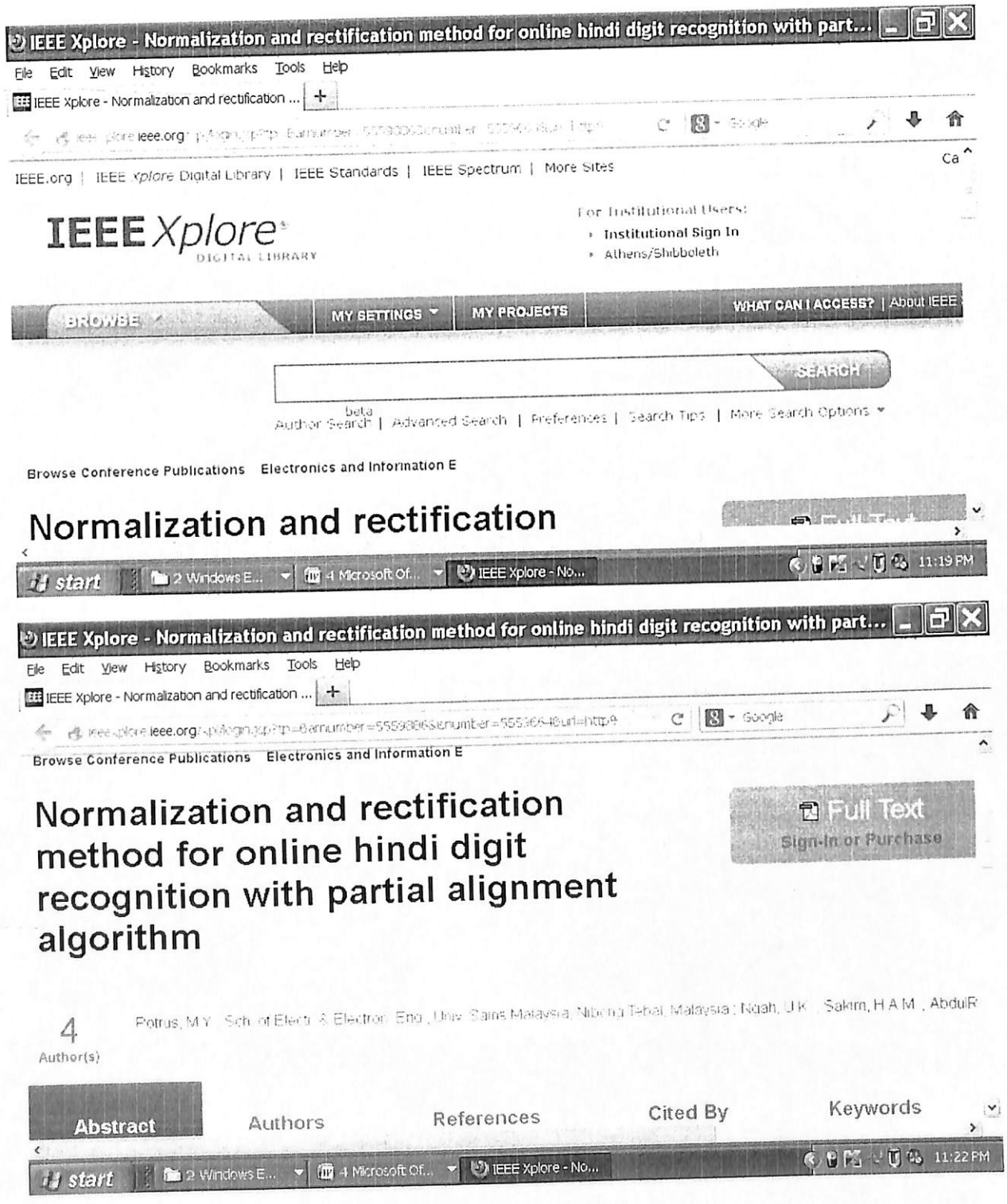
[21] Nebti, S., Boukerram, A., Zavoral, F., Yaghob, J., Pichappan, P. & El-Qawasmeh, E. , Handwritten digits recognition based on swarm optimization methods,Networked Digital Technologies, Springer, Communications in Computer and Information Science, 87, Part 1, (45-54), 2010.

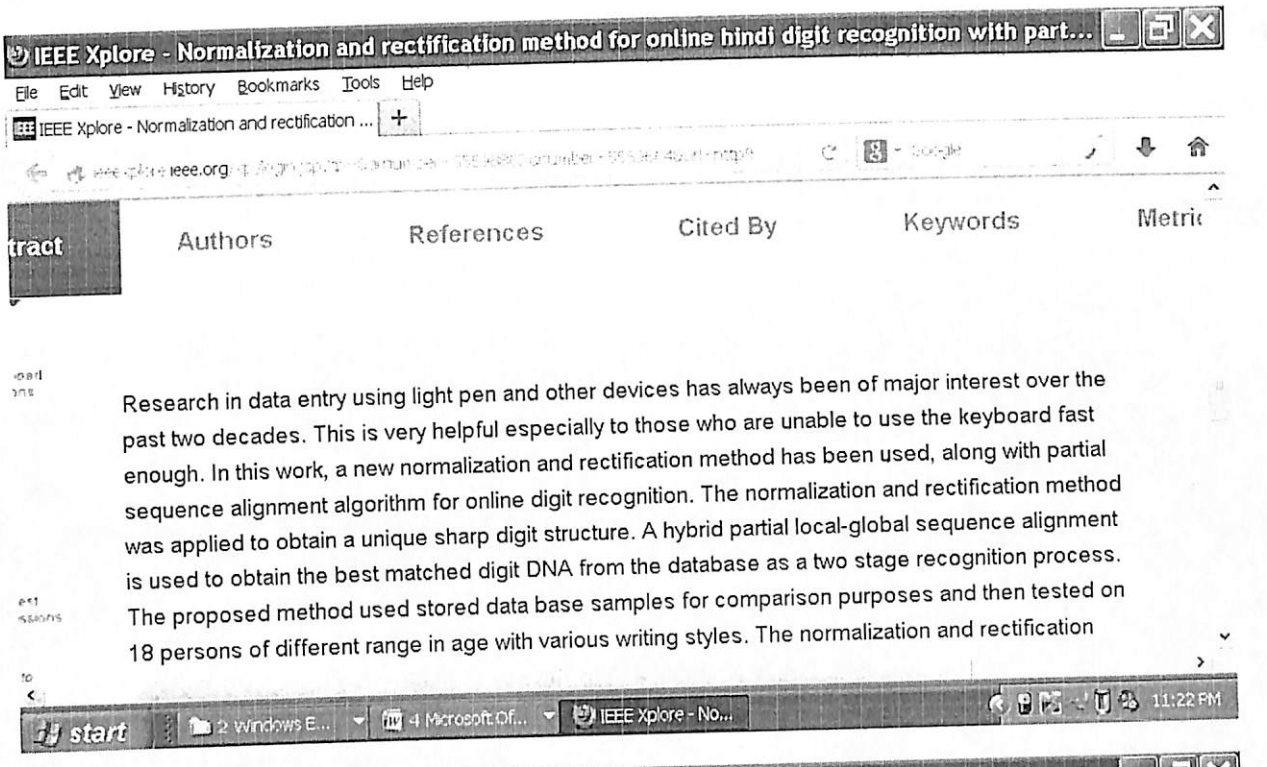
[22] Razzak, M. I.; Anwar, F.; Husain, S.A. ; Belaid, A.; Sher, M., "HMM and fuzzy logic: A hybrid approach for online Urdu script-based languages' character recognition," Knowledge-Based Systems. 23, Issue 8, (914-923), 2010.

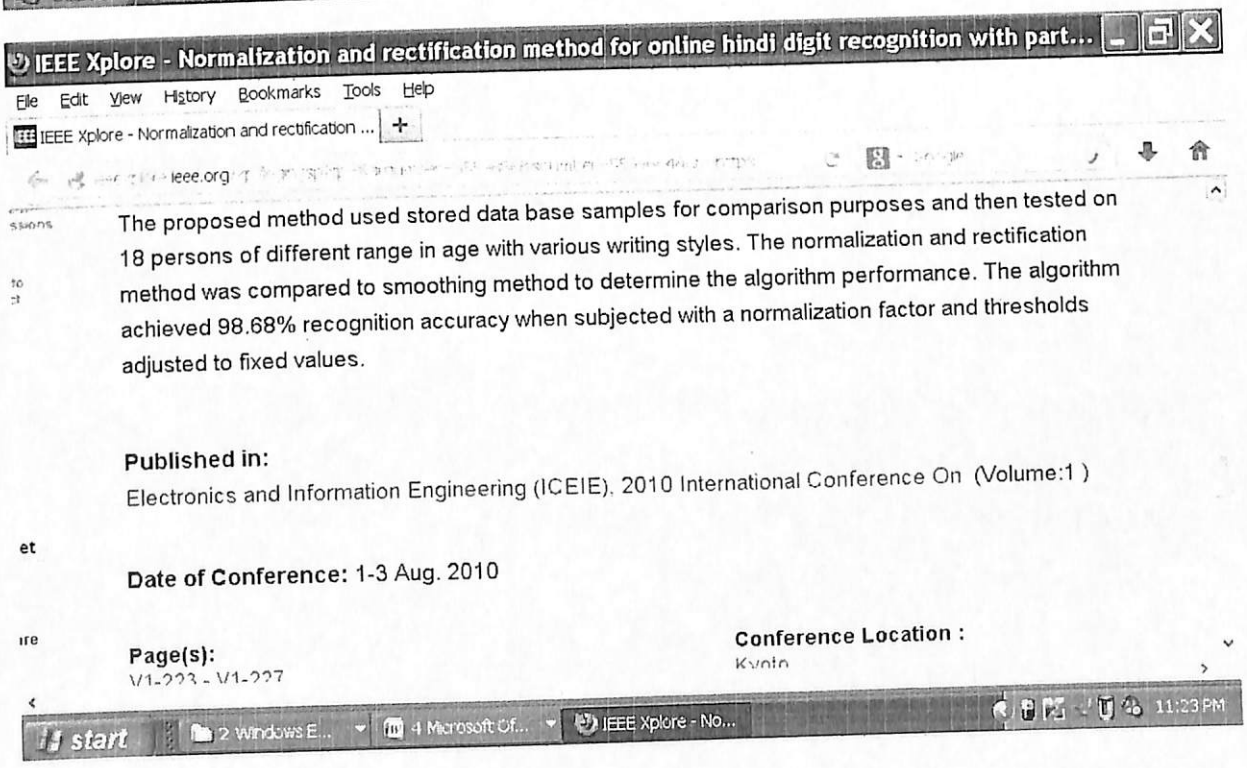
[23] Shi, D.; Shu, W.; Liu, H., "Feature selection for handwritten Chinese character recognition based on genetic algorithms," *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on.* 5 (4201-4206), 1998.

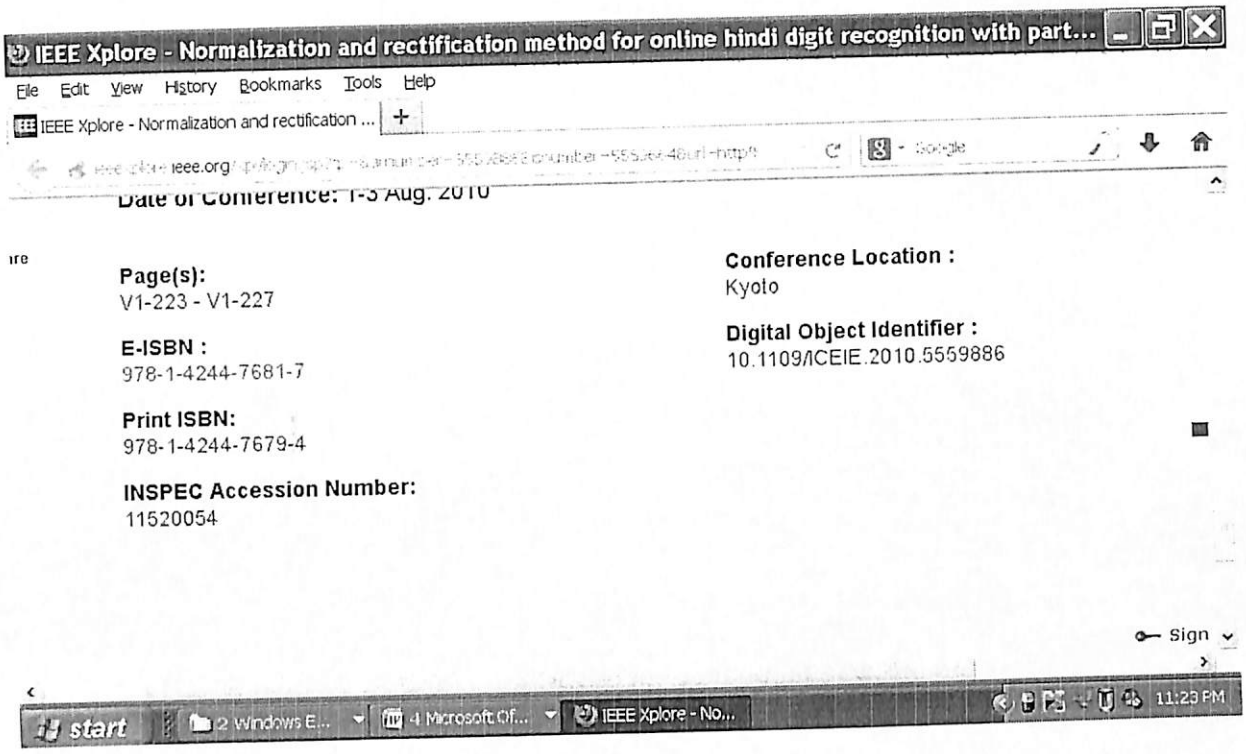
[24] Soryani, M. ; Rafat, N., Application of Genetic Algorithms to Feature Subset Selection in a Farsi OCR, World Academy of Science, Engineering and Technology ,18, (113-116), 2006.

[25] Touj, S.; Ben Amara, N.; Amiri, H.; " Two approaches for Arabic Script recognition-based segmentation using the Hough transform", Proceeding of the ninth internation conference on document analysis and recognition, vol. 2, pp. 654-658, 2007.


IEEE Xplore - Normalization and rectification method for online hindi digit recognition with part...
File Edit View History Bookmarks Tools Help
IEEE Xplore - Normalization and rectification ...
IEEE.org | IEEE Xplore Digital Library | IEEE Standards | IEEE Spectrum | More Sites
IEEE Xplore DIGITAL LIBRARY
For Institutional Users:
Institutional Sign In
Athens/Shibboleth
BROWSE
MY SETTINGS
MY PROJECTS
WHAT CAN I ACCESS?
About IEEE
SEARCH
Author Search | Advanced Search | Preferences | Search Tips | More Search Options
Browse Conference Publications Electronics and Information E
Normalization and rectification
start
2 Windows E...
4 Microsoft Of...
IEEE Xplore - No...
11:19 PM
IEEE Xplore - Normalization and rectification method for online hindi digit recognition with part...
File Edit View History Bookmarks Tools Help
IEEE Xplore - Normalization and rectification ...
Google
Browse Conference Publications Electronics and Information E
Normalization and rectification method for online hindi digit recognition with partial alignment algorithm
Full Text
Sign-In or Purchase
4
Author(s)
Abstract
Authors
References
Cited By
Keywords
start
2 Windows E...
4 Microsoft Of...
IEEE Xplore - No...
11:22 PM


IEEE Xplore - Normalization and rectification method for online hindi digit recognition with part...
File Edit View History Bookmarks Tools Help
IEEE Xplore - Normalization and rectification ...
ieee.org
tract
Authors
References
Cited By
Keywords
Metri
Research in data entry using light pen and other devices has always been of major interest over the past two decades. This is very helpful especially to those who are unable to use the keyboard fast enough. In this work, a new normalization and rectification method has been used, along with partial sequence alignment algorithm for online digit recognition. The normalization and rectification method was applied to obtain a unique sharp digit structure. A hybrid partial local-global sequence alignment is used to obtain the best matched digit DNA from the database as a two stage recognition process. The proposed method used stored data base samples for comparison purposes and then tested on 18 persons of different range in age with various writing styles. The normalization and rectification
start
2 Windows E...
4 Microsoft Of...
IEEE Xplore - No...
11:22 PM


IEEE Xplore - Normalization and rectification method for online hindi digit recognition with part...
File Edit View History Bookmarks Tools Help
IEEE Xplore - Normalization and rectification ...
ieee.org
The proposed method used stored data base samples for comparison purposes and then tested on 18 persons of different range in age with various writing styles. The normalization and rectification method was compared to smoothing method to determine the algorithm performance. The algorithm achieved 98.68% recognition accuracy when subjected with a normalization factor and thresholds adjusted to fixed values.
Published in:
Electronics and Information Engineering (ICEIE), 2010 International Conference On (Volume:1)
Date of Conference: 1-3 Aug. 2010
Page(s):
V1-223 - V1-227
Conference Location :
Kyoto
start
2 Windows E...
4 Microsoft Of...
IEEE Xplore - No...
11:23 PM


IEEE Xplore - Normalization and rectification method for online hindi digit recognition with part...
File Edit View History Bookmarks Tools Help
IEEE Xplore - Normalization and rectification ...
ieee.org
Google
Date of Conference: 1-3 Aug. 2010
ire
Page(s):
V1-223 - V1-227
E-ISBN :
978-1-4244-7681-7
Print ISBN:
978-1-4244-7679-4
INSPEC Accession Number:
11520054
Conference Location :
Kyoto
Digital Object Identifier :
10.1109/ICEIE.2010.5559886
Sign
start
2 Windows E...
4 Microsoft Of...
IEEE Xplore - No...
11:23 PM

2010 International Conference on Computer Applications and Industrial Electronics (ICCAIE 2010), December 5-7, 2010, Kuala Lumpur, Malaysia

An Effective Segmentation Method for Single Stroke Online Cursive Arabic Words

Moayad Yousif Potrus[1], Umi Kalthum Ngah[2], Harsa Amylia Mat Sakim[3]
School of Electrical and Electronic Engineering, Universiti Sains Malaysia
Penang, Malaysia
myp.ld09@student.usm.my[1], eeumi@eng.usm.my[2], harsaamylia@ieee.org[3]

Abstract— **A new method is used for character segmentation from cursive Arabic words. The method is based on statistical approach which uses Normalization and rectification, coordinate transformation and clustering to extract ligatures. The output is then filtered to extract start, overlapped and end segment errors. After applying the filter the characters are completely isolated and ready for recognition. The system, when testing the segmentation on 5 different Arabic sentences and by 20 different users, scored 98.03%, 94.82, 97.33 and 90% for normal segment, starting segment, last segment and overlapped segment.**

Keywords—Character Segmentation, Single Stroke, Arabic Word Ligatures.

I. INTRODUCTION

Development in computer devices during the last decade, have led to demanding market for fast and efficient input devices. Computer devices like tablet PC, PDA and Mobile devices are widely used nowadays. These devices depend on touching or writing in order to input data or perform a task. Traditional writing represents the most comfortable style for a user to write documents or enter data, especially those with slow keyboard typing. Therefore, ongoing online recognition studies attempt to meet the users' demands of efficient and accurate recognizers to make it easier for the task of data entry.

The main challenge for online character researches is to find the right form for successfully recognizing the handwritten characters. The problem may be easy to handle for isolated characters [1]. However, for cursive handwritten words, difficulties may be encountered [2]. Cursive words require more processing. Thus, complex algorithms are used to try to separate the characters from ligatures between them [3]. Researchers have taken a great interest in the Arabic language style (similar to Urdu, Farsi and Jawi) for the last decade. The complex and versatile cursive writing style poses as a great challenge to overcome.

There are very limited researches dealing with online cursive Arabic character segmentation. It is very hard to obtain acceptable results due to the word structure complexity. Many researches introduced a method of writing Arabic words as a sequence of separated characters (multiple strokes) rather than a single stroke per word [4] [5] [6] [7]. The problem with this method is that, the user is obliged to use the writing method proposed by the developer instead of his or her own natural writing style. In addition, an error which occurs in one stroke may cause a whole word error. Another attempt that was used depends on partial segmentation of a multi stroke per word [8]. Again, the stroke addressing is subject to correct position and the system ligature position depends on the recognition system decision and verification. Other methods avoided the segmentation process and attempted to recognize the whole word [9] [10]. These systems are very sensitive to the amount of database template and their writing styles. In addition, covering all the words in the Arabic dictionary plus adding multiple templates per word would be timely expensive.

Few researches attempted to segment the characters from the word. Daifullah *et al.* [11] proposed a method of segmenting the characters based on detecting right to left movement that have angles smaller than θ=30o and then calculating all possible segment combination. This method predicted the possible segments location and depended on recognition for specifying the correct combination. This made it very sensitive to segment the location error when the character overlapped or a mis-segmentation occurs. Kherallah *et a.l* [12] proposed a method to separate characters from ligatures using stroke speed measurement. They suggested that characters can be separated from ligatures whenever a rapid stroke speed change occurs. This assumption can easily fail whenever the stroke speed was constant or the speed difference was not that significant. Alsallakh *et al.* [13] proposed a segmentation method based on horizontal movement detection and dynamic time wrapping for character movement comparison. The system is highly sensitive for curvature ligatures and overlapping between the characters. Abdulkarim *et al.* [14] suggested that, segmentation of pseudo-words in graphemes rests on the detection of two types of points which are typographically significant. These points are summits of the valleys bordering the base line with a parallel tangent and angular points. The word may not have a base line as reference for segmentation as the writing style may impose multi-reference lines due to imperfect writing styles.

In this paper, we propose an effective statistical segmentation technique based on normalization, direction transformation and clustering, to determine the segment location. Afterwards, filters are used to correct segmentation error occurs at the first, last and overlap character. The proposed segmentation system can deal with curved, jittered and noisy single stroke Arabic word. The central idea of our system is to flatten the words to represent a multi connected line. These lines are filtered to obtain the ligature position.

II. SYSTEM DESCRIPTION

The Arabic characters in a cursive writing are classified into four pattern based on their position. These

classifications are isolated form, start form, middle form or end form (as shown in figure 1) [15].

These characters can be classified into 18 main groups for isolated character form, 11for start form, 9 for middle form and 16 for end as shown in table 1. This classification is very important and helpful for recognition since these groups will control the flow of recognition, as well as, minimizing the recognition error.

The problem of cursive Arabic word can be represented as follows: Consider an Arabic word W having N of (x, y) coordinate points and M characters C. The number of segments S is always less by 1 than the number of characters C. Thus, we have $M-1$ segments. Then,

$$(x,y)_i \in S_j \Leftrightarrow (x,y)_i \notin C_k \forall (x,y)_i \in W$$

$$\text{With } i = 1,\ldots..,N,\ j = 1\ldots.M-1 \quad \text{and} \quad k = 1\ldots.M\ .$$

It is required to find which coordinate point belongs to the segment and which belong to the character.

Figure 2 below shows the full diagram for segmentation and recognition system. The components for our system are described in the following sections.

A. *Preprocessing*

The preprocessing phase includes the following two steps:

A.1. Smoothing

The Arabic written word is smoothed to modify and remove rough edges to obtain more enhanced writing with less noisy shape. The output will be more stable and recognizable as it gives a straight form of the character and ligature.

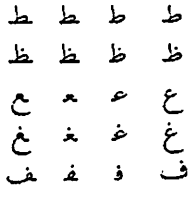


Figure 1. Some Arabic characters and their pattern based on position

TABLE I. GROUPS OF ARABIC CHARACTERS

Form	End Form(EF)	Middle Form(MF)	Start Form(SF)	Isolated Form(IF)
Characters	(ب ت ث ن ي) (ج ح خ) (د ذ) (ر ز) (س ش) (ص ض) (ع غ) (ف ق) (ك) (م) (ه) (و) (ا ل لا)	(بـ تـ ثـ نـ يـ) (جـ حـ خـ) (سـ شـ) (صـ ضـ) (عـ غـ) (فـ قـ) (لـ) (مـ) (هـ)	(بـ تـ ثـ نـ يـ) (جـ حـ خـ) (سـ شـ) (صـ ضـ) (طـ ظـ) (عـ غـ) (فـ قـ) (لـ) (مـ) (كـ) (هـ)	(ا ا ا) (ب ت ث ن) (ي ى) (ج ح خ) (د ذ) (ر ز) (س ش) (ص ض) (ط ظ) (ع غ) (ف ق) (ك) (ل) (م) (ه) (و) (لا لا لا لا)(ء)
No. of Groups	16	9	11	18

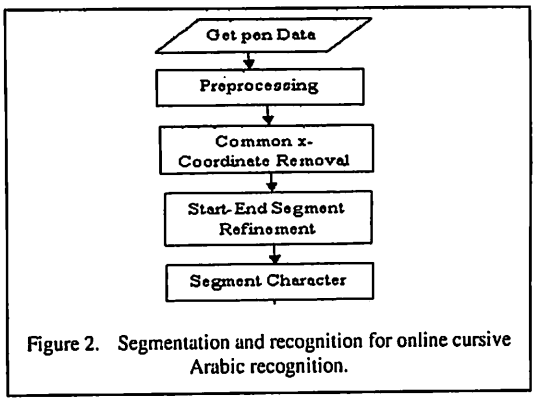


Figure 2. Segmentation and recognition for online cursive Arabic recognition.

A.2. Normalization and Rectification

The complete word will be normalized and rectified for removing errors and construct segment line based writing [16]. The following formula is used to normalize the complete word.

$$S = S - S_j \quad \forall P_j < \lambda = \frac{1}{10}\log\frac{N}{2} \tag{1}$$

Where S is the direction vector for the word, S_j is the direction vector in direction j, P_j is the probability of existence for direction j , λ is the normalization factor and N is the number of points for the word. Then, the output of this process is further rectified to remove errors.

$$S = S - S_{ij} \quad \forall S_{ij} < \varepsilon \tag{2}$$

Where S_{ij} is the partial vector i in the direction j, ε is the rectification factor (selected as 2 empirically). The output of these two steps represents a number of joint lines.

B. *Coordinate Transformation and Clustering*

The coordinate of the system is changed from (x, y) coordinate into four direction vector of Up, Down, right and Left. Then each movement group is clustered to transfer the shape into more rectangular shape.

Consider (x, y) as point coordinate belong to Arabic word W and β is angle between start and end points after y coordinate is changing direction as shown in figure 4. Then, $\forall (x,y) \in W$ we obtain

$$g \in W_{(U,D,L,R)} : g = \begin{cases} U & \Delta x = 0, \Delta y > 0 \\ D & \Delta x = 0, \Delta y < 0 \\ L & \Delta x > 0, \Delta y = 0 \\ R & \Delta x < 0, \Delta y = 0 \\ RU & \Delta x < 0, \Delta y > 0 \\ RD & \Delta x < 0, \Delta y < 0 \\ LU & \Delta x > 0, \Delta y > 0 \\ LD & \Delta x > 0, \Delta y < 0 \end{cases} \tag{3}$$

Where $W_{(U.D.L.R)}$ is the word movement vector of combined direction Up, Down, Left and Right. The result does not include diagonal movement which is transferred into equal RU, RD, LU or LD movement vector as shown in figure 3.

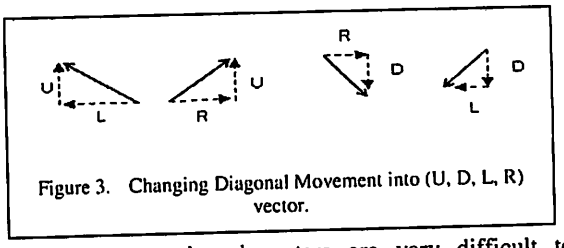


Figure 3. Changing Diagonal Movement into (U, D, L, R) vector.

In some case, the characters are very difficult to segment due to overlapping between consecutive characters' coordinates (Figure 4 (a)). Previous researches [11][13] didn't solve this particular problem due to its difficulty. We propose a method for modifying equation (3) above to exclude movements from left to right (RU and RD) to become (URU and DRD) since these movements forces overlap to occur. This transformation will reduce the length of the character movement towards R vector and increase the movement towards U or D vectors as shown in figure 4 (b).

The clustering operation intended to remove single or double movement in specific direction between two dominant different direction vectors. Let's consider g_i to be the count of consecutive movement for specific direction of U, D, L or R. Then, $G_{W(U,D,L,R)}$ consist of a set of g_{ij} i=1,2,....n and jϵ{U,D,L,R}. Then clustering is performed as;

$$\forall g_{ij} < 2 \text{ (empirically)}, g_{ij} = \begin{cases} g_{(i-1)j} & \text{if } g_{i-1,j} \geq g_{i-1,j} \\ g_{i-1,j} & \text{if } g_{i-1,j} < g_{i-1,j} \end{cases} \quad (4)$$

Figure 5 shows the Arabic word عمل as written by the user. It has inconsistent writing shape to be segmented efficiently. Figure 5 (b) shows the character after smoothing where jittered movements are removed and the shape is more consistant. Then, after applying equation (1) through (5) the shape becomes a flat shape with no diagonal movement as shown in figure 5 (c). At this stage the character is more recognizable as character and ligature.

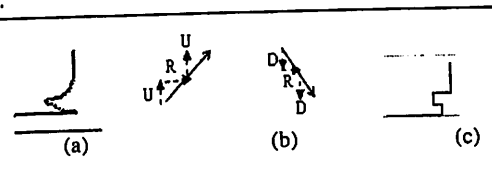


Figure 4. Changing Diagonal Movement into (U, D, L, R) vector. (a) Original overlapped characters, (b) Diagonal movement transformation, (c) Characters after preprocessing, transformation and diagonal movement transformation.

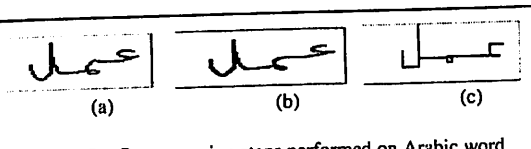


Figure 5. Preprocessing steps performed on Arabic word Arabic word عمل. (a) Original written word. (b) Word after smoothing. (c) Word after preprocessing, coordinate transformation and clustering.

C. *x-Coordinate Scan and Segmentation*

The output shape of figure 4(c) consists of line segments belong either to character body or to line joining these characters. All Arabic characters share one common feature. Character body writing style has the same x-coordinate passed through twice or more. Consider (x^e, y^e) to be the point coordinate along the Arabic word after applying equation (2) above. Then, coordinates belong to line joint segment (x^s, y^s) is obtained as:

$$(x_i^s, y_i^s) = \begin{cases} 0 & \forall (x_i^e, y_i^e) = (x_j^e, y_j^e) \\ (x_i^e, y_i^e) & otherwise \end{cases} \quad (5)$$

Where as, points belong to character body (x^c, y^c) is obtained as:

$$(x_i^c, y_i^c) = \begin{cases} (x_i^e, y_i^e) & \forall (x_i^e, y_i^e) = (x_j^e, y_j^e) \\ 0 & otherwise \end{cases} \quad (6)$$

Where i=1....N and j=1.....N.

Figure 6 (a) shows the result of extracting character body (red color) from joint lines (black color). As noted the segmentation processes so far is working very well and the lines in the black color represents the ligatures between the characters in the word. But the only error represents the selection of first character starting movement and the last character horizontal movement as ligatures instead of character body. In next section these two problems are overcome.

The segmentation points between characters are calculated as follows:

$$P_S(x_k, y_k) = \left(\frac{x_{sk}^S + x_{ek}^S}{2}, y_k^S \right) \quad k = 1...n \quad (7)$$

Where Ps(xk, yk) is the kth segment point, n is number of segments, x_{sk}^S is the starting x coordinate for kth segment, x_{ek}^S is the end x coordinate for kth segment, y_k^S is the y coordinate for kth segment. Figure 6 (b) shows the segmented characters after applying equation (5).

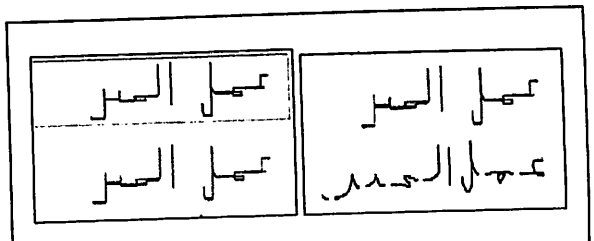


Figure 6. Results of applying (a)x-coordinate scan. Red color highlights the character body while black highlight the joint line. (b) Character segments at segment points.

D. *First and Last Segment Filtering*

Some of the character such as ع or ك in the first segment or characters such as بفلرو in the end

segment may cause segmentation error (as shown in figures 6 and 7). These segments must be filtered to remove ambiguity regarding the segment location. Two filters are used to distinguish between start or end character movement from actual segment locations. These two filters are described as follows:

D.1. Start Segment Filtering

Consider L_s to represent a straight line from segment start point (x_s, y_s) to its end and having (x_{ls}, y_{ls}) coordinate. Then, removing the segment is subject to the condition:

$$\forall x_{ls} \in L_s \exists \mid y_{ls} - y_s \mid > 0 \qquad (8)$$

Equation 8 implies that if the variant in y direction, related to first character, is 0 then the movement is in straight line. Thus, this movement is a part of the first character. Otherwise, if there is a rapid change in the movement then Δy will have a value greater or less than zero representing the ligature following the first character.

D.2. End Segment Filtering

Consider y_s to be the y coordinates of the end ligature point. x_c and y_c is the character (x, y) coordinate. then,

$$\Delta y_i = (y_c - y_s) \qquad (9)$$

The filter removes the ligature if it is subject to one of the following conditions:

$\Delta y = \Delta y_i - \Delta y_{i-1} < 0$ and $\theta < 70$ for all i=1,...N.
Or,
$\Delta y = \Delta yi - \Delta yi\text{-}1 > 0$ and $\theta < 70$ for all i=1,...N.

Where N is the number of ligature points and θ is the angle from the ligature point to the last point in the current segment as shown in figure 7. This filter will ignore final characters such as (ـ or ـ) since the first has $\Delta y > 0$ and $\Delta y < 0$ along its movement, while the second has $\Delta y > 0$ and $\theta > 70$o along its movement. Figure 8 shows the final refined result after applying the filter to the word. It can be seen that the segmentation is done exactly at the positions that it must be.

III. Results and Discussion

The segmentation system was tested with 5 different sentence templates to insure word variation and increasing the possibilities of error to take place (figure 9). These sentences contain a wide combination of difficult segmentations such as overlap, first character error, end character error, jitter in writing.

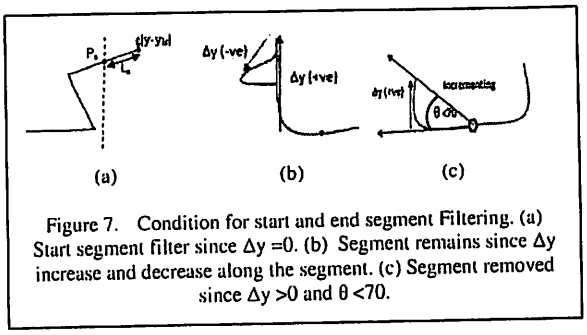


Figure 7. Condition for start and end segment Filtering. (a) Start segment filter since $\Delta y = 0$. (b) Segment remains since Δy increase and decrease along the segment. (c) Segment removed since $\Delta y > 0$ and $\theta < 70$.

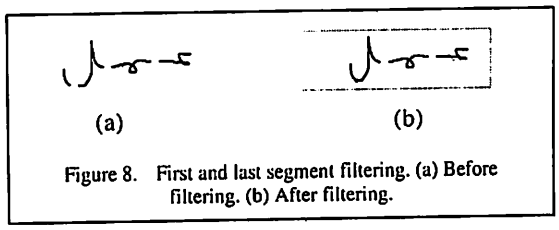


Figure 8. First and last segment filtering. (a) Before filtering. (b) After filtering.

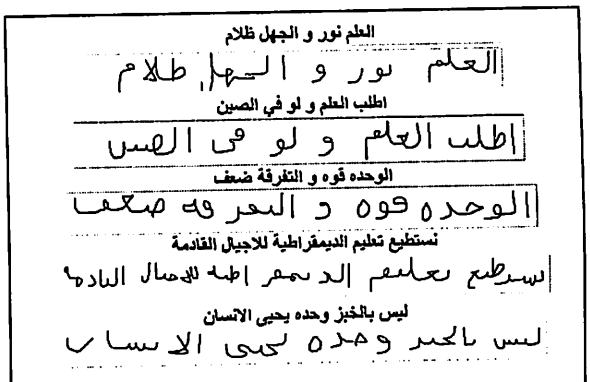


Figure 9. Word templates used to test the segmentation.

The system was tested with 20 different persons to obtain the success percentage of the segmentation depending on how many characters were segmented successfully. These users were asked to perform normal style writing with natural technique. They were also asked to write with speedy and inconsistent writing technique to test the limit of the system. Table 3 shows the result of the segmentation applied to the sentence templates. The results represent the percentage of successfully segmenting and filtering 580, 600, 200 and 1580 of first, last, overlap and normal characters included in these cursive words out of total 2300 characters.

Table 2 shows the result of segmentation using word template of Fig 9. It can be noted that normal character has scored the highest percentage (98.03%) with only characters such as س-ص that contain "rogza" will affect the recognition rate due to ambiguity of separating ligature as it match the character body. The first character filter obtained 94.82% errors occurs due to overlapped segment or miss segment of س-ص. The Last character filter obtained 97.33% with the same reason. The overlapped characters scored 90% with errors mostly occurred due to high overlapping cases as illustrated in figure 10.

The system initially has achieved a very promising result to be used as a first step in online Arabic words recognition system.

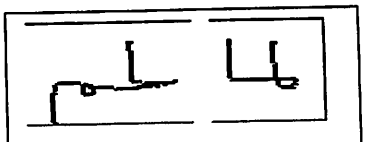


Figure 10. Difficult overlapping cases

TABLE II. TEST RESULT FOR THE TEMPLATES IN FIGURE 10

Group	Total successful segments	Total miss segments	Percentage
Normal	1549	41	98.03%
First Character filter	550	10	94.82%
Last Character filter	584	16	97.33%
Overlapped	180	20	90%
All Characters	2213	87	96.2%

IV. CONCLUSION AND FUTURE WORK

We presented a technique for character segmentation from online Arabic cursive writings. The method is based on statistic and transformation of character movement vector. The segmentation depends on flattening the word and extract movement having uncommon x coordinate that represent possible ligature positions. Then, a combination of two filters is applied to refine the start and last segment of the word. The system were tested with varies Arabic texts to examine the performance of the system. The system scored 98.03%, 94.82, 97.33 and 90% for normal segmentation, starting segment, last segment and overlapped segment success rate. The errors presented in the test are common problem for Arabic as high overlap and rogza problem always exists.

The proposal for future work is to apply a recognition system for testing the system performance under recognition. In addition, it is required to solve high overlap and Arabic rogza segmentation errors.

V. ACKNOWLEDGEMENT

We would like to thank Ministry of Higher Education/Malaysia grant number 203/PELELT/6071189 for their assistant and contribution in finishing this work.

REFERENCE

[1] T. J. Klassen and M. I. Heywood, "Towards the on-line recognition of Arabic characters," in *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*, 2002, pp. 1900-1905.

[2] A. M. Namboodiri and A. K. Jain, "Online script recognition," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, 2002, pp. 736-739 vol.3.

[3] Z. Osman, L. Hamandi, R. Zantout, and F. N. Sibai, "Automatic processing of Arabic text," in *Innovations in Information Technology, 2009. IIT '09. International Conference on*, 2009, pp. 140-144.

[4] T. S. El-Sheikh and S. G. El-Taweel, "Real-time Arabic handwritten character recognition," in *Image Processing and its Applications, 1989., Third International Conference on*, 1989, pp. 212-216.

[5] S. Al-Emami and M. Usher, "On-line recognition of handwritten Arabic characters," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, pp. 704-710, 1990.

[6] A. M. Alimi, "A neuro-fuzzy approach to recognize Arabic handwritten characters," in *Neural Networks,1997., International Conference on*, 1997, pp. 1397-1400 vol.3.

[7] G. Al-Habian and K. Assaleh, "Online Arabic handwriting recognition using continuous Gaussian mixture HMMS," in *Intelligent and Advanced Systems, 2007. ICIAS 2007. International Conference on*, 2007, pp. 1183-1186.

[8] Randa I. Elanwar, Mohsen A. Rashwan, and Samia A. Mashali, Simultaneous Segmentation and Recognition of Arabic Characters in an Unconstrained On-Line Cursive Handwritten Document, World Academy of Science, Engineering and Technology 29, 2007, pp. 288-291.

[9] M. Hussain and M. N. Khan, "Online Urdu Ligature Recognition using Spatial Temporal Neural Processing," in *9th International Multitopic Conference, IEEE INMIC 2005*, 2005, pp. 1-5.

[10] R. Saabni and J. El-Sana, "Hierarchical On-line Arabic Handwriting Recognition," in Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on, 2009, pp. 867-871.

[11] K. Daifallah, N. Zarka, and H. Jamous, "Recognition-Based Segmentation Algorithm for On-Line Arabic Handwriting," in *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, 2009, pp. 886-890.

[12] M. Kherallah, L. Hadad, and A. M. Alimi, "A new Approach for Online Arabic Handwriting Recognition," in *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, 2009, pp. 22-25.

[13] B. Alsallakh and H. Safadi, "AraPen: An Arabic Online Handwriting Recognition System," in *Information and Communication Technologies, 2006. ICTTA '06. 2nd*, 2006, pp. 1844-1849.

[14] A. Elbaati, M. Kherallah, H. El Abed, A. Ennaji, and A. M. Alimi, "Arabic handwriting recognition using restored stroke chronology," in *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, 2009, pp. 411-415.

[15] Http://en.wikipedia.org/wiki/Arabic_alphabet

[16] M. Y. Potrus, U. Ngah, H. A. mat Sakem and S. A. Abdulrahman, "Normalization and Rectification Method for Online Hindi Digit Recognition with Partial Alignment Algorthm", Accepted in The 2010 International Conference on Electronics and Information Engineering (ICEIE 2010).

A Hidden Markov Model-Harmony Search Combination for Online Arabic Character Recognition

Moayad Yousif Potrus[1], Umi Kalthum Ngah[2], Harsa Amylia Mat Sakim[3]

School of Electrical and Electronic Engineering
University Sins Malaysia, Malaysia
myp.ld09@student.usm.my[1], eeumi@eng.usm.my[2], harsaamylia@ieee.org[3]

Abstract

In this paper a hidden markov model and harmony search algorithms are combined for Arabic character recognition. The markov model works as group classifier instead of a main character classifier/recognizer as most of previous work did. Markov model classifies each group of characters (according to their forms) into small sub groups based on common features. This process works as a time reduction by limiting the number of candidate characters for Harmony search recognizer. The harmony search recognizer uses a dominant and common movement pattern as a fitness function. The objective is to minimize the matching score according to the fitness function criteria. The system tested on a database of 4500 words forming 21234 characters in different positions or forms (isolated, start, middle and end). The data set was divided into training set of 7500 characters and a testing set of 13734 characters. The algorithm test results was compared to that obtained by using HMM and HS each alone.

Keywords

Character Recognition, Evolutionary Computation, Arabic Character Recognition, Hidden Markov Model, Harmony Search.

1. Introduction

Arabic character recognition has gained research interest in the past decade. Many competitions are being held with respect to document processing and recognition during conferences to determine successful recognition systems and their performances in this area [14,15]. The research motivation in Arabic characters recognitions originates from its special and complicated styles of different writing forms. Complications in Arabic recognition comes from the possibility of writing a single word with one stroke or with many strokes, writing a single character with many styles and the changes in character states when diacritics (dots, hamza,...etc) are added.

The use of Hidden Markov Model (HMM) was widely considered in recognition systems since it provides fast results and good recognition rates. It has been used in cursive style writings such as Arabic , Korean and Tamil. In the Arabic recognition systems, the matched character could be determined such that each character is represented as an image and for each character a specific set of momentums are calculated and compared [7]. Moreover, HMM was combined with soft computing methods such as neural network [17] or fuzzy logic[19] or with other methods such as re-ranking[2] to enhance the performance of HMM recognition rate.

Evolutionary computation algorithms provided an efficient calculation mechanism for hard NP problems [6]. These algorithms provide near optimum solutions except that the time spent to find the solution is lengthy, making it inefficient for real time applications. Genetic algorithm (GA) was initially used in offline character recognition systems for optimizing feature selection problems. The generated population was used to find the smallest feature subset from a wider feature range which optimizes the separability between different classes. The algorithm was tested successfully on digit dataset [5, 20] and Persian [21]. GA was used as character recognition which was successfully applied to Latin [11] and Arabic [1]. In addition to GA, other heuristic methods were also used in character recognition such as swarm optimization. Nebti *et al.* [16]applied particle swarm optimization (PSO) and a combination of back propagation neural network and bee colony algorithm for digit recognition based on momentum features. Particle swarm was used as a statistical classifier for comparing the generated feature with the digit data set feature to obtain the optimal class. The back propagation-Bee colony combination were used to determine the assigned class using back propagation to classify the digits. In case that no classification is obtained, the bee colony would be used to assign the digit class. Genetic algorithm was used in online Arabic character recognition system, in which, the process of finding the best match was based up on the process of minimization the bit comparison score between two directional statistical chromosomes of the written character and the full character dataset[13].

HS algorithm has -proven to be faster and more efficient in solution search when compared to genetic algorithm [8]. Building new improvisations may involve all the entities in Harmony matrix which may trigger ventures for new solutions or may lead to optimum final solution. In this paper, a combination of HMM and HS algorithm is used for online handwritten Arabic characters. HMM is used as a classifier which extract a small subset of characters form the entire character set. These characters are grouped according to similarity measures based on direction vector. Then, HS algorithm is applied to select the best matched characters from the classified subset of closely similar characters. In short words, HMM is used as the initial step for minimizing the number of characters before HS is applied to determine the matched character.

2. Arabic Character System

Arabic language and characters is used by more than 600 million peoples and its usage is especially extensive in large parts of Asia. Its character set are basically used for other language's character sets such as Persian, Urdu and Jawi. It consists of 28 letters in its basic isolated form and its form of writing is cursive directed from right to left. The cursive form changes the shape of the character when located at specific positions and this can be categories into four groups: isolated, initial, medial or final form as shown in Fig.1. The number of characters is not fixed in each form and does not include all the characters. It is specified as 28 in the isolated form, 23 in the start form, 23 in the middle form and 28 in the end form. Moreover, the main shape of some of these letters are the same (e.g. (ب,ت,ث) or (ى,ي,ئ)) while only the punctuation (dot or hamza ء) will make these letter differs from each other.

Arabic character recognition complexity represents a good example of how complex a recognition system could be. Most of the characters have widely common features which may trigger recognition errors. In addition, the cursive nature of the Arabic writings lead to a plethora of letter shapes and writing styles which are dependent upon each individual writer's style. Moreover, the standard writing movement for any letter could be easily changed, subject to the person's educational background or preferences. Other factors, which may affect the quality or clarity of writings are the instability of the writing devices (e.g. writing pen and tablet device) which may cause inconsistency in the writing patterns, non-continuous connected patterns and character hocks which might occur at the start and end of the writing process. Fig. 2 illustrates some of the common characteristics and difficulties which may be implicated in the Arabic character system.

Isolated	Initial	Medial	Final	Isolated	Initial	Medial	Final	Isolated	Initial	Medial	Final
ا	ا	ـا	ـا	ر	ر	ـر	ـر	ف	فـ	ـفـ	ـف
ب	بـ	ـبـ	ـب	س	سـ	ـسـ	ـس	ك	كـ	ـكـ	ـك
ت	تـ	ـتـ	ـت	ش	شـ	ـشـ	ـش	ل	لـ	ـلـ	ـل
ث	ثـ	ـثـ	ـث	ص	صـ	ـصـ	ـص	م	مـ	ـمـ	ـم
ج	جـ	ـجـ	ـج	ض	ضـ	ـضـ	ـض	ن	نـ	ـنـ	ـن
ح	حـ	ـحـ	ـح	ط	طـ	ـطـ	ـط	ه ھ	هـ	ـهـ	ـه
خ	خـ	ـخـ	ـخ	ظ	ظـ	ـظـ	ـظ	و	و	ـو	ـو
د	د	ـد	ـد	ع	عـ	ـعـ	ـع	ي	يـ	ـيـ	ـي
ذ	ذ	ـذ	ـذ	غ	غـ	ـغـ	ـغ				
ر	ر	ـر	ـر	ف	فـ	ـفـ	ـف				

Figure 1: Arabic letters data set with different letter form.

3. Character Recognition System

The proposed recognition system considers the characters are already segmented and ready to be recognized. It consists of three steps: preprocessing, classification and recognition. Fig. 3 shows the description of the system used in online Arabic character recognition.

3.1. Preprocessing

In the preprocessing step, the characters are enhanced and smoothened to remove writing errors and jitters, in addition to the written character problems described in the previous section and shown in Fig.2 (a and b). Bezier cubic curve approximation algorithm [9] is used to fill the gaps between the characters points, as well as, smoothing the character shape. The algorithm splits the character into groups of four points and then applying the curve approximation using the formula,

$$BZ(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t)P_3 + t^3 P_4 \quad (1)$$

where P represents the character points and t is a parameter varying between [0, 1]. The result of this approximation will be, all the gaps will be filled with substitution points. However, redundant points will be produced along the curve. The application of the algorithm thereafter will remove all the redundant points to produce a consistent set of character features. In addition, a hock removing algorithm is used to remove hocks from the start and end of the characters by detecting the existence of sudden changes in the angle direction [10].

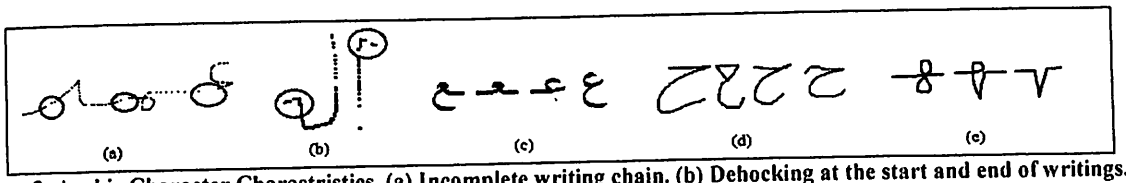


Figure 2. Arabic Character Charastristics. (a) Incomplete writing chain. (b) Dehocking at the start and end of writings. (c) Changes in the letter ع shape at different positions (isolated, start, middle and end). (d) different writing styles for letter ح. (e) Different movement patterns for Arabic letter ه at the middle position.

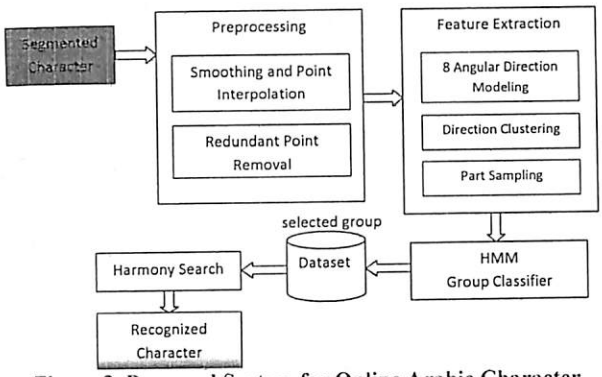


Figure 3. Proposed System for Online Arabic Character Recognition.

3.2. Feature Extraction

Arabic characters set are divided into four groups according to the positional form (isolated, initial, medial and final). For each group, the characters are divided into subgroups consisting of a few groups according to their starting direction feature. The starting direction of Arabic writings can be adopted in order to divide the characters into the subgroups. The main features are:

1. Forward-downward starting curvature writing direction, in which the writing starts along the right to left direction and then curves downwards such as in characters (ء،ى،ع).
2. Forward-upward starting curvature writing direction, in which the writing starts along the right to left direction and then curves upwards such as in the loop based characters (ف،م،ق،و).
3. Backward starting curvature writing direction, in which the writings start opposite to the right to left writing direction such as in characters (ه،ص،د،ح).
4. Vertical-Downward starting direction, in which the starting movement is directed downwards such as in characters (ط، ل،س، ر،ب،ا).

This division shows how the system can divide the main 18 characters (non punctuated character) structures into only 4 subgroups containing 4 to 5 characters. The sub-grouping characteristic can be applied to the other three forms (initial, medial and final) by taking the starting movement, middle movement and end movement respectively. Table 1 shows the division of Arabic characters forms into their subgroups according to the movement feature.

Table 1: Direction Based Sub Grouping of Different Arabic Characters Forms

Isolated Groups (Starting Movement)	Initial Groups (Starting Direction)	Medial Groups (Middle Direction)	Final Groups (End Direction)
(ء،ى،ع) (ف،م،ق،و) (ح،د،ص،ه) (ط، ل،س، ر،ب،ا)	(ك،ع،ف) (س،ا) (خ،ص،ه) (ب،ل،ط)	(ح،خ،ه) (ب،ل) (ع،ق،ص)	(ق، أ،ى،ب) (ص،س،د) (ش،ر،و) (م،ع،ح)

The portion on character feature consists of a number of points which are transformed into feature direction vector using angular direction models based on the eight main directions shown in Fig. 4 (a). This directional vector is clustered to remove the vectors having minimum effects while the dominating directions are preserved. The clustering algorithm starts with changing the direction chain into a direction run length chain. If a character movement C_D consists of N directions of vector v, then after the running length, the encoding will be represented as,

$$C_D = \{v_1(N_1)v_2(N_2)\ldots v_m(N_m)\} \quad (2)$$

where $\sum_{i=0}^{m} N_i = N$

The clustering technique is applied for the reallocation of vector v_i which lies between the vectors v_{i-1} and v_{i+1}, such that, if v_i value falls under or is equal to a threshold value then it is reallocated as,

$$v_i = \begin{cases} v_{i-1} & v_{i-1} \geq v_{i+1} \\ v_{i+1} & v_{i-1} < v_{i+1} \end{cases} \quad (3)$$

A threshold value of 2 was found to be convenient for this work (which was obtained empirically). Fig. 4(b) shows the result of applying clustering technique on the directional vector for character ع. It can be noted from the figure that the final vector only consists of a long chain of the dominant movements. The selected movement direction for training will consist of the direction vectors 7, 6 and 4.

3.3. Hidden Markov Model Classifier

HMM is a statistical tool which can be used efficiently for classification purposes [18]. HMM is presented by the model $\lambda=(A, B, \pi)$, with A representing the transitional probabilities between the states, B is the observation probability and π is the initial state probability. The full parameters that should be considered are:

$\pi=\{\pi_i=P(s_i \text{ at } t=1)\}$
$A=\{a_{ij}=P(s_i \text{ at } t+1 | s_j \text{ at } t)\}$
$B=\{b_i(k)=P(o_k \text{ at } t | s_i \text{ at } t)\}$
$T=$*length of observation sequence.*

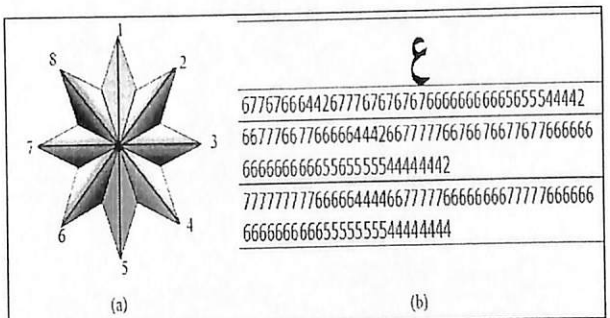


Figure 4. Encoding and sampling technique for Arabic Characters. (a) Eight directional model for character encoding. (b) Character ع encoding and sampling.

N=number of states in the model.
M=number of observation symbols.
S={s_i} 1≤ i ≤ N, states.
O={o_j} 1≤ i ≤ M, discrete set of possible observation symbols

The challenge in HMM modeling is to determine the values of *A*, *B* and π parameters using a set of training sequences. For the Arabic character classification, HMM model with left to right state configuration is used with each state transition to the next following states or return to itself is as shown in Fig. 5. To train this model, a number of states are assigned to each character according to the number of direction parts in its structure (e.g., 3 states for the character ع in Fig 3). The output of the training process is a number of models equal to the number of character per set per form. These models are used to determine the closest similarity between the input vector and the target subset of those shown earlier in table 1. The output subset will be processed by the recognition phase in next section.

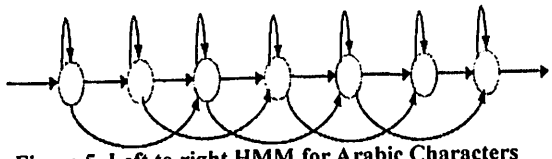


Figure 5. Left to right HMM for Arabic Characters Classification

3.4. Harmony Search Recognizer

Harmony search algorithm uses the same concept as in all evolutionary algorithms i.e. which is based on random population generation. It consists of a harmony memory with size (HMS) which contains the generated population and their corresponding solution,

$$HM = \begin{bmatrix} x_1^1 & \cdots & x_n^1 & f(x^1) \\ \vdots & \ddots & \vdots & \vdots \\ x_1^{HMS} & \cdots & x_n^{HMS} & f(x^{HMS}) \end{bmatrix} \quad (4)$$

Where *n* is the number of variables used to find the function *f*. The solution is constructed by selecting random values from HMS or from a vector bounded by values x_{min} and x_{max} depending on the value generated from a random variable called harmony memory consideration rate ($0 \le HMCR \le 1$).

$$x_i^{new} = \begin{cases} \in HM & propability = HMCR \\ \in [x_i^L, x_i^U] & propability = HMCR \end{cases} \quad (5)$$

When harmony memory value is selected, another parameter is used to decide whether this value is picked or to be tweaked to another value called the pitch adjustment rate (PAR). In discrete harmony search, the PAR adopts the neighbor value according to the shift parameter.

$$x_i^{new}(k) = \begin{cases} x_i^{new}(k) & propability = 1 - PAR \\ x_i^{new}(k+m) & propability = PAR \end{cases} \quad (6)$$

Where m ϵ [-1,1] is the shifting value. To consider the shift direction, a shifting parameter (SP) is used and defined by:

$$x_i^{new}(k) = \begin{cases} x_i^{new}(k-1) & propability = SP \\ x_i^{new}(k+1) & propability = 1 - SP \end{cases} \quad (7)$$

The most important part in metaheuristic optimization problems is to define the objective function. To define such function, let us consider the characters ع, ح and ه which has similar structure in the end part for (ح،ع) and similar starting structure for (ه،ع). The matching function can be defined based on movement structure by partitioning all the characters into similar number of parts *N*. This representation can be separated into 3 parts: start, middle and end movement parts which can be compared separately. For each part, a weight is given to separate the matching according to the character forms, Thus, the matching function can be written as,

$$f_{match} = \begin{cases} w_s C_s + w_m C_m + w_e C_e & Isolated \\ w_s C_s + w_m C_m & Initial \\ w_s C_s + w_m C_m + w_e C_e & Medial \\ w_s C_s + w_m C_m + w_e C_e & Final \end{cases} \quad (8)$$

with,

$w_s = w_e > w_m$ *Isolated*
$w_s > w_m$ *Initaial*
$w_m > w_s > w_e$ *Medial*
$w_e > w_s > w_m$ *Final*

where w is the weight value assigned to the character part. and C is the matching function defined as,

$$C = \sum_{i=1}^{N} V_i \rightarrow V_i = \begin{cases} -1 & v_t = v_s \\ 1 & otherwise \end{cases}$$

where, v_t *and* v_s are the target and source directions. From (8), it can be noted that the weight value is considered differently with each form. For the initial form, the start and middle part is only considered in the matching process because the end part represents the connector between the characters and is similar in all of them. For medial and final form, the start and middle part is mostly considered since most characters are similar in the end part. The matching function cannot be considered as the only scoring function because in some cases, the matching process fails to select the right character. To enhance the scoring function, a parameter p_i^d represents the probability of a direction *d* at position *i* for each character in the training dataset. It is used to reduce the possibility of incorrect direction or non-existent direction to occur in the new improvised harmony vector. For the chosen direction model, each character in the training dataset has N by 8 probability value array to reflect the direction probability per position as,

$$\begin{bmatrix} p_1^1 & \cdots & p_N^1 \\ \vdots & \ddots & \vdots \\ p_1^8 & \cdots & p_N^8 \end{bmatrix}$$

The total direction vector probability is found to be,

$$P = \sum_{i=1}^{N} p_i^d \tag{9}$$

Thus, the final matching objective function to be minimized is defined as,

$$Min \quad F_{score} = f_{match} \times \frac{1}{P} \tag{10}$$

The parameter 1/P minimizes the scoring function whenever the improvised direction is correctly located in the right position between the source and target direction vectors. Whereas, the matching function is minimized when the direction vectors of the dataset and the target are closely matched.

4. Results and Discussion

In order to test the system, a data set of 24960 characters is used. The data set is divided into two parts: 7500 characters are used in HMM training and HS comparison, while 14460 characters are used for testing the HS recognizer. In the HMM training process, the training character dataset of 7500 characters is used. The HMM trained model has at most 5 states and a minimum of 2 states to define each character (the number is specified according to the character dataset direction model). At the end of the training process, the system is tested with the full dataset of 24960 characters to determine the successful classifying rate of each character into its corresponding subgroup. Table 2 shows the result of this classification which scored an overall accuracy of 95.87%. It can be noted that the most number of classification errors occurred in isolated form since many of the written samples may interfere with other groups due to the writing styles.

Table 2: Results of HMM Classification for the trained and test data

Form	No of Samples	Correctly Classified	Errors	Accurcy
Isolated	6120	5800	320	94.77%
Start	9240	9130	150	98.81%
Middle	3680	3490	190	94.84%
End	5650	5370	280	95.04%
Overall Accuracy				**95.87%**

To test the HMM-HS recognizer performance, a comparative study is made to show how metaheuristic optimization can be integrated with other algorithms to be adapted in real time system. The comparative results obtained are intended to show the performance in using HS as a standalone recognizer against the combined HMM-HS classification-recognition process. For this purpose, the recognition test is carried out using segmented words with different character length to determine the recognition time and success rate. The accuracy and time obtained from the test represent the average value obtained after 20 tries per sample. The obtained result is shown in table 3 and Fig. 6.

Table 3. Comparison between HS and combined HMM-HS Algorithm

No of Characters/word	Time HS (ms)	Time HMM-HS (ms)	Recognition (HS)	Recognition Rate (HMM-HS)
1	8400	500	78%	94.50%
2	16250	863	81%	94%
3	18140	1500	82%	93.88%
4	19700	3000	82.10%	93.81%
5	20500	3400	82.15%	92.70%
6	20800	3800	82.15%	92.70%
7	21430	4200	82.15%	92.70%
8	21840	4500	82.15%	92.70%

From the results, it is shown that the time factor is tremendously reduced by a factor ranges from 5 to 10 times when using HMM-HS compared to HS recognizer. In addition, the recognition accuracy is enhanced from 82% to 92% of the full character set of different forms. This shows that metaheuristic optimization algorithms can be used for online recognition process by combining it with other methods to enhance the time and accuracy of the recognition system.

5. Conclusion

The results shown in table 3 and Fig. 6 demonstrates that HMM-HS method scores a higher recognition rate when compared to using HS alone. This proves that the HMM subset classification assignment helps to find a smaller group or subgroup of characters with similar features instead of searching a whole group of characters. In addition, the time reduction obtained from the HMM-HS process is significantly large which reaching only 16.7% of that spent to find the matched character when using HS alone. Therefore, it can be concluded that, using a combination of evolutionary algorithms with HMM can result in a good recognition rate and performance. The time obtained from using this combination can be considered significant as a modification for evolutionary algorithm slow search process, thus, allowing a successful integration in online recognition systems.

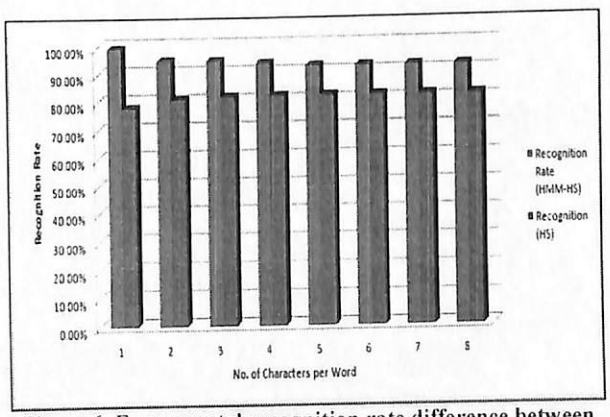


Figure 6. Expermental recognition rate difference between HMM-HS and HS.

References

[1] Aljuaid, H., Mohamad, D., & Sarfraz, M. (2011). Evaluation Approach of Arabic Character Recognition. International Journal of Computer Vision and Image Processing (IJCVIP), 1(2), 58-77.

[2] AlKhateeb,J.; Ren, J.; Jiang,J.; Al-Muhtaseb, H., Offline handwritten Arabic cursive text recognition using Hidden Markov Models and re-ranking, Pattern Recognition Letters, Volume 32, Issue 8, 1 June 2011, pp 1081-1088.

[3] Alma'adeed, S.; Higgens, C.; Elliman, D.; , "Recognition of off-line handwritten Arabic words using hidden Markov model approach," Pattern Recognition, 2002. Proceedings. 16th International Conference on , vol.3, no., 2002.

[4] Choonsuk Oh; Woo Sung Kim; , "Off-line recognition of handwritten Korean and alphanumeric characters using hidden Markov models," *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on* , vol.2, no., pp.815-818 vol.2, 14-16 Aug 1995.

[5] Cordella, L.; De Stefano, C.; Fontanella, F.; Marrocco, C.; , "A feature selection algorithm for handwritten character recognition," *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* , vol., no., pp.1-4, 8-11 Dec. 2008.

[6] Eiben, A.E.; Smith, J.E., Introduction to Evolutionary Computing, Springer, Natural Computing Series,1st edition, 2003.

[7] El-Feghi, I.; Elmahjoub, F.; Alswady, B.; Baiou, A.; , "Offline handwritten Arabic words recognition using Zernike moments and Hidden Markov Models," Computer Applications and Industrial Electronics (ICCAIE), 2010 International Conference on , vol., no., pp.165-168, 5-8 Dec. 2010.

[8] Geem, Z. W., "Music-Inspired Harmony Search Algorithm, " Springer, 2009.

[9] Hain, T.F.; Racherla, S.V.R.; Langan, D.D.; , "Fast, precise flattening of cubic Bezier segment offset curves," *Computer Graphics and Image Processing, 2004. Proceedings. 17th Brazilian Symposium on* , vol., no., pp. 244- 249, 17-20 Oct. 2004.

[10] Huang, B.; Zhang, Y. B.; Kechadi,M., Preprocessing Techniques for Online Handwriting Recognition, Intelligent Text Categorization and Clustering 2009, pp. 25-45.

[11] Kala, R.; Vazirani, H.; Shukla, A. ; Tiwari, R., Offline Handwriting Recognition using Genetic Algorithm, International Journal of Computer Science Issues, Vol. 7, Issue 2, No 1, March 2010, pp. 16-25.

[12] Kannan, R.J.; Prabhakar, R.; Suresh, R.M.; , "Off-line Cursive Handwritten Tamil Character Recognition," *Security Technology, 2008. SECTECH '08. International Conference on* , vol., no., pp.159-164, 13-15 Dec. 2008.

[13] Kherallah, S., Bouri, F.; Alimi, A.M. , "On-line Arabic handwriting recognition system based on visual encoding and genetic algorithm, " Engineering Applications of Artificial Intelligence, 2009. 22(1): p. 153-170.

[14] Margner, V.; El Abed, H.; , "ICDAR 2009 Arabic Handwriting Recognition Competition," *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on* , vol., no., pp.1383-1387, 26-29 July 2009.

[15] Margner, V.; Pechwitz, M.; Abed, H.E.; , "ICDAR 2005 Arabic handwriting recognition competition," *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on* , vol., no., pp. 70- 74 Vol. 1, 29 Aug.-1 Sept. 2005.

[16] Nebti, S., Boukerram, A., Zavoral, F., Yaghob, J., Pichappan, P. & El-Qawasmeh, E. , Handwritten digits recognition based on swarm optimization methods,Networked Digital Technologies, Springer, Communications in Computer and Information Science, 2010, Volume 87, Part 1,pp. 45-54 .

[17] Narima, Z.; Messaoud, R.; Mouldi, B.; , "Neuro-Markovian hybrid system for handwritten Arabic word recognition," Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003 10th IEEE International Conference on , vol.2, no., pp. 878- 881 Vol.2, 14-17 Dec. 2003.

[18] Rabiner, L.R.; , "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE* , vol.77, no.2, pp.257-286, Feb 1989.

[19] Razzak, M. I.; Anwar, F.; Husain, S.A. ; Belaid, A.; Sher, M., "HMM and fuzzy logic: A hybrid approach for online Urdu script-based languages' character recognition," Knowledge-Based Systems, Volume 23, Issue 8, December 2010, pp 914-923.

[20] Shi, D.; Shu, W.; Liu, H., "Feature selection for handwritten Chinese character recognition based on genetic algorithms," *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on* , vol.5, no., pp.4201-4206 vol.5, 11-14 Oct 1998.

[21] Soryani, M. ; Rafat, N., Application of Genetic Algorithms to Feature Subset Selection in a Farsi OCR, World Academy of Science, Engineering and Technology ,18, 2006, pp 113-116.

2010 International Conference on Computer Applications and Industrial Electronics (ICCAIE 2010), December 5-7, 2010, Kuala Lumpur, Malaysia

An Effective Segmentation Method for Single Stroke Online Cursive Arabic Words

Moayad Yousif Potrus[1], Umi Kalthum Ngah[2], Harsa Amylia Mat Sakim[3]
School of Electrical and Electronic Engineering, Universiti Sains Malaysia
Penang, Malaysia
myp.ld09@student.usm.my[1], eeumi@eng.usm.my[2], harsaamylia@ieee.org[3]

Abstract— **A new method is used for character segmentation from cursive Arabic words. The method is based on statistical approach which uses Normalization and rectification, coordinate transformation and clustering to extract ligatures. The output is then filtered to extract start, overlapped and end segment errors. After applying the filter the characters are completely isolated and ready for recognition. The system, when testing the segmentation on 5 different Arabic sentences and by 20 different users, scored 98.03%, 94.82, 97.33 and 90% for normal segment, starting segment, last segment and overlapped segment.**

Keywords—Character Segmentation, Single Stroke, Arabic Word Ligatures.

I. INTRODUCTION

Development in computer devices during the last decade, have led to demanding market for fast and efficient input devices. Computer devices like tablet PC, PDA and Mobile devices are widely used nowadays. These devices depend on touching or writing in order to input data or perform a task. Traditional writing represents the most comfortable style for a user to write documents or enter data, especially those with slow keyboard typing. Therefore, ongoing online recognition studies attempt to meet the users' demands of efficient and accurate recognizers to make it easier for the task of data entry.

The main challenge for online character researches is to find the right form for successfully recognizing the handwritten characters. The problem may be easy to handle for isolated characters [1]. However, for cursive handwritten words, difficulties may be encountered [2]. Cursive words require more processing. Thus, complex algorithms are used to try to separate the characters from ligatures between them [3]. Researchers have taken a great interest in the Arabic language style (similar to Urdu, Farsi and Jawi) for the last decade. The complex and versatile cursive writing style poses as a great challenge to overcome.

There are very limited researches dealing with online cursive Arabic character segmentation. It is very hard to obtain acceptable results due to the word structure complexity. Many researches introduced a method of writing Arabic words as a sequence of separated characters (multiple strokes) rather than a single stroke per word [4] [5] [6] [7]. The problem with this method is that, the user is obliged to use the writing method proposed by the developer instead of his or her own natural writing style. In addition, an error which occurs in one stroke may cause a whole word error. Another attempt that was used depends on partial segmentation of a multi stroke per word [8]. Again, the stroke addressing is subject to correct position and the system ligature position depends on the recognition system decision and verification. Other methods avoided the segmentation process and attempted to recognize the whole word [9] [10]. These systems are very sensitive to the amount of database template and their writing styles. In addition, covering all the words in the Arabic dictionary plus adding multiple templates per word would be timely expensive.

Few researches attempted to segment the characters from the word. Daifullah *et al.* [11] proposed a method of segmenting the characters based on detecting right to left movement that have angles smaller than θ=30o and then calculating all possible segment combination. This method predicted the possible segments location and depended on recognition for specifying the correct combination. This made it very sensitive to segment the location error when the character overlapped or a mis-segmentation occurs. Kherallah *et a.l* [12] proposed a method to separate characters from ligatures using stroke speed measurement. They suggested that characters can be separated from ligatures whenever a rapid stroke speed change occurs. This assumption can easily fail whenever the stroke speed was constant or the speed difference was not that significant. Alsallakh *et al.* [13] proposed a segmentation method based on horizontal movement detection and dynamic time wrapping for character movement comparison. The system is highly sensitive for curvature ligatures and overlapping between the characters. Abdulkarim *et al.* [14] suggested that, segmentation of pseudo-words in graphemes rests on the detection of two types of points which are typographically significant. These points are summits of the valleys bordering the base line with a parallel tangent and angular points. The word may not have a base line as reference for segmentation as the writing style may impose multi-reference lines due to imperfect writing styles.

In this paper, we propose an effective statistical segmentation technique based on normalization, direction transformation and clustering, to determine the segment location. Afterwards, filters are used to correct segmentation error occurs at the first, last and overlap character. The proposed segmentation system can deal with curved, jittered and noisy single stroke Arabic word. The central idea of our system is to flatten the words to represent a multi connected line. These lines are filtered to obtain the ligature position.

II. SYSTEM DESCRIPTION

The Arabic characters in a cursive writing are classified into four pattern based on their position. These

+0484

UNIVERSITI SAINS MALAYSIA

MEMO
PEJABAT TIMBALAN DEKAN
PUSAT PENGAJIAN KEJURUTERAAN ELEKTRIK & ELEKTRONIK
UNIVERSITI SAINS MALAYSIA (KAMPUS KEJURUTERAAN)
14300 NIBONG TEBAL,PULAU PINANG
NO. TEL: 04-5996004 NO. FAKS: 04-5941023

RECEIVED
25 JUL 2013
UNIVERSITI SAINS MALAYSIA
RCMO

Tarikh: 23 Julai 2013

Kepada:

Encik Mohd. Izhar Shuib
Penolong Pegawai Penyelidik
Unit Dasar, Strategi, Pemantauan & Pembangunan Penyelidikan
Bahagian Penyelidikan & Inovasi
Aras 6, Bangunan Canselori
Universiti Sains Malaysia
11800 Pulau Pinang
No. Tel: 04-653 6012/4354
No. Faks: 04-6568470

Daripada:

Norashikin Ismail,
Setiausaha
b.p Prof. Madya Dr. Nor Ashidi Mat Isa
Timbalan Dekan (Penyelidikan)

Nota Pesanan:

Tuan,

Bersama – sama ini dimajukan Laporan Akhir Fundamental Research Grant Scheme (FRGS) bertajuk seperti di bawah untuk makluman dan tindakan pihak tuan selanjutnya;

"Investigation on the Usefulness of Bioinformatics Application Techniques for Arabic Character Recognition"
Prof. Madya Dr. Umi Kalthum Ngah

Pr. Lee,
- mohon komen & ulasan utk penutupan geran ini.

Sekian, terima kasih.

-yavd-
30/7/13.

s.k Fail