OFF-LINE HANDWRITTEN JAWI CHARACTER SEGMENTATION USING HISTOGRAM NORMALIZATION AND SLIDING WINDOW APPROACH FOR HARDWARE IMPLEMENTATION

***Zaidi Razak*[1], *Khansa Zulkiflee*[2], *Noorzaily Mohamed Noor* [3], *Rosli Salleh* [4], *Mashkuri Yaacob*[5]**

Faculty of Computer Science and Information Technology,
University of Malaya, 50603 Kuala Lumpur, Malaysia.
Email: zaidi@um.edu.my[1], kulchazul@yahoo.com[2],
zaily@um.edu.my[3], rosli_salleh@um.edu.my [4], mashkuri@um.edu.my [5]

ABSTRACT

The task of segmenting text into characters is a necessary preprocessing step in the development of most character recognition systems because incorrectly segmented characters are likely to be incorrectly recognized. The segmentation of off-line handwritten Jawi text poses a higher challenge due to its cursive nature and various writing styles. In this paper, histogram normalization and sliding windows are used for hardware implementation of real-time off-line handwritten Jawi script character segmentation. Existing algorithms for character segmentation are compared with the proposed method. The hardware design is presented along with justifications of the proposed approach. The main advantage of the proposed algorithm is its simple design which enables it to be implemented in hardware without requiring a large amount of resources. The character segmentation algorithm was implemented and the results show a 98% segmentation accuracy.

Keywords: Optical Character Recognition (OCR), character segmentation, cursive script, image processing, handwriting recognition.

1.0 INTRODUCTION

In this paper we used histogram normalization and sliding window technique for the hardware implementation of Jawi text character segmentation. The projection histogram technique is used in character and word recognition for horizontal and vertical segmentation [1]. The vertical projection histogram method segments the document vertically by detecting the space between each character. This method also locates vertical strokes in printed documents or any regions of various lines in handwriting [2]. As reviewed by A. M. Zeki et al. in [3], the projection histogram analysis of text lines has been used as a basic method to segment non-recursive writing. It is less suitable for slanted characters which normally occur in handwritten documents.

Several methods have been proposed to segment Arabic characters especially using vertical projection histogram. As reviewed in Gouda et al. [4], each word is segmented into basic

characters based on the baseline by using a vertical histogram. Syiam et al. [5] implemented a clustering technique (k-means algorithm) on the vertical histogram. This improved the performance of histogram technique with recognition of handwritten characters. The character is clustered to identify the similarities of the characters. Romeo-Pakker et al. [6] proposed two methods in segmenting the Arabic character. The first method detects the junction in each connected character and the second method is detects the upper contour of each word. These methods are also implemented in [7] and [8].

The sliding window technique is a common localization technique in signal processing domain [9]. Since the character string runs in certain direction, a movable window called sliding window following the same order can be used to draw an interested zone and then extract features within it. Generally, the height of the sliding window is the same as that of text line. The sliding window width and the shift step are assigned by researchers or determined through experiments.

Bushofa and Spann [10] used sliding windows to find an angle of joined character that is formed between the joined characters. Although this method achieved good results, its success rate in finding the correct angle is influenced by the noise in the image. As an alternative, they used a more reliable segmentation algorithm compared to the histogram method. The correct position of segmentation is selected and indicated based on the angle that is formed between each pair of the joined characters.

V. Märgner et al. [11] proposed a method based on detecting the character baseline. The baseline estimation is implemented in the feature abstraction module. A sliding window is used to collect the character features.

In the following section of this paper, a description of the proposed system is presented in Section 2. Section 3 justifies why the approach is applied. The hardware design is described in Section 4. The experimental results are presented in Section 5. The conclusion and future work are presented in Section 6.

2.0 CHARACTER SEGMENTATION ALGORITHM

The proposed character segmentation algorithm consists of histogram generation, histogram gradient sign normalization and character segmentation. The character segmentation flow chart diagram for our algorithm is shown in Figure 1.

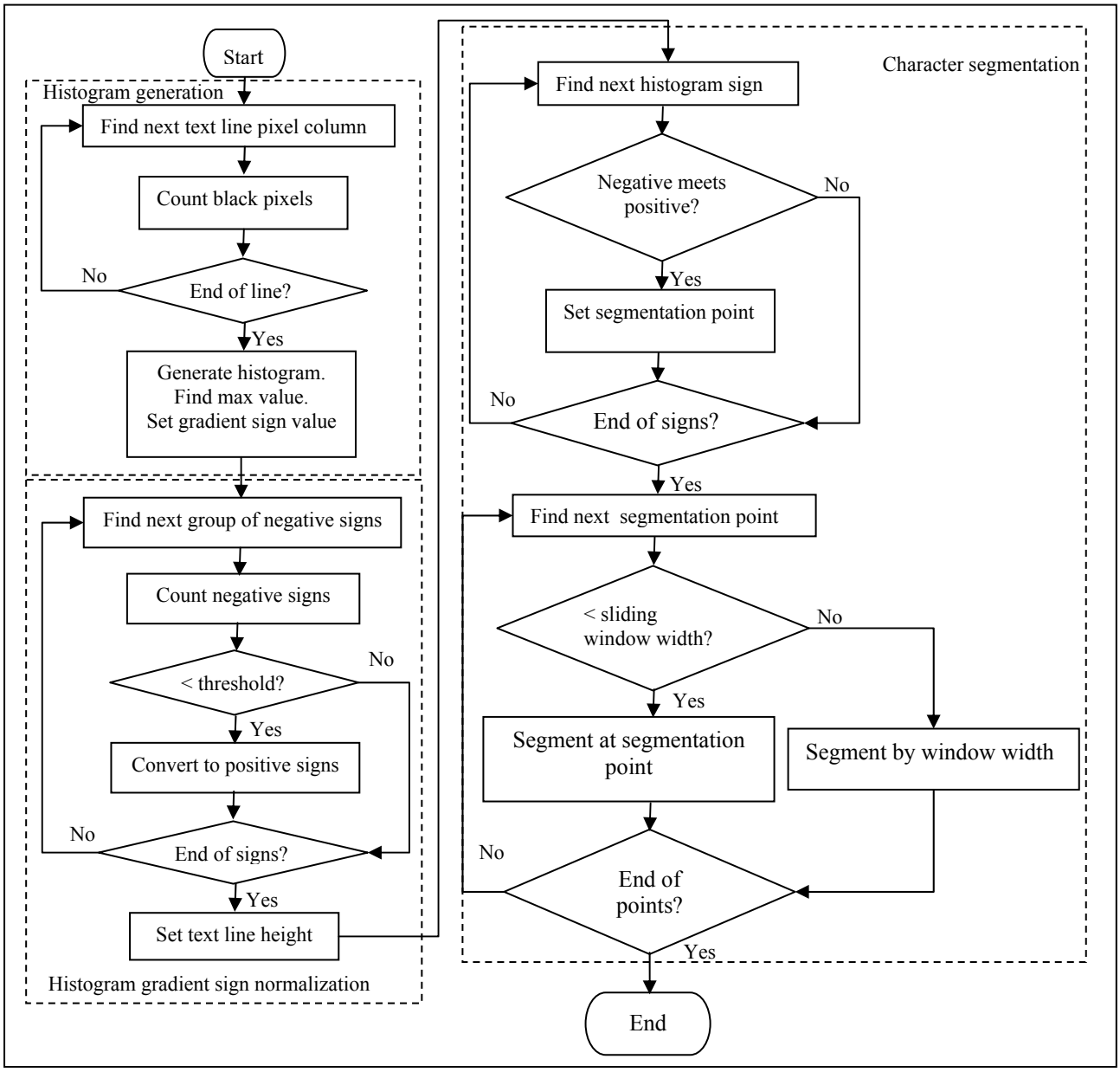


Fig. 1: Character segmentation flow diagram

2.1 Histogram generation

Line segmentation is performed using a tangent value to find the separation point (SP) for accurate line segmentation without data loss [12]. The tangent from the graph is calculated to find other representations (i.e. tangent representation of either 1 or negative 1). False local minima that exist in the graph are eliminated by collecting the number of 0s (black pixels) in corresponding rows. These values will be stored in one array as temporary storage.

After performing line segmentation, the document image is divided into separate text lines. Then, a histogram representing the number of black pixels in each column is generated for every text line. The maximum value in the histogram is used to set the threshold. Positive and negative signs are assigned to the histogram gradient values.

2.2 Histogram gradient sign normalization

Consecutive negative sign values in the histogram gradient graph for each group of consecutive negative sign values are counted. If the number of negative sign values is less than the threshold, all negative sign values in the group are converted to positive.

2.3 Character segmentation

The text line height is assumed as the sliding window width. In the histogram, the pixel counts where the negative sign gradients meet the positive sign gradients are set as the character segmentation points. If the length between neighboring segmentation points is less than the sliding window width then the particular segmentation points are used for segmentation. Otherwise, if the length between neighboring segmentation points is more than the sliding window width then the sliding window length is used for segmentation.

2.4 Pseudo code

The pseudo code below represents the character segmentation process performed based on our character segmentation algorithm which includes histogram generation, histogram gradient sign normalization and character segmentation.

1.0 For each pixel column in the text line
1.1 Count the number of black pixels
End For
2.0 Generate histogram representing the number of black pixels in each column
3.0 Find the maximum value in the histogram
4.0 Set the threshold using the maximum value in the histogram
5.0 Assign positive and negative signs to the histogram gradient values.
6.0 For each group of consecutive negative sign values in the histogram
6.1 Count the negative sign values
6.2 If the number of negative values is less than the threshold
6.2.1 Convert all the negative sign values to positive
End If
End For
7.0 Set the text line height as the sliding window width
8.0 For each histogram value
8.1 If negative sign gradients meets positive sign gradients
8.1.1 Set pixel count as character segmentation point
End If
End For

9.0 For each character segmentation point
9.1 Find next segmentation point
9.2 If length between the segmentation points is less than the sliding window width
9.2.1 Segment based on the segmentation points
End If
9.3 Else
9.3.1 Segment based on the sliding window width
End Else
End For

3.0 JUSTIFICATION OF APPROACH PROPOSED

In our approach, pre-processing such as edge detection, contour tracing, and Artificial Neural Network (ANN) methods are not performed since 100% accuracy is not crucial in our approach. Therefore, less processing time is required. For feature extraction, a sliding window is used to find the pixel distribution for feature extraction. Our approach is focused on analyzing rather than pre-processing and post-processing.

4.0 HARDWARE DESIGN

The character segmentation method described will be implemented in real-time hardware. The character segmentation hardware design consists of five entities which are the Histogram, Normalize Histogram, Determination of Segmentation Point (DSP), Control Unit and Segmentation entities as shown in Figure 2.

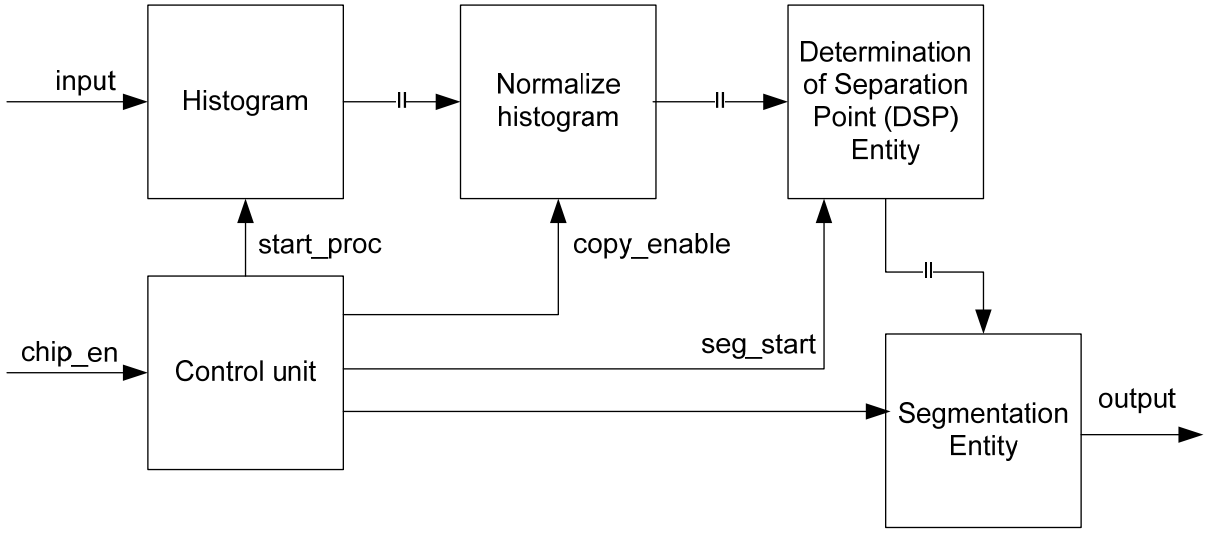
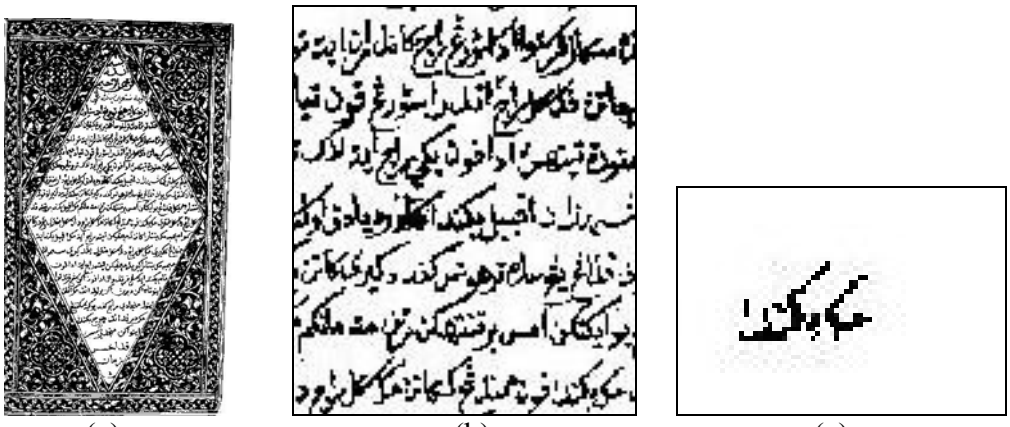


Fig. 2: Hardware design

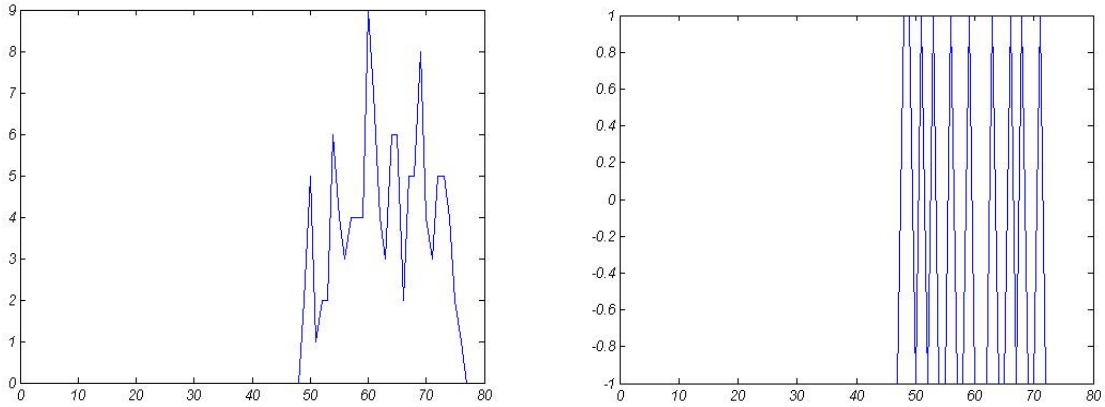
The Histogram entity receives a document image text line from the line segmentation process. The histogram gradient values and threshold value are sent to the Normalize Histogram entity. The Normalize Histogram entity sends normalized histogram gradient values to the DSP entity. Then, the DSP entity sends segmentation points to the Segmentation entity for character segmentation. The Control Unit entity controls all the process sequence and data manipulation.

5.0 EXPERIMENTAL RESULTS

Our fast algorithm compares favorably and shows highly accurate character segmentation although there are a few cut-off characters. These can be eliminated through a local analysis of each line. We envisage the on-chip implementation of the algorithm will also mirror similar fast processing performance based on the simplicity of our approach. In our experiments, we used Jawi manuscripts with high noise. Character segmentation results using our approach are shown in Figure 3.



(a) (b) (c)



(d) (e)

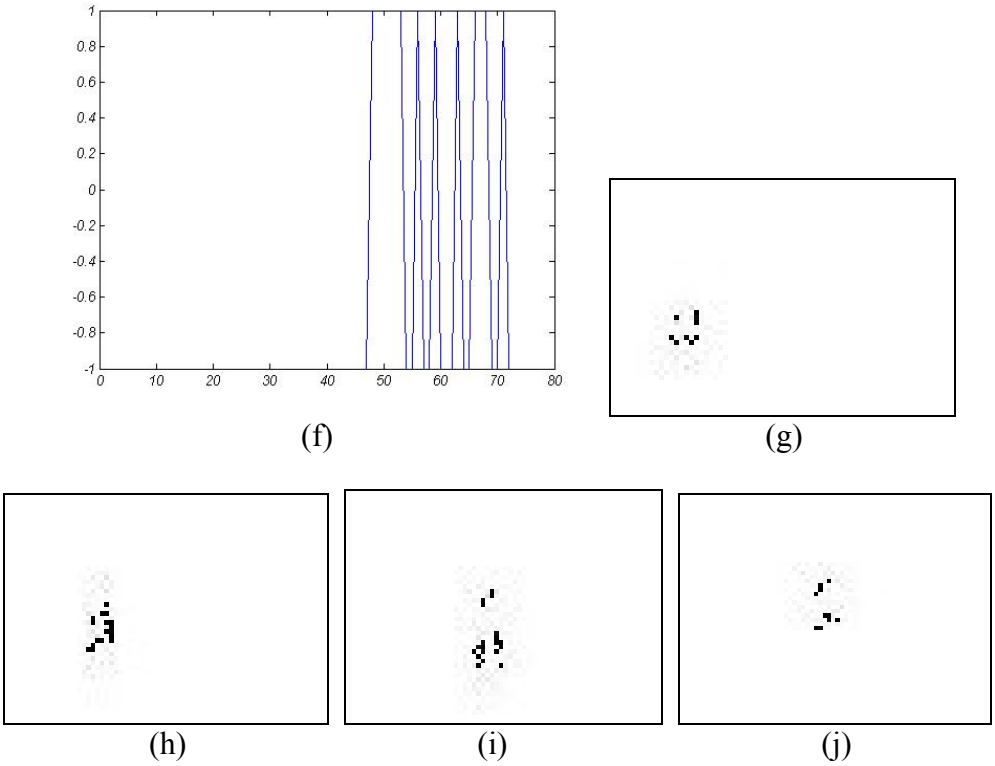


Fig. 3: Character segmentation results. (a) Original Jawi manuscript image, (b) Binary Region Of Interest (ROI), (c) Segmented word, (d) Black pixel histogram, (e) Histogram gradient graph, (f) Normalized histogram gradient sign, (g) – (j) Segmented characters

Inaccuracies occurred due to character overlapping which is one of the Jawi characteristics. Character overlapping will affect the black pixel distribution histogram and cause segmentation errors. Benchmarking results for our approach and previously proposed approaches are shown in Table 1.

Table 1: Benchmarking results

Authors	**Method**	**Experiment Data**	**Accuracy**	**Segmentation Errors**
Syiam et al. [5]	K-mean clustering algorithm (clustering technique) and applied to vertical histogram	Arabic handwriting	90% segmentation correctness achieved. The recognition rate is 91.5%	The grouping of two characters at the end of the sub-word. Produce over-segmentation for س or لا
Romeo-Pakker et al. [6]	Divide the segmentation process into two methods (detection junction of characters and segment based on the upper contour)	Arabic/ Latin handwriting	The character junction detection method achieved 93.5% of good segmentation. The upper contour segmentation method achieved 99.3% of good segmentation	In the character junction detection method, 2% of the characters were not segmented. 4.5% of characters were segmented unnecessarily. In the upper contour segmentation method, 0.3% of the characters were not segmented. 0.4% of characters were segmented unnecessarily.
Our approach	Sliding window and tangent normalization	"Hikayat Hang Tuah" Jawi manuscript from the 17th century	98% accuracy.	Inaccuracies occur due to character overlapping

The benchmarking was performed in order to evaluate the performance of our approach compared to the other approaches for segmenting Arabic handwriting. Evaluation was conducted based on the accuracy and segmentation errors. Accuracy is calculated based on the percentage of correctly segmented characters in its whole form. Segmentation errors are calculated based on the percentage of incorrectly segmented characters.

6.0 CONCLUSION AND FUTURE WORK

We have proposed a Jawi character segmentation algorithm using sliding window and tangent normalization. A 98% of segmentation accuracy rate for scanned "Hikayat Hang Tuah" manuscript from the 17th century was achieved. This percentage is based on the ROI. We were

unable to get perfect segmentation because of overlapping characters. In the future, we will analyze horizontal histogram for each character segment and also for the secondary entities. The application will be expanded to Field-Programmable Gate Array (FPGA) design.

REFERENCES

[1] R. G. Casey and Eric Lecolinet, "A Survey of Methods and Strategies in Character Segmentation", in *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Jul. 1996, Vol. 18, No. 7, pp. 690-706.

[2] U. Pal and Sagarika Datta, "Segmentation of Bangla Unconstrained Handwritten Text", in *Proceedings of the Seventh International Conference on Document Analysis and Recognition* (ICDAR'03) IEEE, 3-6 Aug. 2003, pp. 1128-1132.

[3] A. M. Zeki, "The Segmentation Problem in Arabic Character Recognition The State Of The Art", in *the First International Conference on Information and Communication Technologies*, ICICT, 27-28 Aug. 2005, pp. 11-26.

[4] A.M. Gouda, M.A. Rashwan, "Segmentation of Connected Arabic Characters Using Hidden Markov Models", in *the International Conference on Computational Intelligence for Measurement Systems and Applications* (CIMSA), 14-16 Jul. 2004, pp. 115-119.

[5] M. Syiam, T. M. Nazmy, A. E. Fahmy, H. Fathi, K. Ali, "Histogram Clustering and Hybrid Classifier for Handwritten Arabic Characters Recognition", in *Proceedings of the IASTED International Conference on Signal Processing, Pattern Recognition, and Applications*, SPPRA 2006, Innsbruck, Austria, 15-17 Feb. 2006, pp. 44-49.

[6] K. Romeo-Pakker, H. Miled, Y. Lecourtier, "A new approach for Latin/Arabic character segmentation", in *IEEE Proc. ICDAR'95*, Montreal, Canada, 14-16 Aug. 1995, pp. 874–877.

[7] A. Zahour, B. Taconet, P. Mercy, S. Ramdane, "Arabic Hand-written Text-line Extraction", in *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, ICDAR'01, 10-13 Sept. 2001, pp. 281 – 285.

[8] M. Omidyeganeh, K. Nayebi, R. Azmi and A. Javadtalab, "A New Segmentation Technique for Multi Font Farsi/Arabic Texts", in the *IEEE International Conference on Acoustics Speech and Signal Processing* (ICASSP'05), 18-23 Mar. 2005, pp. 757-760.

[9] T. Su, T. Zhang, D. Guan, and H. Huang, "Off-line recognition of realistic Chinese handwriting using segmentation-free strategy", in *Pattern Recognition*, Jan 2009, Vol. 42, No. 1, pp. 167-182.

[10] B.M.F. Bushofa and M. Spann, "Segmentation of Arabic Character Using Their Contour Information", in the *13th International Conference on Digital Signal Processing Proceeding* (DSP'97), Santorini, Greece, 2-4 Jul. 1997, Vol. 2, pp. 683-686.

[11] V. Märgner, M. Pechwitz, and H. El Abed, "Baseline Estimation For Arabic Handwritten Words", in *Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition* (IWFHR'02), 2002, pp. 479-484.

[12] Z. Razak, K. Zulkiflee, R. Salleh, M. Yaacob and E. M. Tamil, "A Real-Time Line Segmentation Algorithm for an Offline Overlapped Handwritten Jawi Character Recognition Chip", in the *Malaysian Journal of Computer Science*, 2007, Vol. 20, No. 2, pp. 69-80.

BIOGRAPHY

Zaidi Razak obtained his Bachelor's degree in Computer Science and Master in Chip Design from University of Malaya and is currently a lecturer at the Faculty of Computer Science and Information Technology, University of Malaya.

Khansa Zulkiflee obtained her Bachelor's degree in Computer Science from University of Technology Malaysia and is currently a research assistant at the Faculty of Computer Science and Information Technology, University of Malaya.

Noorzaily Mohamed Noor obtained his Bachelor's and Master degree in Computer Science from University of Malaya and is currently a lecturer at the Faculty of Computer Science and Information Technology, University of Malaya.

Rosli Salleh obtained his Bachelor's degree in Computer Science from University of Malaya and Master's degree in Data Communication Networking and PhD in Computer Science both from University of Salford Manchester United Kingdom. He has also obtained CCNA professional qualifications. Currently, he is a lecturer at the Faculty of Computer Science and Information Technology, University of Malaya.

Mashkuri Yaacob is currently the third vice-chancellor of Universiti Tenaga Nasional. Mashkuri, who obtained his Bachelor's degree in Electrical Engineering from the University of New South Wales, Sydney, also holds master's and doctoral degrees in electronics and computer engineering from the University of Manchester in the United Kingdom.