Contour vs Non-Contour based Word Segmentation from Handwritten Text Lines: an Experimental Analysis

Fajri Kurniawan*^{Corresponding author}, Amjad Rehman Khan, Dzulkifli Mohamad
Department of Computer Graphics and Multimedia, Universiti Teknologi Malaysia, Skudai,
Johor Bahru, 81310, Malaysia
fajri_9636@yahoo.com, amjadbzu2003@yahoo.com, dzulkifli@utm.my
doi: 10.4156/jdcta.vol3.issue2.kurniawan

Abstract

This paper compares contour based and non-contours based techniques for extracting words from unconstrained handwritten text lines. Proposed novel approach is based on contours of the words rather only considering threshold for inter-word gaps as previous studies. In this approach, contour of each word is examined along with threshold for inter-word gaps to extract words with high confidence. Unlike previous studies, preprocessing technique is not applied, that enhance the speed significantly. Furthermore, a simple technique for punctuation detection is proposed to increase accuracy of word extraction. For fair comparison text lines are taken randomly from IAM benchmark database and threshold calculation is kept same for all techniques. Experiments thus performed, exhibit improved results and speed over the conventional word extraction methods. Furthermore, developed techniques and results are compared with the other approaches available in the literature using same benchmark database.

Keywords

Word segmentation, preprocessing, contour detection, punctuation detection.

1. Introduction

The words segmentation approaches in printed/handwritten text lines are usually based on various heuristics and assumption that the gaps between words (inter-word gap) are larger than those inside the words (intra-word gap) [2-8]. Words extraction from handwritten text lines usually involves calculation of threshold for inter-word gap. Hence, the text line is decomposed into series of single connected or horizontally overlapped connected components [7-8]. Following decomposition, a threshold distance is determined to decide the gap either inter-word or intra-

word [5], finally the words are extracted. The extracted words may contain punctuation marks.

Additionally, there are recognition free approaches to determine word segment points such as neural networks [9], scale space techniques [10] and semantic knowledge [11]. On the hand, such approaches increase complexity and ultimately processing speed is reduced. Thus, research in this area is still active and onward towards its maturity [13].

Unlike the previous techniques for word extraction, proposed technique does not rely only on the horizontal distance threshold but to make the approach robust contour features are also activated. Hence, it allows inter-word gaps to be smaller than the threshold that increases its performance and robustness.

This paper exhibits improved results with simple techniques without any preprocessing for word extraction in contrast to Varga and Bunke [1]. In this regards, a brief review of [1] is presented.

Complex tree structure is proposed using extension of baseline method [7] for word extraction along with preprocessing (slant, skew correction and upper, lower, middle line detection). Besides that, punctuations were assumed on upside and downside of lower and upper baselines respectively. Hence only punctuations those outside core zone are detected. Four variants BB-MWR (bounding box median white run length), BB-AWR (bounding box average white length), CH-MWR (convex hull median white run length) and CH-AWR (convex hull average white length) were introduced to calculate threshold. Two sets of experiments were performed H_{dev} and H_{test} . H_{dev} set contained 7880 words in 1044 text lines by 120 writers to optimize variants and H_{test} consist of 1025 text lines. Finally, the word extraction accuracy reported 95.67%.

In contrast to [1], authors have introduced simple approach for word extraction and punctuation detection without any preprocessing. Experiments performed on 1000 text lines, taken randomly from IAM database [12], showing improvement on accuracy and speed. The rest of the paper is organized in three sections. Section 2 presents the proposed approach, in section 3

experimental results are discussed and finally conclusion is drawn in section 4.

2. Proposed Word Extraction Approach

In this section, authors demonstrate how to apply contour based approach for word extraction. Contour and punctuation detection are described and demonstrated. Furthermore, threshold computation is explained.

2.1 Contour Detection

A very simple technique is adopted to detect contour of the words in the text line. The contours are traced in the text line from right to left. The foreground pixel that have no foreground pixels on its right, top and bottom up to some threshold distance (both vertical and horizontal) are treated as contour (see figure 1). Let say an images denoted by P where $P \in \{0,1\}$,

$p_{i,j} \in P, \begin{cases} i=1,2,\dots,h \\ j=1,2,\dots,w \end{cases}$, h,w is height and width of P respectively.

1. Trace P where $p_{i,j} \in \begin{cases} i=h, h-1, \dots, 2, 1 \\ j=w, w-1, \dots, 2, 1 \end{cases}$
2. Define connected foreground pixel as
 $C = \{c \in \{p_{i,j}\} \mid p_{i,j} = 1, p_{i+1,j} = 1, i=1, 2, \dots, a, a \leq h\}$
3. Take $c_{a,b} \in C$, If $c_{a,b+1} = 1$ then move tracing to column left ($j = j-1$) and do step 2 else
 $candidate_contour_{i,j} = average(C)$
4. if $candidate_contour_{i,j+1}, (i \geq 1) = 1$ and
 $candidate_contour_{i,j+1}, (i \leq h) = 1$ move tracing to left column ($j = j-1$) and do step 2 else
 $contour = \{contour_coordinate_{i,j}\}$
5. move tracing to left column ($j = j-n$) and do step 2. $n = 10$ set heuristically.

Figure 1. Checking up and down of contour coordinate.

The decision of the word extraction is not determine on the basis of contours only but along with derived threshold otherwise it may mislead in unconstrained cursive handwriting. Contours traced of text line are shown in figure 2.

Figure 2. Traced contours marked by arrowhead.

2.2 Punctuation Detection

Punctuations can cause inter-words gap become smaller and therefore mislead intra-word gap to inter-word gap [8]. Hence, punctuation is a big problem and therefore it causes many errors in accurate word segmentation. Many preprocessing techniques are employed in this regard [1, 3, 4, 10, 11, 13]. Authors have proposed new technique to detect punctuations as well. A small size square is drawn by taking each detected corner as center. The contour is detected as punctuation, if square boundary does not cross any foreground pixel. The size of square is small as possible that calculated empirically. After that, the detected punctuations are treated separately and are not taken into consideration when segmenting the line. A possible weakness of the proposed punctuation detection technique is unable to detect punctuation that larger than normal size or too close to character's contour. The results for punctuation detection are exhibited in fig. 3.

Figure 3. Detected punctuations are enclosed in squares.

2.3 Threshold Computation

In this paper, threshold is computed from median white run-length (MWR) and Average white run length (AWR) in a line. MWR is the median of the set of white run-length in a line. Meanwhile, AWR is the median of the number of white pixels in a line divided by the median number of black-white transitions in the same line. Nevertheless, MWR is rather simple to classify most short gaps as intra-word gaps [13].

2.4 Bounding Block

Following contour and punctuation detection, word components are extracted based on contour and threshold value. Segmentation points are determined from left to right side. Segmentation of each component consists of two segmentation lines termed as bounding block (BB). Bounding blocks for each word component are shown in figure 4.

Figure 4. Components extracted in separate blocks.

First and second segmentation points of words are located interchangeable. First segmentation determine by tracing foreground pixel column by column from left to the right. First segmentation is the column with zero number of pixels. Second segmentation was determined using contour coordinate with check gap to the right column along with threshold value. This approach is successful even inter-word distance was minimum.

3. Experimental Results and Discussion

3.1 Text Lines Database

For experiments, 1000 text lines (with 8,610 words) are extracted randomly from IAM database [12] and fed to the proposed system. Figure 5 presents some example of text lines taken from IAM database. It is shown that there are different types of text lines with different styles, gaps and slants. In this regards, it can represent the real life problems.

Figure 5. Example of text lines.

3.2 Analysis and Comparison of Results

For measuring performance of a segmentation algorithm, there exist several methods [13]. In this paper word extraction rates (WER) is computed similar to [1, 3]. The accuracy of word extraction was measured as the percentage of correctly extracted words in relation to the number of words as under.

$$WER = \frac{\text{number_of_correctly_extracted_words}}{\text{number_of_all_words}} \quad (1)$$

The segmentation task is also considered as an inter-word gap finding task known as Gap accuracy (GA) calculated as under.

$$GA = \frac{\text{correctly_found_gaps} - \text{gaps_wrongly_found}}{\text{all_gaps}} \quad (2)$$

Words extracted are shown in figure 6 (i-iv). Speed is another important issue, overlooked by many researchers. Complexity of the technique is inversely proportional to speed [14]. As the proposed approach is simple and without preprocessing, hence the speed is optimal. Although our experiments are only preliminary, but the results compare favorably with those of other researchers. Performance of the novel method with other research reported in literature is compared in table 1.

Table 1. Word extraction accuracies (in %) of different methods.

Threshold calculation	Baseline method [7,10]	Tree-structure method [1]	Counter-based method		
			Threshold calculation	Accuracy	
MWR(BB)	91.48	92.90	MWR(CB)	96.35	96.17
AWR(BB)	92.78	93.25	AWR(CB)	95.81	95.63
MWR(CH)	93.51	94.23			
AWR(CH)	94.75	95.17			

BB: Bounding Box, CH: Convex Hull, CB: Contour Based

Table 1 exhibit that proposed approach has improved word segmentation accuracy over all four variants employing different threshold computational techniques [4, 5, 6, 8]. For the threshold type, MWR produced better results over AWR, however novel method brought the MWR and AWR results closer to each other. The results reported in table 1 are comparable to those of [1, 7, 10] since the performance was evaluated in a similar way as in this paper. Finally, we can conclude that the punctuation detection, despite of being calculated in a rather simple way, improved the word extraction accuracy significantly.

(i)	<p>movement of employed people is not practicable</p>	(ii)	<p>Any impurity in the lead fraction must have</p>
(iii)	<p>Tables C D and E of Appendix 16 give details of</p>	(iv)	<p>Tough they may gather some left wing support, a</p>

Figure 6. Extracted words, 1st (i), 2nd (ii) 3rd (iii) and 4th (iv) text lines.

4. Conclusion

A comparison based on contour vs. tree structure, baseline has been presented. The techniques are experimented and compared using IAM benchmark database. We have conducted a number of experiments by varying threshold calculation practiced by other researchers. The best comparison results with our

experiments are compared in table 1. The results show that contour based word segmentation achieved better word segmentation rate than other techniques. We must point out that our technique is free from all normalization that reduced computation complexity and therefore enhanced speed significantly unlike other techniques.

5. Acknowledgement

This research is fully supported by Ministry of Science and Technology Innovation (MOSTI).

6. References

- [1] T. Varga and H. Bunke, "Tree structure for word extraction from handwritten text lines", Proceedings of the eight international conference on document analysis and recognition (ICDAR'05), IEEE, 2005.
- [2] E. Kavallieratou, N. Fakotakis and G. Kokkinakis, "An unconstrained handwriting recognition system", Int. journal on Document Analysis and Recognition, Vol. 4, No. 4, 2002, pp. 226-242.
- [3] S. Kim, C. Jeong, H. Kwag and C. Suen, "Word segmentation of printed text lines based on gap clustering and special symbol detection", In Proc. 16th Int. Conf. on Pattern Recognition. Quebec city, Canada, 2002, pp. 320-323.
- [4] S. Kim, G.S. Lee and C. Suen, "Word segmentation in handwritten Korean Text lines based on gap clustering techniques", In proc 6th International conference on document analysis and recognition. Seattle, WA, USA, 2001, pp. 189-193.
- [5] U. Mahadevan and R. Nagabushnam, "Gap metrics for word separation in handwritten lines", In Proc. 3rd Int. Conf. on Document Analysis and Recognition, Montreal, Canada, 1995, pp. 124-127.
- [6] F. Luthy, T. Varga and H. Bunke, "Using Hidden Markov Models as a Tool for Handwritten Text Line Segmentation", Ninth International conference on Document analysis and Recognition, 2007.
- [7] U.V. Marti and H. Bunke, "Text line segmentation and word recognition in a system for general writer independent handwriting recognition", In Proc. 6th Int. Conf. on Document Analysis and Recognition, Seattle, WA, USA, 2001, pp. 159-163.
- [8] G. Seni and E. Cohen, "External word segmentation of offline handwritten text lines", Pattern Recognition, Vol. 27, No. 1, 1994, pp. 41-52.
- [9] G. Kim and V. Govindaraju, "Handwritten phrase recognition as applied to street name images", Pattern Recognition, Vol. 31, No. 1, 1998, pp. 41-51.
- [10] R. Manmatha and N. Srima, "Scale space technique for word segmentation in handwritten documents", Scale-Space Theories in Computer Vision, 1999, pp. 22-33.

- [11] M. Feldbach and K. Tonnie, "Word segmentation of handwritten dates in historical documents by combining semantic a-priori-knowledge with local features", In Proc. 7th Int. Conf. on Document Analysis and Recognition, Edinburgh, Scotland, 2003, pp. 333-337.
- [12] U. Marti and H. Bunke, "The IAM database: An English sentence database for off-line handwriting recognition", International Journal of Document Analysis and Recognition, Vol. 15, 2002, pp. 65-90.
- [13] M. Liwicki, M. Scherz and H. Bunke, "Word extraction from on-line handwritten text lines", The 18th international conference on pattern recognition (ICPR'06), IEEE, 2006.
- [14] R.G. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 7, 1996, pp. 690-706.