http://dx.doi.org/10.5392/IJoC.2014.10.3.035

Text Line Segmentation using AHTC and Watershed Algorithm for Handwritten Document Images

KangHan Oh, SooHyung Kim*, InSeop Na, GwangBok Kim
School of Electronics and Computer Engineering
Chonnam National University, 77 Yongbong-ro, Buk-gu, Gwangju 500-757, Korea

ABSTRACT

Text line segmentation is a critical task in handwritten document recognition. In this paper, we propose a novel text-line-segmentation method using baseline estimation and watershed. The baseline-detection algorithm estimates the baseline using Adaptive Head-Tail Connection (AHTC) on the document. Then, the watershed method segments the line region using the baseline-detection result. Finally, the text lines are separated by watershed result and a post-processing algorithm defines the lines more correctly. The scheme successfully segments text lines with 97% accuracy from the handwritten document images in the ICDAR database.

***Key words**: Watershed, Text Line Segmentation, Handwritten Document*

1. INTRODUCTION

Text line detection is a necessary step in unconstrained handwritten document recognition. In optical character recognition field, text line and word segmentation is a significant step and character recognition strongly depends on accuracy of text line segmentation. For example, incorrect line segmentation leads to incorrect character recognition. Text line segmentation in handwritten document images is a challenging job for handwritten document analysis and character recognition. It is difficult mainly because there are several regions 1) character shapes originated from the variability of writing styles 2) variation of skew and distance 3) overlapping or touching lines 4) variable character size. These problems can be seen in Fig. 1.

In this paper, we propose a text line segmentation method using watershed and AHTC in the handwritten document images. The AHTC algorithm extracts initial text baseline and then the watershed algorithm segments text line regions based on estimated baseline computed by AHTC algorithm. From segmented text line region, whole units which are pixels or connected components are assigned to the nearest label.

The organization of the rest of the paper is as follows: in section 2, we refer to related work, in section 3: we describe in detail the proposed method for text line segmentation on the handwritten document images. The experimental results are presented in section 4 and conclusions are discussed in section 5.

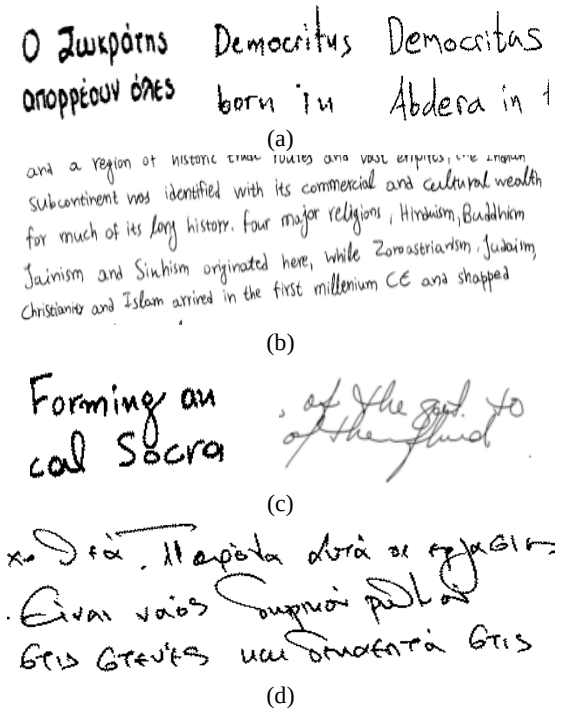


Fig. 1. Text line segmentation challenges. (a) Variability of writing styles, (b) variation of skew and distance, (c) overlapping or touched lines, (d) variable character size

* *Corresponding author, Email: shkim@chonnam.ac.kr*
Manuscript received May. 15, 2014; revised Aug. 22, 2014; accepted Aug. 29, 2014

2. RELATED WORKS

In this section, we give a brief review of related work about text line segmentation in handwritten document images. In the past few decades lots of text segmentation methods have been proposed in order to efficiently segment text line in the handwritten document images. In this field several other text line segmentation approaches which are different paradigms can be classified as projection based, run-length smearing, and grouping [1].

The projection profile analysis was applied in [2], [3] to segment the boundaries of the text lines. This method a histogram crossing an entire text block along a predetermined direction of the text lines is created. Then valleys that represent interline gaps are located to segment the text lines. In the projection profile fields, they are effective when document has enough gaps between text lines and horizontally aligned text lines. But it cannot segment text line with different skew angles which often appear in the handwritten documents. An author [4] divide image into vertical stripes and profile projection on each strip in order to overcome the skew problem.

The Run-length smearing [5] method is usually used for tolerating noise and run-away black strokes. The expected result from smearing process is that the most of foreground (text characters) are grouped together. And then the text lines and text blobs are classified by connected component algorithm. The method works well for printed documents with mostly text. But it cannot segment touching lines or connection between text lines and text blobs. The grouping approaches can segment complex layouts.

The reference [7] proposed a text line segmentation method using perceptual grouping algorithm. In this method, the text lines are segmented by grouping neighboring connected components using perceptual criteria (similarity, continuity, proximity). But this method cannot segment text line when the alignment contains anchors of different directions. The authors in [8] proposed a text line segmentation method using the local minima detection of connected components and a chain code representation. This method gradually segments line until unique text line is formed. The grouping method can handle text lines closed to each other, touching text.

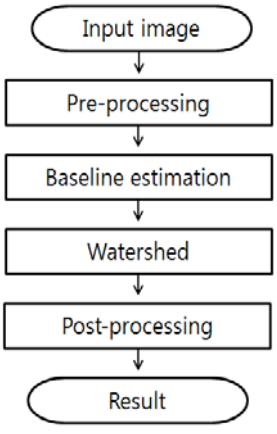
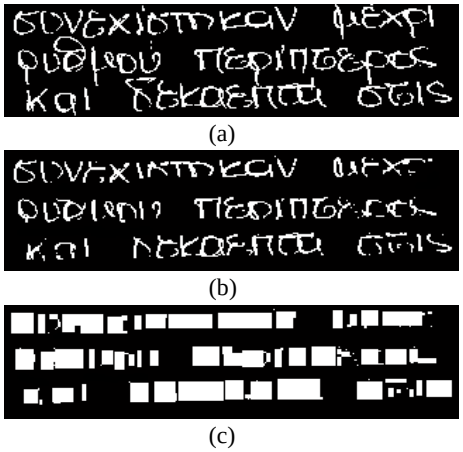


Fig. 2. The flowchart of proposed method

3. RROPOSED METHOD

A flowchart of the proposed method is described in Fig. 2. First, preprocessing part makes input document image suitable for efficiently segmenting baseline using several methods (connected component, adaptive local connectivity map). And, in the Baseline estimation step, we estimate baseline using AHTC algorithm. And then the watershed detects line region using initial markers which are estimated baseline from previous step. Finally, the text strings are assigned to the appropriate line region.



(a)

(b)

(c)

Fig. 3. Preprocessing process (a) input image (b) ALCM result with Threshold (c) connected component result with Rectangle

3.1 Preprocessing

In this section, the main goal is to make text by set of sparse blobs because the proposed method belongs to the grouping method field. We use the adaptive local connectivity Map (ALCM) [9] on the binary image in order to remove pixels which are located in gap between lines. The ALCM is defined as a transform

$$ALCM : f \rightarrow A \tag{1}$$

By a one-direction convolution

$$A(x,y) = \int_R f(x,y) G_c(t-x,y)dt \tag{2}$$

Where

$$G_c(x,y) = \begin{cases} 1 & if\ |x| < c \\ 0 & otherwise \end{cases} \tag{3}$$

Where R represent given data, we first reverse the input binary image therefore we consider only white pixel for computing ALCM. And we scan the image from left to right using sliding window of size c, to compute the cumulative intensity at a pixel by adding up all the intensity values in a neighborhood of size c. And then pixels which have small intensity value are removed by threshold value (average). Fig. 3 (b) shows ALCM result with threshold. Finally, the rectangles

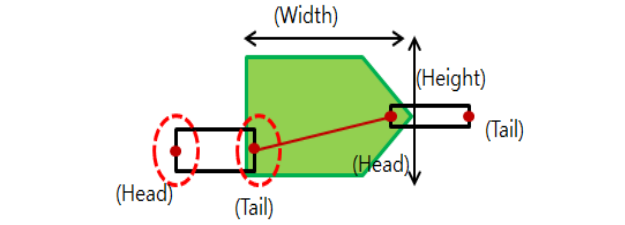
are created around each character using connected component (CC) algorithm.

In the preprocessing, sometimes big rectangle is created from touched two text which are belonged in different lines. In order to separate big rectangle blob, we compute an average of rectangle height in whole rectangle blob and we find the rectangle blob which bigger than average height and then system divides the rectangle blob. Fig. 4 shows this process

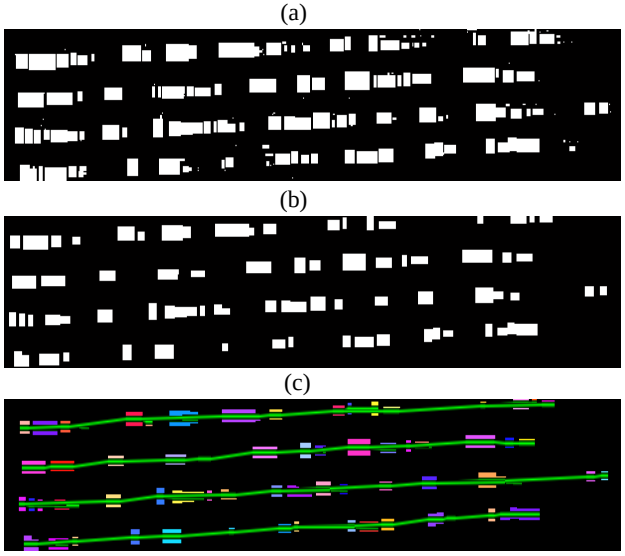
(a)

(b)

Fig. 4. Process of dividing big blob Rectangle (a) input (b) divided big blob



(a)



(b)

(c)

(d)

Fig. 5. AHTC algorithm (a) the structure of AHTC algorithm (b) input (c) after applying erosion (d) estimated base text lines

3.2 Baseline extraction using AHTC

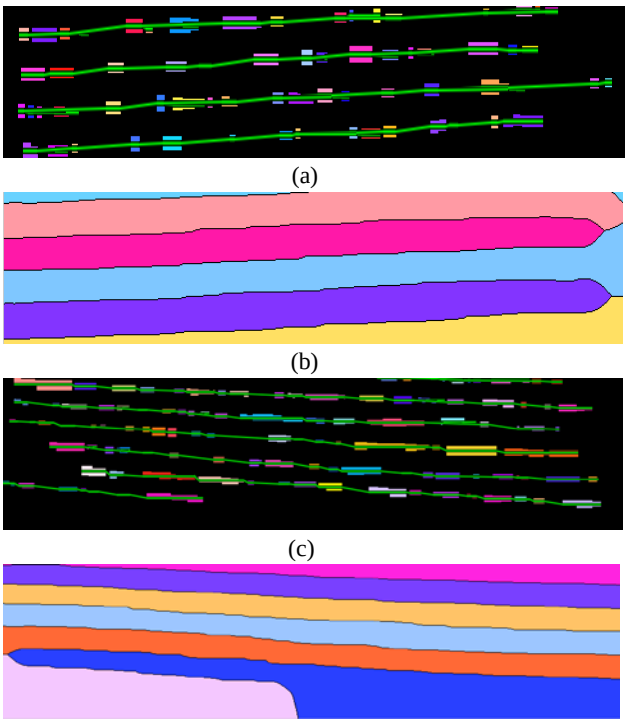
The aim of this stage is to define baseline of text string in the input image. In order to segment baseline, we proposed the Adaptive Head-Tail connection (AHTC) algorithm. This model contains 4 steps, they are:

Step 1: The erosion operation is then applied to remove the small blobs between the text lines.

Step 2: we defined Head and Tail points on each rectangle blob.

Step 3: in order to generate group of text strings, a forward searching area is defined based on average of blob size such as Figure 5(a) (green region). (Optimal searching area is mentioned in experimental results part)

Step 4: we scan document from right to left. Whole of the rectangle blob scans forward region and finds nearest Tail points using the Euclidean distance measure. And then proposed algorithm connects between Head point and Tail point. Figure 5 shows this process



(a)

(b)

(c)

(d)

Fig. 6. The watershed segmentation. (a), (c) input baseline (b), (d) result of textline region

3.3 Watershed Segmentation

In this stage, the text line region is extracted from estimated baseline result using Watershed algorithm. The watershed algorithm is a well-known image segmentation approach [10]. In geography, watershed means the ridge that divides areas drained by different river systems. If image is viewed as geological landscape, the watershed lines determine

boundaries which separate image regions. The watershed transform computes catchment basins and ridgelines, where catchment basins corresponding to image regions and ridgelines relating to region boundaries [10]. Fig. 7 illustrates watershed line and catchment basin.

In this paper, we use watershed algorithm to detect text line region based on estimated baseline in the previous process. The estimated baselines are utilized as initial markers in the watershed processing. Figure 6 shows the result of segmented text line region.

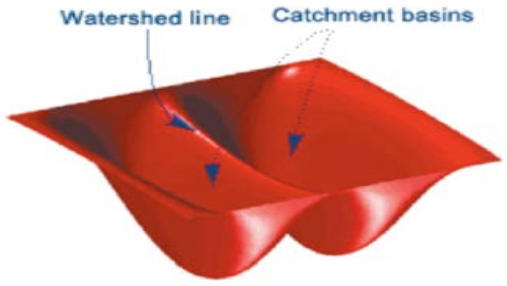
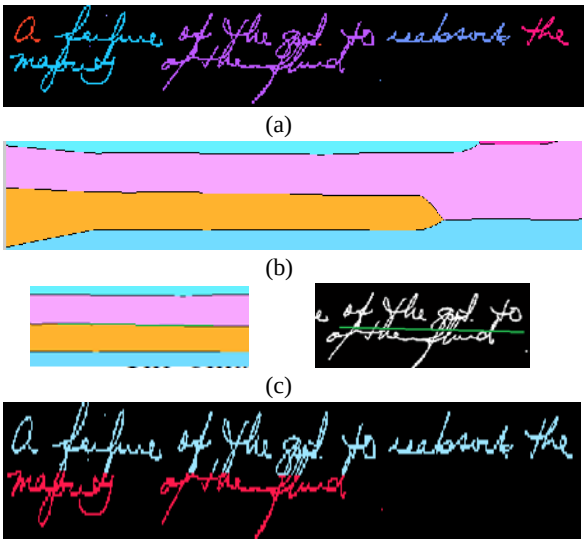


Fig. 7. Watershed lines and catchment basin [17]



(a)

(b)

(c)

(d)

Fig. 8. Splitting characters crossing multiple line. (a) multi line character (b) result of watershed (c) segmentation of the contours with watershed result (d) separated result

3.4 Text line segmentation and Post-processing

In this step, the aim is to assign each connected component of text to lines using segmented text line region. From binary image, connected component are created as sets of connected text pixels. And then we compute overlap score between connected component of text and segmented text line region (Fig. 6 (d)). And then each connected component of text is assigned to text line region which has high overlap score. The overlap score OS is given by the following equations:

$$\mathrm{OS} = \frac{text \cap segmented\ text\ line\ region}{text} \qquad (4)$$

Some connected components may belong to more than one line pattern. These components represent characters that cross multiple text lines such as Fig. 8(a). Although these crossing connected components can be easily detected, it is not easy task to separate them. To segment the touching connected component, if overlap score OS result is smaller than 0.7, our algorithm considers it as an ambiguous result therefore we use boundary line between text lines from watershed result. Fig. 8(c) shows this process.

4. EXPERIMENTAL RESULTS

We implemented the proposed algorithm using MATLAB 2012 on a Intel(R) Core(TM)2 Quad CPU Q9550 to verify it's performance. The result of proposed method is evaluated on public dataset (ICDAR 2009 [13]). We evaluated the performance using equation (5). The test dataset (100 documents) which has corresponding ground truth. The performance of proposed method is evaluated with 100 images. The evaluator's acceptance threshold is 95%.The performance evaluation is based on counting the number of matches between the segmented text line by proposed algorithm and the ground truth. The detection rate (DR), recognition accuracy (RA), and F-measure (FM) are given by the following equations:

$$\mathrm{DR} = \frac{o2o}{N},\ \ \mathrm{RA} = \frac{o2o}{M},\ FM = \frac{2DR\ RA}{DR+RA} \qquad (5)$$

Table 1. Comparison of the text line segmentation

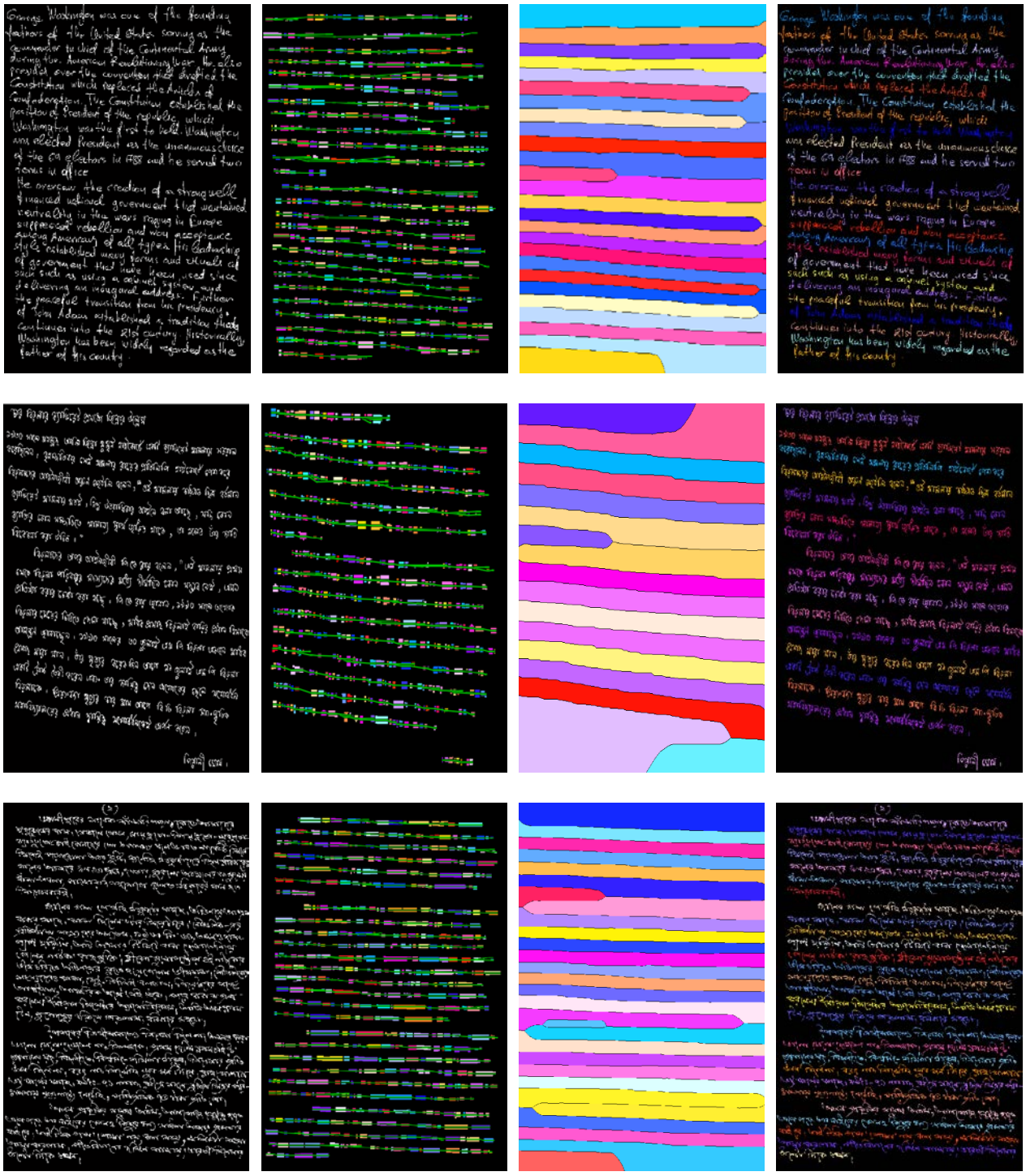
Height / Width	*5*	*7*	*13*	*15*	*30*
$\frac{width}{4}$	93.18	96.02	96.12	95.95	93.25
$\frac{width}{8}$	94.58	96.62	**96.85**	96.85	94.95
$\frac{width}{20}$	91.28	92.22	93.94	90.65	89.35

Table 2. Segmentation results for various searching area

	M	*o2o*	*DR (%)*	*RA (%)*	*FM (%)*
CUBS[14]	1626	1589	97.54	97.72	97.63
TEI[15]	1637	1549	95.09	94.62	94.86
IRISA[16]	1626	1578	96.87	96.45	96.66
ILSP [12]	1655	1559	95.70	94.20	94.95
The proposed method	1634	1573	96.56	96.27	96.85

Table 1 shows a comparison between the proposed method and the several competitive methods of text line segmentation in ICDAR 2010 Handwritten Segmentation Contest [13]. The proposed method has good performance compared to the other state-of-the-art text line segmentation method. The best method is CUBS with 97% accuracy however it is so heuristic for separating touched text between two lines whereas the proposed method has simple process using watershed result. Fig. 9 shows some of the text line

segmentation results. Table 1 represents FM result from various searching rage values. The best searching ranges are 13 height and width/8.



(a) (b) (c) (d)

Fig. 9. Text line segmentation results. (a) Original image (b) Baseline estimation using AHTC (c) Text region segmentation using watershed (d) Text line segmentation results.

5. CONCLUSION

In this paper, we segments text line in the handwritten documents using AHTC and watershed. The efficiency of proposed method increases performance of segmenting text line. Although the results are encouraging, future works are required to segment text line area in more complex scenes and the AHTC algorithm process should be accurate.

ACKNOWLEDGMENT

This research was supported by DIOTEK Co., Ltd. under the R&D program. And this research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2014-014400).

REFERENCES

[1] L. Likforman-Sulem, A. Zahour, and B. Taconet, "Text line segmentation of historical documents: a survey," Int. J. Doc. Anal. Recognit, vol. 9, no. 2, 2007, pp. 123-138.
[2] M. Bulacu, R. Koert, L. Schomaker, and T. Zant, "Layout an alysis of handwritten historical documents for searching the archive of the Cabinet of the Dutch Queen," in: Proceeding ICDAR, 2007, pp. 357-361.
[3] Y. G. Ciardiello, G. Scafuro, M. T. Degrandi, M. R. Spada, and M. P. Roccotelli, "An experimental system for office document handling and text recognition," Proc 9th Int. Conf. on Pattern Recognition, 1988, pp. 739-743.
[4] A. Zahour, B. Taconet, P. Mercy, and S. Ramdane, "Arabic hand-written text-line extraction," Proc. ICDAR, 2001, pp. 281-285.
[5] L. O'Gorman, "The document spectrum for page layout analysis," IEEE Trans. Pattern Anal. Mach. Intel., vol. 15, no. 11, 1993, pp. 1162-1173.
[6] L. Gorman, "The document spectrum for page lay-out analysis," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, no. 11, 1993, pp. 1162-1173.
[7] M. Feldbach and K. D. Tonnies, "Line detection and segmentation in historical church registers," Proc. ICDAR, 2001, pp. 743-747.
[8] L. Likforman-Sulem and C. Faure, "Extracting text lines in handwritten documents by perceptual grouping," Proc. Advances in handwriting and drawing: a multidisciplinary approach, 1994, pp. 117-135.
[9] L. O'Gorman, "The document spectrum for page layout analysis," IEEE Trans. Pattern Anal. Mach. Intell., vol. 15, no. 11,1993, pp. 1162-1173.
[10] Manisha Bhagwat, R. K. Krishna, and Vivek Pise, "Watershed Trans-formation," International Journal of Computer Science and Communication, vol. 1, no. 1, 2010, pp. 175-177.
[11] P. Soille, *Morphological Image Analysis*, 2nd ed., New York: Springer, 2002.
[12] L. O'Gorman, "The document spectrum for page layout analysis," IEEE Trans. Pattern Anal. Mach. Intell.,vol. 15, no. 11,1993, pp. 1162-1173.
[13] B. Gatos, "ICFHR 2010 Handwriting Segmentation Contest," in ICFHR, 2010, pp. 737-742.
[14] Z. Shi, S. Setlur, and V. Govindaraju, "A Steerable Directional Local Profile Technique for Extraction of Handwritten Arabic Text Lines," Proc. 10th International Conference on Document Analysis and Recognition (ICDAR'09), 2009, pp. 176-180.
[15] A. Nicolaou and B. Gatos, "Handwritten Text Line Segmentation by Shredding Text into its Lines," Proc. 10th International Conference on Document Analysis and Recognition (ICDAR'09), Barcelona, 2009, pp. 626-630.
[16] A. Lemaitre, J. Camillerapp, and B. Coüasnon, "Interest of perceptive vision for document structure analysis," Proc. Human Vision and Electronic Imaging XV, 2010.
[17] MATLAB Notes, http://www.mathworks.de/company/news_notes/win02/watershed.html

KangHan Oh
He received the B.S in Computer Science from Hanoi University of Science and Technology, Vietnam in 2010. And he received the M.E. in the Department of Computer Science, Chonnam National University, Korea. In 2013. His main research interests include pattern recognition, image processing, and text recognition, object segmentation and object tracking.

SooHyung Kim
He received his B.S. degree in Computer Engineering from Seoul National University in 1986, and his M.S. and Ph.D degrees in Computer Science from Korea Advanced Institute of Science and Technology in 1988 and 1993, respectively. From 1990 to 1996, he was a senior member of research staff in Multimedia Research Center of Samsung Electronics Co., Korea. Since 1997, he has been a professor in the Department of Computer Science, Chonnam N ational University, Korea. His research interests are pattern recognition, document image processing, medical image processing, and ubiquitous computing.

InSeop Na
He received his B.S., M.S. and Ph.D. degree in Computer Science from Chonnam National University, Korea in 1997, 1999 and 2008, respectively. Since 2012, he has been a research professor in Department of Computer Science, Chonnam National University, Korea. His research inte rests are image processing, pattern recognition, character recognition and digital library.

GwangBok Kim
He received his B.E. degree in in Electronic & Computer Engineering from Chonnam National University, Korea in 2013. He has been taking the M.E. course in Electronics & Computer ngineering at Chonnam National University, Korea. His research interests are pattern recognition, machine learning and Image processing.