International Journal of Networked and Distributed Computing, Vol. 4, No. 1 (January 2016), 11-21

Extraction of Text Regions from Complex Background in Document Images by Multilevel Clustering

Hoai Nam Vu, Tuan Anh Tran, Na In Seop, Soo Hyung Kim*

Department of Computer Science, Chonnam National University,
77 Yongbong-ro, Buk-gu,
Gwangju, 500-757, South Korea
E-mail: nambkhn128@gmail.com, trtanh@hcmus.edu.vn, ypencil@hanmail.com, shkim@chonnam.ac.kr

Abstract

Textual data plays an important role in a number of applications such as image database indexing, document understanding, and image-based web searching. The target of automatic real-life text extracting in document images without character recognition module is to identify image regions that contain only text. These textual regions can then be either input of optical character recognition application or highlighted for user focusing. In this paper we propose a method which consists of three stages-preprocessing which improves contrast of grayscale image, multi-level thresholding for separating textual region from non-textual object such as graphics, pictures, and complex background, and heuristic filter, recursive filter for text localizing in textual region. In many of these applications, it is not necessary to identify all the text regions, therefore we emphasize on identifying important text region with relatively large size and high contrast. Experimental results on real-life dataset images demonstrate that the proposed method is effective in identifying textual region with various illuminations, size and font from various types of background.

Keywords: Multilevel, K-means, Connected Component, Thesholding.

1. Introduction

Text information retrieval from general document images provides many interesting applications in document layout analysis and understanding, such as optical character recognition and image data compression. Until now, many algorithms were proposed for extracting text from binary document images [1], [2], [3]. In recent years, developments in multimedia technology have led to a number of document images with decorated style character block in complex background. These character blocks of document image and their complex background are always highly contrastive for attracting audience focus. However, most of current methods can not work well for identifying text region from various type of document image. Compared to binary document images, text identifying in various complex background images comes with a huge number of challenges associated with the complexity of background image, variety, change of character size, character brightness and color, the mixture of textual object and background. a few newly developed thresholding method are useful in separating text re-

* Corresponding Author.

gion from other non-text regions. These approaches [4], [5], [6] heavily depend on results of binary algorithm. However, binary images obtained by thresholding method techniques are sensitive to noise, distortion and the quality of input image. In [7], Parker proposed a local gray intensity gradient thresholding method which is effective for identifying text region in badly illuminated document images. This method is based on binary image, therefore, its application is restricted to identifying text region from background which is not too complex in the view of illumination change. A local and adaptive binarization method was proposed by Sauvola et al. This method firstly performs a rapid classification of local contents of page to background, pictures and text. Then a soft decision method (SDM) and text binarization method (TBM) are applied for calculating threshold value for each pixel of image. It can effectively identifying text region from images with complex backgrounds on condition that the contrast of image is absolutely high.

Many methods support a different viewpoint for identifying text region by modeling the features of text objects and backgrounds.[8] proposed the logical level technique to utilize local linearity features of character strokes, while [9] utilizes local statistical features of textual object. These approaches implement symmetric local windows with predefined size, and several pre-determined prior value of local features, and so that characters with stroke widths that are substantially thinner or thicker than prior value, or characters in extraordinary illumination contrasts with background may not be identified. Ye et al.'s method [10] integrates global thresholding, local thresholding and double edge extraction techniques to identify text region from document image with different complexities. In Amin and Wu's approach [11] Otsu's method firstly applied, then connected component labelling process is applied on thresholding image to determine the sub-image of interest, and these sub-images then are implemented another thresholding process to identify text region. Thus, in case of text regions are overlapping on other non-text region or background region, this method is hard to determine to exactly extract text region.

In recent year, several color segmentation based technique for text region extraction from color document image have been developed. Zhong [12] proposed two methods and a hybrid approach for locating text region in color images, such as in CD jackets and book covers. The first technique implements a histogram-based color clustering process to obtain connected-components with uniform colors and then several heuristic criteria are applied to classify them as textual object or non-textual objects. The second method locates text regions based on their distinctive spatial variance. In Jain an Yu's work [13] they proposed a method which decomposed color document into a set of foreground images in RGB color space. Yang and Ozawa [14] use HSI color space to divide color document into homogenous region to extract title and author information from book covers. Lluis and Dimosthenis [15] method based on human perception of content to extract textual object from natural scene.

To sum up, identifying text region from complex document image meets with difficulty which brought by following properties of complex background document images. Character in complex background document images may have different illumination, size, font styles, and may be adjoined with other non-textual objects with gradational, sharp variations in contrast.

In this research, we propose a method which consists of three stages- enhancement of contrast, multilevel thresholding, and connected-component based filter for identifying and extracting text region from these complex document images.

The remainder of this paper is organized as follows: Section 2 presents key idea of our proposed method. Experimental results and discussion are reported in Section 3. Section 4 will conclude this paper.

2. Proposed method

The problem of identifying text region from complex background document image come from the difference of illumination, size, font styles, and inhomogeneous background object can not be solved well only using global thresholding segmentation algorithm. The changing always happens locally both

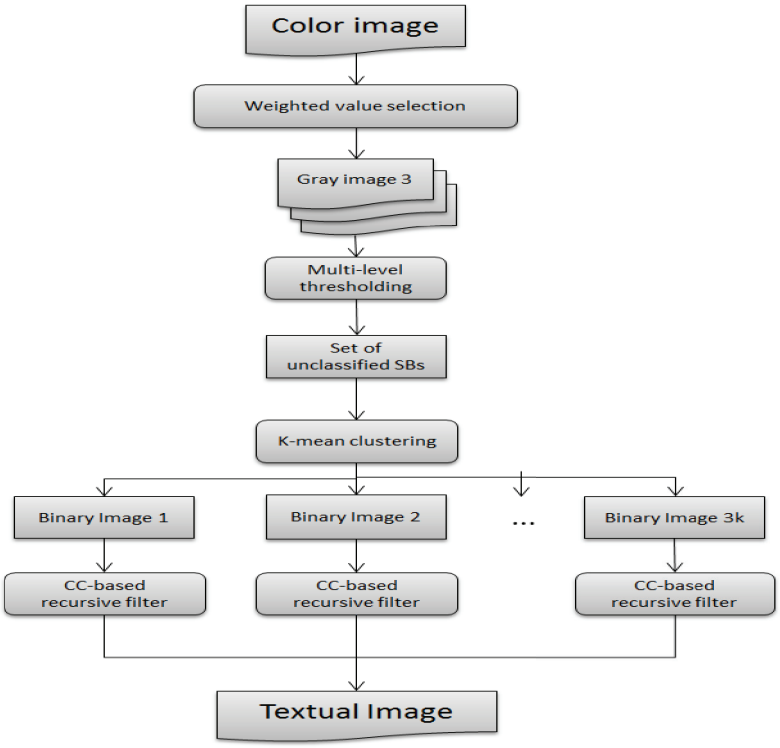

Figure 1: Flowchart of our proposed method.

in illumination and color of character. In this research, set of grayscale image is extracted from R, G, B plane with different weighted value. After that, multi-level thresholding process is applied to each grayscale individually to identify text region candidate plane of each grayscale image. Having all text region candidate plane extracted, a simple connected components based recursive filter is implemented to figure out what connected components is textual components. Finally, a combination scoring approach is reported to calculate the probabilistic text region score based on set of resultant images. If the region has enough score (larger than threshold) then it is classified as textual component. The flowchart of our proposed system is given in Fig. 1.

$$g(i,j) = \alpha_1 R(i,j) + \alpha_2 G(i,j) + \alpha_3 B(i,j) \qquad (1)$$

Where i, j are the position of pixel, g is grayscale image extracted from R, G, B grayscale image. In our experimentation, three set of $(\alpha_1, \alpha_2, \alpha_3)$ are chosen for creating three grayscale images. If the number of grayscale images increases, the precise of our system also increases, however, the processing time also increases.

2.1. *Contrast Enhancement*

In general, textual objects included in complex background document images are always in high contrast to other non-textual objects as well as background components. The purpose of document image synthesis is to attract audience focus to the content of the document. However, due to some distortion of scanning and printing process the contrast of document image may be reduced. This makes the following stage of our system much more difficult. We can not recognize whether entire object is textual object or not even by our own eyes. In order to recover, preserve as well as improve the quality of complex document image a contrast enhancement technique so-called histogram equalization [16] is implemented. Histogram equalization adjusts image intensities by

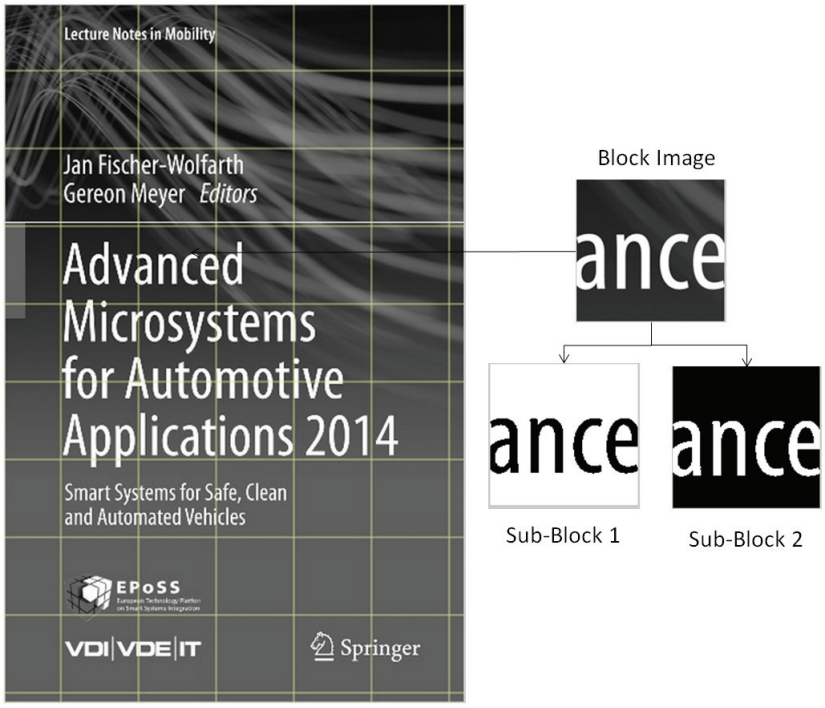

Figure 2: Example of Block and Sub-Block images in our system.

calculating probability of gray level in order to enhance the contrast of image. The transformation function is used to transform entire image to enhanced image is floor() function which rounds down the value of variable to the nearest integer. The equation of histogram equalization is as follow:

$$g_{i,j} = floor\left((L-1)\sum_{n=0}^{f_{i,j}} p_n\right) \qquad (2)$$

Where $g_{i,j}$ is the gray intensity of output image at position (i,j). L is the highest gray level. p_n is the probability of pixel value n of the input image.

2.2. *Multi-level Thresholding*

First of all, input grayscale image is divided into small blocks of size HxW to use the local feature of input image. Then multi-level thresholding technique is applied to each block individually to extract set of SBs (sub-block). Fig.2 shows an example of SB extracted from the original block image $\beta^{i,j}$ In the multi-level thresholding approach, mean and variance of gray intensity of input image are used to find optimal set of thresholds for segmenting the image into multiple levels. The algorithm is applied recursively on sub-class computed from the previous step so as to find threshold and a new sub-class for the next step. The recursive process is stopped by using PSNR (peak signal to noise ratio) value, measured in decibel (dB).

$$PSNR = 20log_{10}\left(\frac{255}{RMSE}\right) \qquad (3)$$

Where RMSE is the root mean-squared error, defined as follow

$$RMSE = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[I(i,j) - \widehat{I}(i,j)\right]^2} \qquad (4)$$

Here I and $\widehat{I}$ are the original and thresholded images, of size M x N respectively.

In order to find optimal multiple thresholds for segmenting the input image, some statistical features need to be defined. Let p_n denote the normalized

histogram of block ($\beta^{i,j}$ where i,j is the index of block position in original image) image.

$$p_n = \frac{number\,of\,pixels\,with\,intensity\,n}{total\,number\,of\,pixels} \quad n = 0,1,...,L-1. \tag{5}$$

For the n thresholds, $\beta^{i,j}$ is divided into n+1 classes, gray intensities of pixels in $\beta^{i,j}$ are classified by applying a threshold set **S**, which consists of n thresholds where $\mathbf{S} = \{s_k | k = 1,2,...n\}$. These classes are denoted as C_0, C_1,..., C_n. Then the statistical feature of class C_k such as cumulative probability, the mean, the variance denoted by w_k, μ_k and σ_k, respectively can be computed as

$$w_k = \sum_{g=t_k+1}^{t_{k+1}} p_n, \qquad \mu_k = \frac{\sum_{g=t_k+1}^{t_{k+1}} n p_n}{w_k}$$
$$\sigma_k = \sqrt{\frac{\sum_{g=t_k+1}^{t_{k+1}} p_n (n-\mu_k)^2}{w_k}} \tag{6}$$

Figure 3: Result of clustering stage; (a) is the input grayscale image, (b),(c),(d),(e) is the 1st, 2nd, 3rd, 4th cluster, respectively, (f) is result of sauvola binarization for comparison.

Following steps is the procedure of multi-level thresholding algorithm.

1. Range $\mathbf{R} = [a,b]$; initially a=0 and b=255.

2. Find mean (μ) and standard deviation (σ) of the all pixels in **R**.

3. Sub-ranges' threshold boundaries s_1 and s_2 are computed as $s_1 = \mu - \gamma_1\sigma$ and $s_2 = \mu + \gamma_2\sigma$; where γ_1 and γ_2 are free parameters.

4. Pixels with intensity values in the interval $[a,s_1]$ and $[s_2,b]$ are assigned threshold value equal to the respective weighted means of their values.

5. $a = s_1 + 1, b = s_2 - 1$.

6. Calculate PSNR value for stopping condition. If PSNR < 0.1 dB then stopping iterating.

7. Repeat step 4 with $s_1 = \mu$ and $s_2 = \mu + 1$

The result of multi-level thresholding is a few Sub-block components (SBs) which contain only pixel having grayscale value between their two boundaries. Same procedure is repeated through all $\beta^{i,j}$ in the input image to extract set of SBs in order to perform the next step of our system so called SBs clustering into meaningful binary images.

All the features of SBs are extracted from the grayscale image of $\beta^{i,j}$ such as number of pixel contained on SBs, number of pixels contained on left boundary, right boundary, etc. All these features together with w_k, μ_k and σ_k is used to create feature vector that is the input for K-means clustering algorithm.

Given a set of observation $(x_1, x_2, ..., x_n)$, where each observation is a d-dimensional real vector, the aim of well-known K-means clustering is an algorithm to cluster n observations into k partitions an tries to achieve is to minimize total intra-cluster variance.

$$arg_s min \sum_{i=1}^{k} \sum_{x_j \S_i} \|x_j - \mu_i\|^2 \tag{7}$$

Where there are k clusters $S_i, i = 1,2,...,k$ and μ_i is the center of all observation point x_j in S_i

In our scenario, number of observations is the number of SBs extracted from previous step and each dimensional is a feature extracted from SB.

After K-means clustering process, all textual objects, non-textual objects such as picture, graphic, table are classified into meaningful homogeneous region (binary image). These binary images can be

effectively analyzed in detail in following last process. Textual objects in these binary images are distinctly separated from other textual objects and non-textual objects as shown in Fig. 3. That is mandatory condition so as to apply connected component based filter the last step in our proposed system. It can be clearly seen from the Fig. 3 at cluster 1st all textual objects have been separated completely from the other non-textual objects and background objects

2.3. *Connected-Component based Filter*

For textual object extracting process, the last step of our proposed system, we implement a recursive algorithm which bases on connected component analysis. Our algorithm which shown in Fig. 4 includes three main part- connected component analysis, heuristic filter, non-heuristic filter.

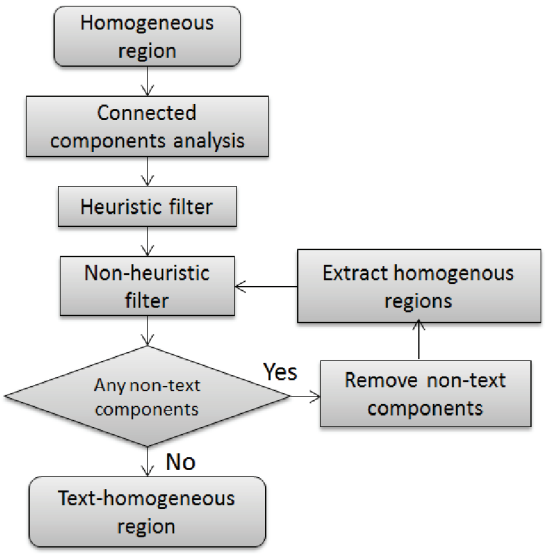

Figure 4: Flowchart of connected-component based filter.

2.3.1. *Region Segmentation*

The real-life document image is usually included various regions such as textual regions, non-textual regions, complex background regions, line, table,etc. Moreover, text string horizontally or vertically is often homogeneous and white spaces between them are almost the same. These properties are the key idea to segment document image into various different regions. Firstly, histogram of horizontal projection is extracted, after that Run Length Encoding is implemented on the horizontal projection to find out the large of white line and black line. Then, homogeneity of each region is taken into account which bases on the variance of black line and white line. Lastly, in order to get position to split, the most distinctive space of black and white line is identified based on some heuristic rules. The process is repeated until entire region obtained becomes homogeneous. At the same time, all steps are performed on vertical direction and get the homogeneous region following this direction.

2.3.2. *Connected Component Analysis*

Connected components analysis is the process of extracting and labeling connected component from a binary image. All pixels of the elements of an image that are connected and have the same value are extracted and assigned to a separate component. Let CCs be all connected components in the region extracted from the previous step, CC_i is the ith connected component in the region and B(CC_i) is the bounding box of it. Based on the characteristics of text a process so-called recursive filter is implemented to classify textual region and non-textual region in a level of homogeneous region.

2.3.3. *Heuristic Filter*

We find CC_i which can not be text without attention o its relative position in the region which is considering. It can be clearly seen that, these condition must be precise and very stringent, because they have a strong influence in whether we are looking at a separate region or not. Firstly, because of the human vision, as well as quality of the camera is various and limited, the CC_i that has a low area (number of foreground pixel), will be removed. Second, the CC_i is classified as non-text if B(CC_i) contains more than three other B(CC_j). Third, if the rate of CC_i's area with B(CC_i)'s area is too low, it may be the noise or diagonal components. Fourth, if the ratio between the width and the height of B(CC_i) is too low or too

high , the CC_i is also not a text element. As we already know, binarized images always create noise and generate "miss connected component". Therefore, this filter should only choose the extremely different elements. Other suspect components will be selected in non-heuristic filter.

2.3.4. Non-Heuristic Filter

In this stage, the structure of textual object and the relationship between the CC_i are examined in detail. Textual object and textual string usually appear in rows or columns, therefor the median and variance of these properties are taken into account to classify them. Three important factors obtained from the connected components are area, width and height. In the region, if area (or width, height) of CC_i has the maximum value and has a big difference between them and median value of homogeneous region. At that time it is considered to be non-textual object. We will compute the distance between these CC_i and the nearest connected components in the same row. If this distance is perceived to be larger than the mean of white space, then the CC_i will be classified as non-textual object. Experimentation showed that the use of the median generated desirable results, especially in the case that there are many connected components in considering region.

2.3.5. Recursive Filter

After implementing connected component analysis, all information about non-textual object is extracted. If the region does not contain any non-textual object more, it means that the regions we are considering contain only textual object. Conversely, if the region still contains some non-textual object, we will continue to segment them by using recursive filter. First of all, non-textual objects are removed from entire region after saving their information. Then, we segment this region to get smaller region (higher level of homogeneity) by using simple X-Y cut technique [17]. These regions will be checked again by connected component analysis to identify non-textual object. The goal of this process is to remove all non-textual objects in all level of homogeneity. The number of homogeneous level is unlimited; iteration will stop when all level of homogeneous region do not contain any non-textual object. This will ensure that all suspected parameters should be taken into account precisely and clearly. Now, all non-textual objects are eliminated, however, from the original document image, we just get many high level homogeneous text regions. These region need to be rearranged following their original position. Finally, all textual objects in the original document are obtained in the original size and position. Fig. 5 shows comparison results of textual object between our proposed method and other well-known methods. Lluis and Dimothenis method extracts textual object well but still misses some characters and non-textual objects are considered as textual objects.

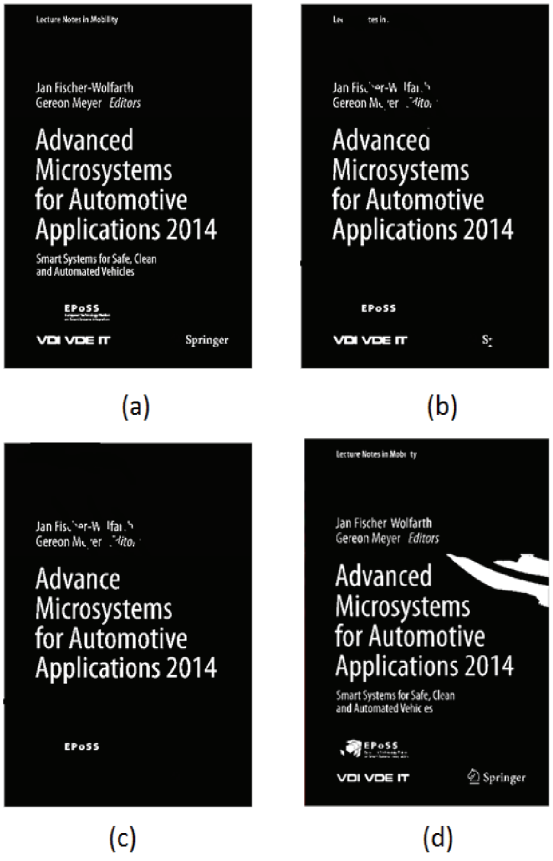

Figure 5: Result of textual regions extraction stage; (a) our result, (b) result of Jain and Yu' method, (c) result of K.Sobottka, (d) result of Lluis and Dimosthenis

Figure 6: Sample images in test dataset

3. Experimental Results

In this section, the performance of our system is evaluated and compared to other well-known text extraction techniques. Our test dataset includes 100 real-life document images are chosen for experiments on performance evaluation of text extraction. These images as shown in Fig. 6 are collected from various sources such as book cover, advertise banner, magazine, etc. These images also includes textual objects in various color, font styles, and sizes, including adjoined or overlapped text with pictorial, graphics, table, and background objects. In the view point of character accuracy-based evaluation with accuracy score provided by Eq. (8), our proposed method has achieved high performance compared to other well-known methods in literature. In addition, processing time is fast which proves the efficiency of our proposed method. However, if the gray color of input image varies in wide range, the processing time of our proposed method will increase linearly. The number of clusters in K-means clustering algorithm also affects the processing time of our system. In our scenario, K is chosen as 4 practically when running on our dataset and ICDAR2013 dataset.

Table 1: Accuracy comparison on our test dataset

Method	Accuracy
Jain and Yu's method[13]	79.5
K. Sobottka[6]	83.3
Lluis and Dimosthenis[15]	85.1
Our proposed method	**94.3**

Table 2: The processing time for text extraction

Steps of Processing	Average Time (Second)
Pre-processing	0.53
K-mean algorithm	2.56
Texual objects extraction	3.03

Our proposed system is implemented on an In-

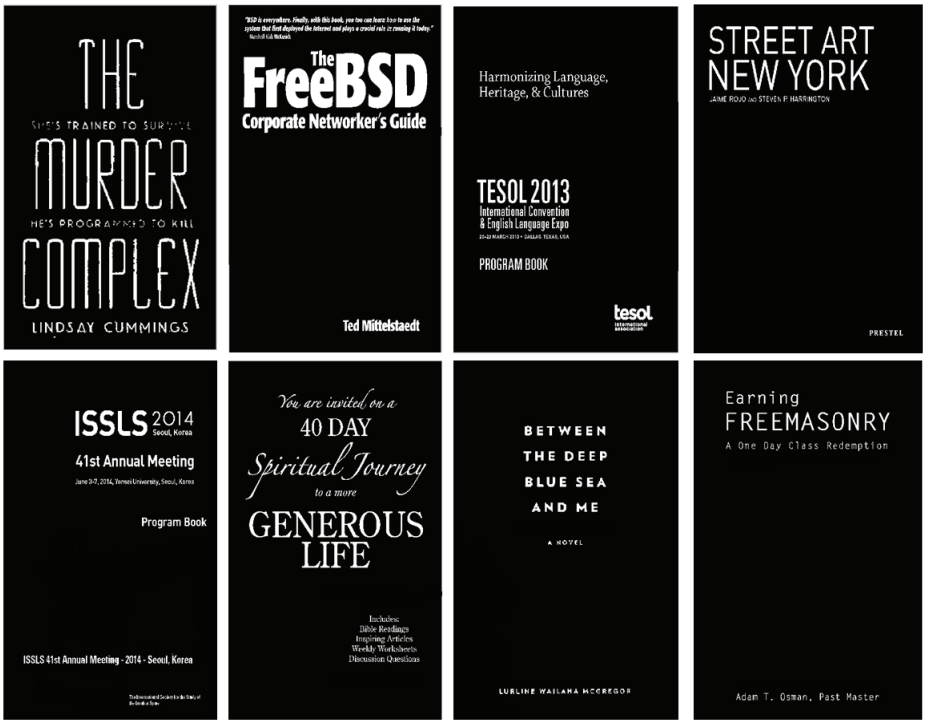

Figure 7: Final result of our system

tel core i5 3.2GHz personal computer using MATLAB language. The processing time depends heavily on the size of the image. Our dataset image size changes from 0.5 Megapixel to 2.5 Megapixel. Most of the processing time is spent on K-means algorithm and connected component analysis of textual objects classification of the last stage. The accuracy score is calculated using following equation.

$$acc\_rate = \frac{No.of\,texual\,objects\,extracted}{No.of\,total\,texual\,objects} \quad (8)$$

Fig. 7 shows final results of our proposed system. Those are textual objects extracted from the sample images given in Fig. 6. In addition, comparison result between our proposed method and the other methods is given in Table 1. And the processing time of our proposed method is also shown in Table. 2. These results prove the efficiency of our proposed method. The method of Jain and Yu depends heavily on the binary result, in case of noisy binary image, that method may be failed to extract some textual component from the input image.

Table 3: Accuracy comparison on ICDAR2013

Method	F-score
USTB_texStar	87.74
TH-TextLoc	80.96
Our proposed method	**77.51**
I2R_NUS_FAR	77.27
Baseline	76.27
Text Detection	75.81
I2R_NUS	75.34
BDTD_CASIA	72.53
OTCYMIST	71.09
Inkam	55.00

Performance of our proposed method is also evaluated on ICDAR2013 dataset for text localization purpose. ICDAR2013 dataset for text localization includes born-digital images from web and email. The resolution of images are low varying from 1.73 KB to 664 KB, therefor our proposed

Figure 8: Results on ICDAR2013 dataset

method is not the highest performance one.

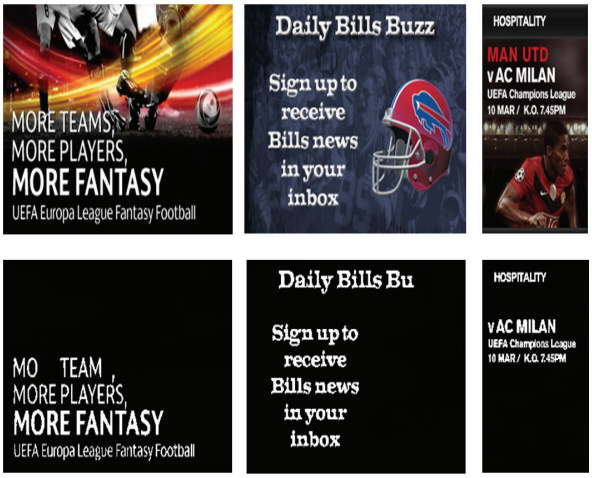

Figure 9: Failure cases on ICDAR2013 dataset

4. Conclusion

In some case, the contrast of dataset images is not that high, that also affects the performance of our proposed method. However, our proposed method has still achieved promising results in comparison to other methods which listed in ICDAR2013 competition about text localization task. Table. 3 shows the comparison result of our proposed method to other methods on ICDAR2013 dataset. F-score value is calculated following [18] under report of Robust Reading Competition ICDAR2013 [19]. Some samples images and our results on ICDAR2013 dataset are shown in Fig. 8. In case of absolute low contrast image, our proposed method is failed to extract all textual object in the image as shown in Fig. 9

In this paper a technique to automatically extract textual objects from real-life document image is proposed. This can be applied to various type of document image such as magazine, book cover, paper, CD cover with complex background. To utilize the local feature of input image, a division of document into small block is implemented, after that K-means clustering algorithm is applied to set of small sub-block after multi-level thresholding method is implemented on set of block image. Lastly, a connected-component based filter is used for separating and extracting the textual objects from the other objects in the set of binary image obtaining from the previous stage. Our proposed technique has achieved comparative results compared to the other well-known text extraction method, which prove the efficiency and advantages in case of real-life docu-

ment image. However, our proposed system still remains some shortcomings such as choosing optimal value of weighted parameter of RGB plane, dependence of K number on dataset, absolutely low contrast input image. In the future research, we plan to apply some color processing before our system to improve performance of our system with low contrast image. And some textual characteristics are taken into account to improve the last stage of our proposed system connected-component based filter.

Acknowledgments

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.O'Gorman, R.Kasturi, Document Image Analysis, IEEE Computer Society Press, Silver Spring, MD, 1995.
2. L.A.Fletcher, R.Kasturi, A robust algorithm for text string separation from mixed text/graphics images, IEEE Trans. Pattern Anal. Mach. Intell. 10(6)(1988) 910-918.
3. J.L.Fisher, S.C.Hinds, D.P.D'Amato, Rule-based system for document image segmentation, in: Proceedings of the 10th International Conference on Pattern Recognition, 1990, pp. 567-572.
4. Y.Liu, S.N.Srihari, Document image binarization based on texture features, IEEE Trans. Pattern Anal. Mach. Intell. 19(5)(1997) 540-544.
5. M.Cheriet, J.N.Said, C.Y.Suen, A recursive thresholding technique for image segmentation, IEEE Trans. Image Process. 7(6)(1998) 918-921.
6. Sobottka, K. and Kronenberg, H. and Perroud, T. and Bunke, H., Text extraction from colored book and journal covers, International Journal on Document Analysis and Recognition, 2000, pp. 163-176.
7. J.R.Parker, Gray level thresholding in badly illuminated images, IEEE Trans. Pattern Anal. Mach. Intell. 13(8)(1991) 813-819.
8. M.Kamel, A.Zhao, Extraction of binary character / graphics images from grayscale document images, CVGIP: Graphical Models Image Process. 55(3)(1993) 203-217.
9. N.B.Venkateswarlu, R.D.Boyle, New segmentation techniques for document image analysis,Image and Vision Comput. 13(7)(1995), 573-583.
10. X.Ye, M.Cheriet, C.Y.Suen, Stroke-model-based character extraction from gray-level document images, IEEE Trans. Image Process. 10(8)(2001) 1152-1161.
11. A.Amin, S.Wu, A robust system for thresholding and skew detection in mixed text/graphics documents, Int. J. Image Graphics 5(2)(2005) 247-265.
12. Y.Zhong, K.Karu, A.K.Jain, Locating text in complex color images, Pattern Recognition 28(10)(1995) 1523-1535.
13. A.K.Jain, B.Yu, Automatic text location in images and video frames, Pattern Recognition 31(12)(1998) 2055-2076.
14. H.Yang, S.Ozawa, Extraction of bibliography information based on the image of bookcover, IEICE Trans. Inf. Syst. E82-D(7)(1999) 1109-1116.
15. Lluis Gomez and Dimosthenis Karatzas, Multi-script Text Extraction from Natural Scenes, in Proc. ICDAR, 2013.
16. Rafael C. Gonzalez, and Richard E. Woods, "Digital Image Processing", 2nd edition, Prentice Hall, 2002.
17. G. Nagy and S. Seth. Hierarchical representation of optically scanned documents. In Proc. of the 17th Conf. on Pattern Recognition, pp. 347-349, 1984.
18. C.Wolf and J.-M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," International Journal of Document Analysis and Recognition (IJDAR), vol.8, no.4, pp. 280-296, 2006.
19. Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; Gomez i Bigorda, L.; Robles Mestre, S.; Mas, J.; Fernandez Mota, D.; Almazan Almazan, J.; de las Heras, L.-P., "ICDAR 2013 Robust Reading Competition," Document Analysis and Recognition (ICDAR), 2013 12th International Conference on , vol., no., pp.1484,1493, 25-28 Aug. 2013.
20. Antonio Clavelli, Dimosthenis Karatzas, and Josep Llados. (2010). "A framework for the assessment of text extraction algorithms on complex colour images " In 9th IAPR International Workshop on Document Analysis Systems (1926).