Available online at www.sciencedirect.com
ScienceDirect
Signal Processing 88 (2008) 1907–1916
SIGNAL PROCESSING
www.elsevier.com/locate/sigpro

Fast communication

# A new approach for text segmentation using a stroke filter

Cheolkon Jung[a,*], Qifeng Liu[b], Joongkyu Kim[a]

[a] *School of Information and Communication Engineering, Sungkyunkwan University, 300 Cheoncheon-dong, Suwon, Kyunggido 440-746, Republic of Korea*
[b] *Samsung Advanced Institute of Technology, Yongin, Kyunggido 446–712, Republic of Korea*

Received 29 August 2007; received in revised form 4 February 2008; accepted 4 February 2008
Available online 13 February 2008

**Abstract**

We propose a new method for achieving robust text segmentation in images by using a stroke filter. It is known that to segment text accurately and robustly from a complex background is a very difficult task. Most of the existing methods are sensitive to text color, size, font, and background clutter, because they use simple segmentation methods or require prior knowledge about text shape. In this paper, we attempt to consider the intrinsic characteristics of the text by using the stroke filter and design a new and robust algorithm for text segmentation. First, we describe the stroke filter briefly based on local region analysis. Second, the determination of text color polarity and local region growing procedures are performed successively based on the response of the stroke filter. Finally, the feedback procedure by the recognition score from an optical character recognition (OCR) module is used to improve the performance of text segmentation. By means of experiments on a large database, we demonstrate that the performance of our method is quite impressive from the viewpoints of the accuracy and robustness.


*Keywords:* Text segmentation; A stroke filter; Color polarity determination; Text information extraction

## 1. Introduction

Texts in images and videos contain important and useful information, which can help a machine to understand their content. Therefore, text information is widely employed in the fields of automatic annotation, indexing, summarization, and searching of videos. The extraction of text information is very important because texts contain high-level semantic information. In general, the extraction of text information from images and videos involves three major steps: text localization, text segmentation, and text recognition [1–3].

In this study, we focus on the text segmentation, which is employed to separate text pixels from the background in a text image. Performing text segmentation is considerably difficult in video images as compared to scanned images by using image scanners. This is because the scanned images generally have a clean and white background, whereas video images often have a very complex background without prior knowledge about the text color. Although some systems have been successfully developed for extracting videotexts [4], few researchers have addressed text segmentation in

*Corresponding author. Tel.: +82 31 290 7199; fax: +82 31 290 7687.
*E-mail addresses:* ckjung@ece.skku.ac.kr (C. Jung), qfliu1976@yahoo.com.cn (Q. Liu), jkkim@skku.edu (J.K. Kim).

0165-1684/$ - see front matter 
doi:10.1016/j.sigpro.2008.02.002

video images. The text segmentation strategies adopted up to recently can be classified into two main categories: (1) difference-based (or top-down) and (2) similarity-based (or bottom-up) methods. The first method is based on the difference in contrast between the foreground and background, for example, the fixed thresholding method [5], Otsu's adaptive thresholding method [6], global and local thresholding method [7], Niblack's method [8], and the improved Niblack method [9]. In general, these methods are simple and fast; however, they tend to fail when the foreground and background are similar.

Alternatively, the similarity-based method clusters pixels with similar intensities. For example, Lienhart [10] used the split and merge algorithm, and Wang et al. [11] used a method in which edge detection, watershed transform, and clustering were combined. However, these methods are unstable because they exploit many intuitive rules for the text shape. As an alternative, Chen et al. [12] converted text pixel clustering into a labeling problem by using the Gibbsian EM algorithm. This method is effective but time consuming.

In the meantime, only a few papers address the determination of text color polarity. Most of the existing methods assume that the text color is always brighter or darker than the background. Apparently, this assumption limits the practical applications of these methods. Chen et al. [12] determined the text color polarity by multiple hypotheses testing using an optical character recognition (OCR) module. This method, however, is effective but consumes a significant amount of time [12]. Sato et al. [13] analyzed the connected components with some prior knowledge. This method, however, is not robust because text components are generally connected to the background. An effective method proposed by Song et al. [14] is based on the intrinsic relationship between the text and background edges. However, this method fails when the text localization is not accurate or the edge clutter in the background is included.

The main problem in most of the existing methods is that they are sensitive to text color, size, font, and background clutter; this is because these methods simply exploit the general segmentation methods or the prior knowledge about the text shape. In this paper, we attempt to consider the intrinsic characteristics of text and design a robust algorithm particularly for text segmentation. Fig. 1 shows the overall flowchart of our method.

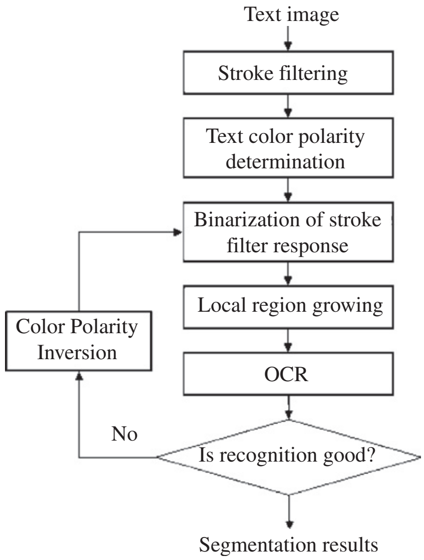


Fig. 1. Overview of our text segmentation method.


Fig. 2. Text image.

The remainder of this paper is organized as follows. Section 2 explains the stroke filter in brief. Sections 3 and 4 present the text color polarity determination and local region growing procedures. Section 5 describes the feedback procedure by the recognition score from an OCR module. Section 6 shows some experimental results and the corresponding analysis. The conclusion is provided in Section 7, and the future study is discussed.

## 2. A stroke filter

A stroke is defined as a straight line or arc used as a segment of text, and the texts in images and videos comprise one or several strokes, as shown in Fig. 2. An image is defined as a text, if and only if, several stroke-like structures exist in it. A stroke filter is designed based on this definition using the local region analysis. In order to design the stroke filter, we first define a local image region to be a stroke-like structure, if and only if: (1) it is different from its lateral regions, (2) intensities of its lateral regions

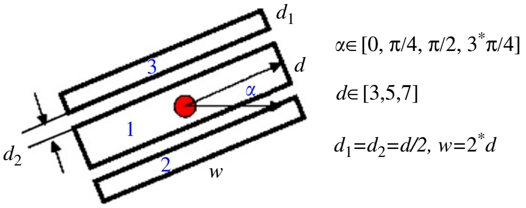


Fig. 3. A stroke filter.

are similar, and (3) it is nearly homogenous with respect to its intensities.

For each pixel in the source image, we compute its stroke filter response as follows. As shown in Fig. 3, the central point of a stroke filter denotes an image pixel $(x, y)$ around which three rectangular regions can be observed. Let index 1 denote the central region and indices 2 and 3 denote each of the lateral regions. The orientation and scale of these local regions are determined by the parameters $\alpha$ and $d$, where $d$, the width of the rectangular region, is determined based on prior knowledge of the text obtained by conducting experiments on text images. The distance, $d_2$, between the central region and lateral regions in the stroke filter is due to the fact that dark or bright lines are often embedded around some texts to convey their meanings efficiently, and blurred edges appear around texts as a result of their compression, as shown in Fig. 2. According to the definition of a stroke-like structure, we define the bright and dark stroke filter responses of the pixel $(x, y)$, $R^{\mathrm{B}}$ and $R^{\mathrm{D}}$, respectively, as follows:

$$R^{\mathrm{B}}_{\alpha,d}(x,y) = \frac{\mu_1 - \mu_2 + \mu_1 - \mu_3 - |\mu_2 - \mu_3|}{\sigma},$$
$$R^{\mathrm{D}}_{\alpha,d}(x,y) = \frac{\mu_2 - \mu_1 + \mu_3 - \mu_1 - |\mu_2 - \mu_3|}{\sigma}. \qquad (1)$$

The terms on the right-hand sides of Eq. (1) have clear physical meanings corresponding to the constraints of the definition of a stroke-like structure, where $\mu_i$ denotes the estimated mean of the intensities in the region $i$, where $i = 1, 2, 3$. The bright and dark stroke filter responses are proportional to $\mu_1 - \mu_2 + \mu_1 - \mu_3$ and $\mu_2 - \mu_1 + \mu_3 - \mu_1$, respectively, and are negatively proportional to $|\mu_2 - \mu_3|$. The parameter $\sigma$ denotes the standard deviation of the intensities in region 1 and is a measure of the extent to which the intensities of the region are spread out. Therefore, this parameter can reflect the homogeneity of the stroke and is inversely proportional to the response. Note that the greater the probability that the pixel $(x, y)$ belongs to a stroke-like structure, the higher the response.

By means of stroke filtering, we extract the stroke features ($R^{\mathrm{B}}$, $O^{\mathrm{B}}$, $S^{\mathrm{B}}$, $R^{\mathrm{D}}$, $O^{\mathrm{D}}$, $S^{\mathrm{D}}$) of any pixel $(x, y)$ as follows (note that ($R^{\mathrm{D}}$, $O^{\mathrm{D}}$, $S^{\mathrm{D}}$) have similar expressions):

$$\begin{cases} R^{\mathrm{B}}(x,y) = \max\limits_{(\alpha,d)} R^{\mathrm{B}}_{\alpha,d}(x,y), \\ O^{\mathrm{B}}(x,y) = \arg\max\limits_{(\alpha)} R^{\mathrm{B}}_{\alpha,d}(x,y), \\ S^{\mathrm{B}}(x,y) = \arg\max\limits_{(d)} R^{\mathrm{B}}_{\alpha,d}(x,y), \end{cases} \qquad (2)$$

where $R$, $O$, and $S$, respectively, denote the response, orientation, and scale of the stroke filter, whereas $B$ and $D$ denote the bright and dark stroke filters, respectively.

Some intermediate results in a real text image are shown in Fig. 4. Fig. 4(b) shows the text part of the corresponding original image in Fig. 4(a), where the text is located within the dotted circle. Fig. 4(c) shows the stroke response map formed by combining the bright and dark stroke responses, and Fig. 4(d) shows the binarized map of Fig. 4(c). Figs. 4(e) and (f) show the stroke width and angle map, respectively. The bright regions in Figs. 4(e) and (f), respectively, indicate the width and orientation angle with large scales. Observe that a stroke filter can simultaneously remove most step-like edges and enhance the text parts well, as shown in Fig. 4(d). Notice, however, that some non-text regions between two strokes are detected because all the dark and bright strokes are filtered, which means that we should be able to determine the text color polarity in order to select the response to be employed. In other words, the text color polarity determination procedure, which can determine whether the color of text is bright or dark, is required. This procedure is described in Section 3.

## 3. Text color polarity determination

For the correct text segmentation, it is very important to determine the color polarity of video text (i.e., whether the color of text is bright or dark) [14,19]. In order to determine the text color polarity automatically, we use two features as follows. We first perform bright and dark stroke filtering to obtain $R^{\mathrm{B}}$ and $R^{\mathrm{D}}$. Then, two features for the determination of text color polarity can be obtained, the first one $F_{\mathrm{R}}$ of which is the ratio of the sums of the magnitude of the bright and dark stroke


Fig. 4. Intermediate results by a stroke filter in a real text image: (a) original image, (b) text image, (c) stroke response map, (d) binarized map of (c), (e) stroke width map, and (f) stroke orientation angle map.

filter responses:

$$F_{\mathrm{R}} = \sum_{(x,y)} R^{(\mathrm{B})}(x,y) \Big/ \sum_{(x,y)} R^{(\mathrm{D})}(x,y). \tag{3}$$

We assume that the bright (dark) texts should have a strong response to a bright (dark) stroke filter as compared to a dark (bright) stroke filter. The second feature $F_{\mathrm{E}}$, inspired by Ref. [14], is the ratio of the sums of the number of edge points in the binarized response maps of the bright and dark stroke filters:

$$F_{\mathrm{E}} = N^{(\mathrm{B})}/N^{(\mathrm{D})}, \tag{4}$$

where $N^{(\mathrm{B})}$ and $N^{(\mathrm{D})}$ denote the number of edge points in the binarized map of the bright and dark stroke filter responses, respectively.

In Fig. 5, $N^{(\mathrm{B})}$ and $N^{(\mathrm{D})}$ denote the number of edge points in (b) and in (c), respectively, on a synthetic image (a). $F_{\mathrm{E}}$ is useful for the case of bright texts on a bright background (BonB) or dark texts on a dark background (DonD). Through experiments, we find that bright (dark) texts have few edge points in the binarized bright (dark) stroke filter response map, i.e., $N^{(\mathrm{B})} < N^{(\mathrm{D})}$ ($N^{(\mathrm{B})} > N^{(\mathrm{D})}$).

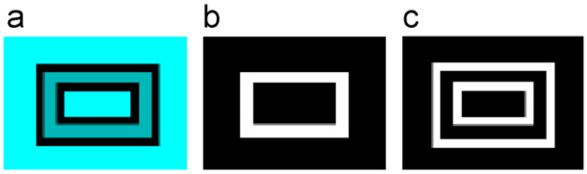


Fig. 5. Bright and dark stroke filter (SF) response maps: (a) synthetic image, (b) bright SF response map and (c) dark SF response map.

We manually select approximately 100 samples (50 for training and 50 for testing) from approximately 800 video images, which include images under complex conditions such as BonB or DonD. Some representative samples are plotted in the feature space spanned by ($F_{\mathrm{R}}$, $F_{\mathrm{E}}$), as shown in Fig. 6. Here we use a general support vector machine (SVM) classifier with a radial basis function (RBF) kernel. The determination accuracy of the text color polarity is approximately 96%, and this demonstrates that the feature ($F_{\mathrm{R}}$, $F_{\mathrm{E}}$) is stable and distinctive.

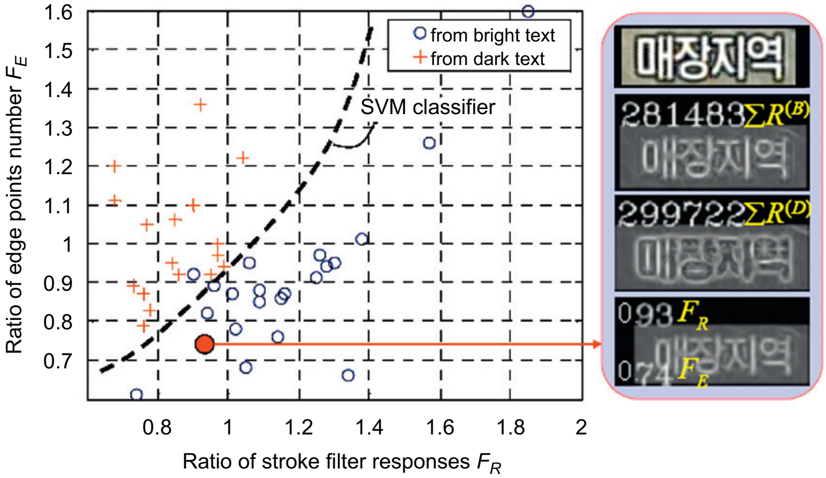


Fig. 6. Feature space and SVM classifier.


Fig. 7. Test examples: (a) source image, (b) binarized stroke response map, and (c) the local region growing result.

## 4. Local region growing

As shown in Fig. 7(b), many text pixels tend to be missed in the binarized stroke filter response map, and the missed text pixels should be recalled for accurate and robust text segmentation. In order to recall these pixels, we perform a local region growing procedure. The binarized response map is regarded as the initial segmentation result for the local region growing procedure. From the initial segmentation result and text image, we estimate the global probability density function (PDF), PDF($s$), of text intensity, $s$. Then, a non-text pixel is changed to the text pixel if the following three conditions are simultaneously satisfied: (1) the number of text pixels is greater than 3 in the local region ($3\times 3$ neighbors of the non-text pixel), (2) the probability of the non-text pixel's intensity, Pr($s$), is greater than a threshold $\mathrm{Th}_1$, and (3) the difference of intensity between the non-text pixel and its neighbors is lower than a value $\mathrm{Th}_2$. Here, we set the values of $\mathrm{Th}_1$ and $\mathrm{Th}_2$ to 0.155 and 30, respectively, and Pr($s$) is calculated by the global PDF as follows:

$$\Pr(s) = \mathrm{PDF}(s). \tag{5}$$

The local region growing procedure is repeated until no pixel is changed in the segmentation map. Therefore, the novelty of the local region growing procedure is that it is based on the stroke filter response and combines the global PDF and local similarities to achieve a reliable performance. The local region growing procedure is described in Table 1. In this table, the white and black pixels represent the text and non-text pixels, respectively.

Table 1
Local region growing algorithm

*Input*: $I$—initial segmentation result (binarized stroke filter response map obtained in Section 2); $S$—source text image

*Step* 1: From $I$ and $S$, we estimate PDF of text color

*Step* 2: For each white pixel in $I$, if the number of white pixels in its $3\times 3$ neighbors is within [3,9], then go to Step 3, else Step 2

*Step* 3: For each black pixel in the $3\times 3$ regions, if it is: (1) similar to its text neighbors and (2) of high probability according to PDF, then it is marked as text. Repeat Steps 3 and 2, until no pixel is changed

*Output*: Refined segmentation result

The test examples are shown in Fig. 7. Here, Figs. 7(a)–(c) show the original source image, the binarized stroke filter response map, and local region growing result, respectively. From Fig. 7(c), it can be observed that some missed text pixels are successfully recalled by the local region growing procedure.

## 5. Feedback from the OCR module

As mentioned in Section 3, 4% errors occur during the determination of the text color polarity, and these errors will have a detrimental effect on the recognition results. In order to improve the accuracy of the determination of text color polarity and the performance of text segmentation, we apply an additional verification step for the segmentation result of the local region growing procedure by an OCR module. We use the feedback of an average recognition score of characters from the OCR module, and the average recognition score is determined as follows.

In general, there are several characters in the segmented text image, and a character separation function that can decompose the segmented text image into sub-images of individual symbols is required. Character separation is one of the most important processes in the OCR module because the performance of character separation significantly affects the accuracy of character recognition. Our character separation function uses the feedback from the recognition score of the character recognition function, as shown in Fig. 8; furthermore, this method employs a split-merge strategy in which the split segments are merged into a character by using the recognition score obtained by the character recognition function [15]. The optimal separation path is determined by using two scores: (1) the geometric score that is used to estimate the likelihood of "being a character" by geometric features and (2) the recognition score obtained by the character recognition function. The geometric score is calculated by using two character evaluators, the squareness (SQU) and the internal gap (GAP), which are estimated by the Parzen window [16]. If the geometric score is low, the sub-image is eliminated in the separation paths and not recognized, by which the overhead of recognition can be reduced. The recognition score is determined by the distance between the recognition model and a separated sub-image. Here, the extracted feature is the angular directional feature (ADF) [17], and each character is recognized by linear discriminant analysis (LDA) in the character recognition function.

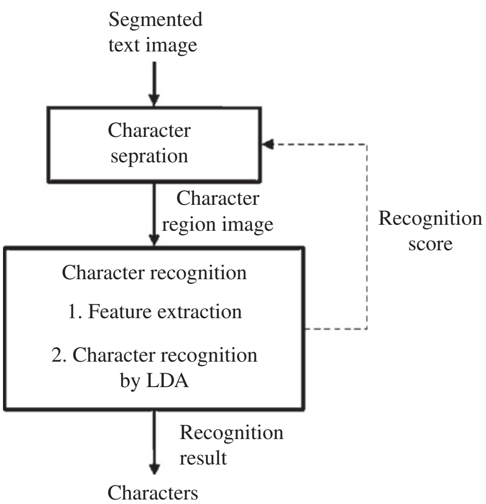


Fig. 8. An OCR module.

The average recognition score, $S_A$, of the segmented text image is represented by these two scores as follows:

$$S_A = \frac{1}{N}\sum_{i=1}^{N} S_i,$$
$$S_i = k_G \times S_G + k_R \times S_R, \tag{6}$$

where $i$ and $N$ denote the index and the number of characters, $k_G$ and $k_R$ are constants and $S_i$, $S_G$, and $S_R$ denote the total, geometric, and recognition scores of each character, respectively. In our OCR module, the two constants, $k_G$ and $k_R$, are assigned as 0.3 and 5, respectively.

If $S_A$ is smaller than a predefined threshold $Th_R$, the color polarity inversion and subsequent procedures are performed as shown in Fig. 1, where the threshold $Th_R$ is determined by analyzing the statistical distribution of the average recognition scores of the training samples [15]. We set the value of $Th_R$ as $-6.0$, and half of the 4% errors are corrected by the verification of the OCR module.

## 6. Experimental results

The experiments were performed by using a PC (CPU: Intel Pentium 4, 1.4 GHz) with VC++6.0. Our experimental database contains 792 images of news programs from the South Korean TV channels, KBS, MBC, and SBS, with resolutions of $720 \times 480$. We also use 90 images of resolution $720 \times 480$ from the channels CNN, BBC, and NBC. This database comprises a variety of cases, including texts with a wide spectrum of font-sizes, font-color and languages, and texts on complex backgrounds, texts of poor quality, etc. The total number of text lines and characters in these images are 966 and 6315, respectively.

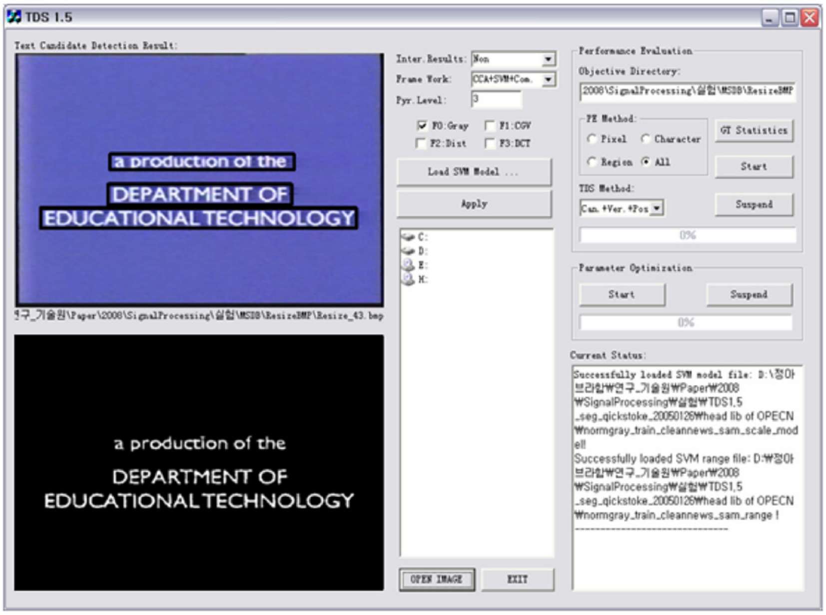


Fig. 9. User interface of the proposed text segmentation system.


Fig. 10. Examples of experimental results: (a) original source images and (b) segmented images.

The user interface of the proposed text segmentation system is illustrated in Fig. 9. The top-left window in this figure shows the results of the text localization process which are bounded in the black boxes [18]. The bottom-left window in this figure is the results of the text segmentation process. Here, the white and black regions represent the text and background, respectively. The right-side window in the figure shows the results of the performance evaluation on the test.

The proposed method exhibits a robust performance for the majority of the test images. Fig. 10 illustrates some examples of the segmentation results on English and Korean text images. It can be seen from the results that most of the text regions are well segmented despite the different languages and color polarities of the texts. Some other representative results are shown in Fig. 11 from which we can see that our text segmentation method is robust against challenging cases of BonB (a and b) or DonD, complex background with various edge and color clutter (c and e). In Fig. 11(d), each character in text regions has two colors which shows color variation of white to yellow in the vertical direction. Notice, however, that the color polarities in characters are same (bright), and texts are well segmented. This shows that our method is robust against color variation in one character as far as the color polarities in the character are the same. These results show that our method can simultaneously handle a wide range of complex cases in video images due to the benefit of the stroke filter from which the intrinsic characteristics of the text are utilized. On the other hand, most existing


Fig. 11. Some experimental results. Left: original source image. Right: segmented image.

segmentation methods fail when the background is complex (Figs. 11(c and e)), or the text image is BonB (Figs. 11(a and b)).

There are of course some undesirable experimental results in our method. For example, if one text image contains texts with different color polarities (Fig. 11(f)), texts with dark color polarity are not segmented, while those with bright color polarity are well segmented. This is due to the fact that our method basically assumes that the text image consists of texts with one color polarity. We expect that this problem can be resolved by adding a procedure, which separates the text image into individual sub-images with each color polarity before text segmentation. Furthermore, our method showed some instability when the height of stroke crowded text is less than 10 pixels (Fig. 11(g)), since the gap between each stroke is so near that our stroke filter cannot detect and distinguish the strokes. In our future work, we plan to scrutinize these problems.

We compare our text color polarity determination method with the state-of-the-art method proposed by Song [14] (hereinafter called the Song method), which is based on the statistic analysis. The accuracy of each method is evaluated as the relative frequency of the correctly determined text color polarities as follows:

$$\text{Accuracy} = \frac{\text{Number of correctly determined color polarity}}{\text{Number of text image}}. \tag{7}$$

Table 2 shows the experimental results of our method and the Song method. From this table, we can see that our stroke filter based determination method achieves a considerably higher (3.4%) accuracy than the statistic based Song method. However, it can be observed that the processing

Table 2
Performance comparison of two algorithms

| Method | Accuracy (%) | Time cost per frame (ms) |
|---|---|---|
| Our method | 94.0 | 15 |
| Song method | 90.6 | 10 |

Table 3
CER evaluation of two text segmentation methods

| Method | English CER (%) | Korean CER (%) | Overall CER (%) |
|---|---|---|---|
| Our method | 11.6 | 14.9 | 13.3 |
| Otsu method | 15.8 | 60.1 | 38.0 |

time for our method is approximately 5 ms/image slower than that for the Song method. Here, note that the feedback process of Section 5 by the OCR module is not included for the determination of text color polarity, and if this procedure is included, the final accuracy of our text segmentation method becomes 96.2%.

In order to evaluate the performance of our text segmentation method, the character error rate (CER) is calculated for another test. Here, CER is defined as

$$\mathrm{CER} = \frac{N_\mathrm{e}}{N}, \tag{8}$$

where $N_\mathrm{e}$ denotes the number of characters wrongly recognized by the OCR module and $N$ denotes the total number of characters. In this experiment, the proposed text segmentation method is compared with the Otsu [6] thresholding method (hereinafter called the Otsu method). The Otsu method is a simple but a classic solution that is employed by many text segmentation schemes. Since this method does not deal with the determination of color polarity, we used our proposed color polarity determination algorithm. The evaluation result of the CER of these two methods is summarized in Table 3. From this table, we can see that our method achieves significantly lower error rates for both English and Korean characters. This is because our method possesses the capability of handling multilanguage and complex backgrounds by using the stroke filter. Notice that in most cases, the CERs of Korean characters are larger than that of English characters and this seems to be due to the fact that Korean characters are structurally more complex than English characters.

Table 4
CER evaluation of intermediate results by each module

| Method | Initial segmentation results (%) | Local region growing results (%) | Feedback results (%) |
|---|---|---|---|
| Our method | 27.4 | 15.1 | 13.3 |

Table 4 shows the evaluation of the CER for the intermediate results by each module in our method where three intermediate results are obtained: the initial segmentation results by stroke responses (Sections 2 and 3), results after local region growing (Section 4), and the results after feedback by the OCR module (Section 5) which corresponds to the final result. As shown in this table, the local region growing and feedback procedures reduce 12.3% and 1.8% errors in the character recognition, respectively, which means that most of reduction in error rates occurs during the local region growing procedure.

## 7. Conclusions

A new method of text segmentation in images using a stroke filter is proposed in this paper. The benefit of the stroke filter is that it is capable of discovering the intrinsic characteristics of the text by making use of the relationship among the local regions. By using the response of the stroke filter, we could determine the text color polarity robustly with the accuracy of 94.0%, which exceeds that of the Song method by 3.4%. When the feedback procedure from an OCR module is included, the accuracy of our method even more improves up to 96.2%. We also make use of the response of the stroke filter as the initial seed region for the local region growing procedure. In this procedure, we obtain a reliable performance by combining the global PDF and local similarities, and consequently 12.3% errors could be reduced. Furthermore, 1.8% errors are additionally reduced by applying the feedback procedure from the OCR. Some experimental results and performance comparisons with other methods are reported in detail, thereby confirming that our method is capable of robustly handling texts from complex backgrounds by using the stroke filter. In addition, we believe that the stroke filter will play an important role in fields other than text segmentation. For example, the stroke filter could be applied to text verification using the features of the filter, or extended to the fields of line-like structure detection, such as road detection from remote images.

Our future study will mostly be focused on improving the OCR module. Although current OCR module is sufficiently efficient, it still leaves quite a room for improvement in the recognition rate. Currently, tensor techniques are newly proposed for image representation and object recognition, and we will continue developing our OCR module by using them [20–22] as well.

## Acknowledgments

The partial work reported in this paper was conducted while the authors were with the Samsung Advanced Institute of Technology (SAIT). The authors would like to thank the editor and anonymous reviewers for their valuable comments.

## References

[1] N. Dimitrova, H.J. Zhang, B. Shahraray, I. Sezan, A. Zakhor, T. Huang, Applications of video content analysis and retrieval, IEEE Multimedia 9 (3) (2002) 43–55.

[2] Y. Wang, Y. Liu, J.C. Huang, Multimedia content analysis using both audio and visual clues, IEEE Signal Process. Mag. 17 (6) (2000) 12–36.

[3] R. Liehart, A. Wernicke, Localizing and segmenting text in images and videos, IEEE Trans. Circuits Syst. Video Technol. 12 (4) (2002) 256–268.

[4] K. Jung, K.I. Kim, A.K. Jain, Text information extraction in images and video: a survey, Pattern Recognition 37 (5) (2004) 977–997.

[5] T. Sato, T. Kanade, E.K. Hughes, M.A. Smith, S. Satoh, Video OCR: indexing digital news libraries by recognition of superimposed caption, ACM Multimedia Syst. Special Issue Video Libr. 7 (5) (1998) 385–395.

[6] N. Otsu, A threshold selection method from gray-scale histogram, IEEE Trans. Syst. Man Cybern. 9 (1979) 62–66.

[7] F. Chang, G.C. Chen, C.C. Lin, W.H. Lin, Caption analysis and recognition for building video indexing system, Multimedia Syst. 10 (4) (2005) 344–355.

[8] C. Wolf, J. Jolion, Extraction and recognition of artificial text in multimedia documents, Pattern Anal. Appl. 6 (2003) 309–326.

[9] K. Zhu, F. Qi, R. Jiang, L. Xu, M. Kimachi, Y. Wu, T. Aizawa, Using adaboost to detect and segment characters from natural scenes, in: Proceedings of the International Workshop on Camera-based Document Analysis and Recognition, 2005, pp. 52–58.

[10] R. Lienhart, Automatic text recognition in digital videos, in: Proceedings of SPIE Image Video Processing, vol. 4, 1996, pp. 2666–2675.

[11] K. Wang, J.A. Kangas, W. Li, Character segmentation of color images from digital camera, in: Proceedings of the International Conference on Document Analysis and Recognition, 2001, pp. 210–214.

[12] D. Chen, J. Odobez, H. Bourlard, Text detection and recognition in images and video frames, Pattern Recognition 37 (2004) 595–608.

[13] T. Sato, T. Kanade, E.K. Hughes, M.A. Smith, Video OCR for digital news archive, in: Proceedings of the International Conference on Pattern Recognition, 2000, pp. 831–834.

[14] J. Song, M. Cai, M.R. Lyu, A robust statistic method for classifying color polarity of video text, in: Proceedings of the IEEE International Conference on Acoustics, Speech, Signal Processing, 2003, pp. 581–584.

[15] Y.H. Tseng, H.J. Lee, Recognition-based handwritten Chinese character segmentation using a probabilistic Viteribi algorithm, Pattern Recognition Lett. 20 (1999) 791–806.

[16] M.S. Kim, K.T. Cho, H.K. Kwag, J.H. Kim, Segmentation of handwritten characters for digitalizing Korean historical documents, in: Proceedings of the International Conference on Document Analysis and Recognition, 2004, pp. 114–124.

[17] K.T. Lim, A study on machine printed character recognition based on character type classification, J. Inst. Electron. Eng. Korea 40 (2003) 26–39.

[18] C. Jung, Q. Liu, J.K. Kim, Accurate text localization in images based on SVM output score, Image Vision Comput. (2006), submitted for publication.

[19] S. Antani, D. Crandall, R. Kasturi, Robust extraction of text in video, in: Proceedings of the International Conference on Pattern Recognition, 2000, pp. 831–834.

[20] D. Tao, X. Li, X. Wu, S.J. Maybank, General tensor discriminant analysis and Gabor features for gait recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1700–1715.

[21] D. Tao, X. Li, X. Wu, W. Hu, S.J. Maybank, Supervised tensor learning, Knowl. Inf. Syst. 13 (1) (2007) 1–42.

[22] D. Tao, X. Li, S.J. Maybank, X. Wu, Human carrying status in visual surveillance, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2006, pp. 1670–1677.