An Efficient Method for Text Detection in Video Based on Stroke Width Similarity

Viet Cuong Dinh, Seong Soo Chun, Seungwook Cha,
Hanjin Ryu, and Sanghoon Sull

Department of Electronics and Computer Engineering, Korea University, 5-1 Anam-dong,
Seongbuk-gu, Seoul, 136-701, Korea
{cuongdv,sschun,swcha,hanjin,sull}@mpeg.korea.ac.kr

Abstract. Text appearing in video provides semantic knowledge and significant information for video indexing and retrieval system. This paper proposes an effective method for text detection in video based on the similarity in stroke width of text (which is defined as the distance between two edges of a stroke). From the observation that text regions can be characterized by a dominant fixed stroke width, edge detection with local adaptive thresholds is first devised to keep text- while reducing background-regions. Second, morphological dilation operator with adaptive structuring element size determined by stroke width value is exploited to roughly localize text regions. Finally, to reduce false alarm and refine text location, a new multi-frame refinement method is applied. Experimental results show that the proposed method is not only robust to different levels of background complexity, but also effective to different fonts (size, color) and languages of text.

1 Introduction

The need for efficient content-based video indexing and retrieval has increased due to the rapid growth of video data available to consumers. For this purpose, text in video, especially the superimposed text, is the most frequently used since it provides high-level semantic information about video content and it has distinctive visual characteristic. Therefore, the success in video text detection and recognition would have a great impact on multimedia applications such as image categorization [1], video summarization [2], and lecture video indexing [3].

Many efforts have been made for text detection in image and video. Regarding the way used to locate text regions, text detection methods can be classified into three approaches: connected component (CC)-based method [4, 5, 6], texture-based method [7, 8], and edge-based method [9, 10]. The CC-based method is based on the analysis of geometrical arrangement of edges or homogeneous color that belongs to characters. Alternatively, the texture-based method treats text region as a special type of texture and employs learning algorithms, e.g., neural network [8], support vector machine (SVM) [11], to extract text. In general, the texture-based method is more robust than the CC-based method in dealing with complex background. However, the main drawbacks of this method are its high complexity and inaccuracy localization.

Y. Yagi et al. (Eds.): ACCV 2007, Part I, LNCS 4843, pp. 200–209, 2007.

Another popularly studied method is the edge-based method, which is based on the fact that text regions have abundant edges. This method is widely used due to its fast performance in detecting text and its ability to keep geometrical structure of text**.** The method in [9] detects edges in an image and then uses the fixed size horizontal, vertical morphological dilation operations to form text line candidate. Real text regions are identified by using the SVM. Two disadvantages of this method are its poor performance in case of complex background and the use of fixed size structuring element in dilation operations.

To deal with the background complexity problem, edge detection-based method should be accompanied by a local threshold algorithm. In [10], the image is first divided into small windows. A window is considered to be complex if the "number of blank rows" is smaller than a certain specific value. Then, in the edge detection step, a higher threshold is assigned for these complex windows. However, the "number of blank rows" criterion appears sensitive to noise and not strong enough to handle different text sizes. Therefore, how to design an effective local threshold algorithm for detecting edge is still a challenging problem of text detection in video.

The main problem of the above existing methods is that they are not robust to different text colors, sizes, and background complexity, since they simply use either general segmentation method or some prior knowledge. In this paper, we attempt to discover the intrinsic characteristic of text (namely the stroke width similarity) and then exploit it to build a robust method for text detection in video. From the knowledge of font system, it turns out that, if characters are in the same font type and size, their stroke widths are almost constant. In another view, a text region can be considered as a region with a dominant fixed stroke width value. Therefore, the similarity in stroke width can be efficiently used as a critical characteristic to describe the text region in video frame.

The contributions of this paper can be summarized as follow:

- Exploiting the similarity in stroke width characteristic of text to build an effective edge detection method with local adaptive threshold algorithm.
- Implementing a stroke-based method to localize text regions in video.
- Designing a multi-frame refinement method which can not only refine the text location but also enhance the quality of the detected text.

The rest of this paper is organized as follows: Section 2 presents the proposed method for text detection in video. To demonstrate its effectiveness, experimental results are given in Section 3. In Section 4, the concluding remarks are drawn.

2 Proposed Method

In the proposed method, text regions in video are detected through three processes. First, edge detection with local adaptive threshold algorithm is applied to reveal text edge pixels. Second, dilation morphological operator with adaptive structuring element size is exploited in the stroke-based localization process to roughly localize text regions. Finally, a multi-frame refinement process is applied to reduce false alarm, refine the location, and enhance the quality of each text region. Figure 1 shows the flow chart of the proposed system.

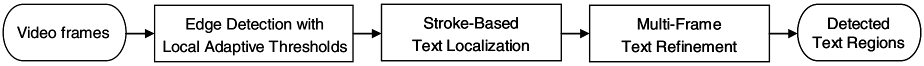

Video frames
Edge Detection with Local Adaptive Thresholds
Stroke-Based Text Localization
Multi-Frame Text Refinement
Detected Text Regions

Fig. 1. Flowchart of the proposed text detection method

2.1 Motivation

From the knowledge of font system, it turns out that, if characters are in the same font type and font size, their stroke widths are almost constant. Therefore, in the proposed method, the stroke width similarity is used as a clue to characterize text regions in frame. Generally, the width of any stroke (of both text and non-text objects) can be calculated as distance (measured in pixel) in horizontal direction between its double-edge pixels. Figure 2(a) shows an example of double-edge pixels (A and B). It can be seen from the figure that stroke widths from different characters are almost similar.

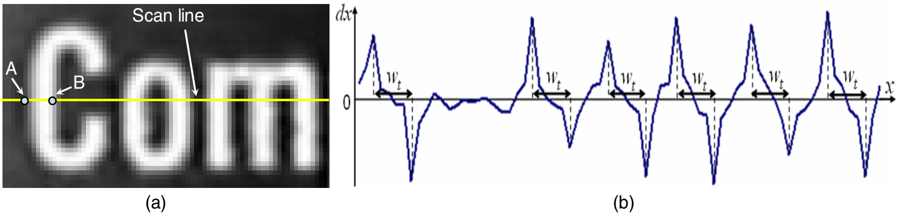


Fig. 2. An example of text image. (a) Text image. (b) Edge values for the scan line in (a), w_t is the stroke width value.

In general, the color of text often contrasts to its local background. Therefore, for any double-edge pixels of a stroke, this contrast makes an inversion in sign of the edge values, i.e. the gradient magnitude of edge pixels, in horizontal direction (*dx*) between two pixels on the left- and right-hand side of the stroke. Figure 2(b) shows the corresponding edge values in horizontal direction of a given horizontal scan line in Fig. 2(a); it is clear that the stroke can be modeled as double-edge pixels within a certain range, delimited by a positive and a negative peak nearby. By using the double-edge pixel model to describe the stroke, we can take the advantages of: 1) Reducing the effect of noise; 2) Applicability even with low-quality edge image.

2.2 Edge Detection with Local Adaptive Threshold Algorithm

First, the Canny edge detector with a low threshold is applied to video frame to keep all possible text edge pixels and each frame is divided into $M \times N$ blocks, typically 8×8 or 12×8. Second, by analyzing the similarity in stroke width corresponding to each block, blocks are classified into two types: simple blocks and complex blocks. Then, a suitable threshold algorithm for each block type is used to determine the proper threshold for each block. Finally, the final edge image is created by applying each block with the new proper threshold.

2.2.1 Block Classification

For each block, we create a stroke width set which is the collection of all stroke width candidates contained in this block. Due to the similarity in stroke width of characters, the values in the stroke width set of the text region on the simple background are concentrated on some close values. Whereas stroke width candidates of the text region on the complex background or background region may also be created by other background objects. As a result, the element values in this set may spread over a wide range of values. Therefore, text regions on a simple background can be characterized by a smaller value of the standard deviation of stroke width than those on other regions.

Based on this different characteristic, blocks in the frame are classified into two types: simple blocks and complex blocks. A block is classified as a simple one if the standard deviation of stroke width values is smaller than a given specific value. Otherwise, it will be classified as a complex one. For the simple block, the threshold of the edge detector should be relatively low to detect both low-contrast and high-contrast texts. On the contrary, the threshold for the complex block should be relatively high to eliminate background and highlight text.

2.2.2 Local Adaptive Threshold Algorithm

In each block, the stroke width value corresponding to text objects often dominates in population of the stroke width set. Therefore, it can be estimated by calculating the stroke width with the maximum stroke width histogram value. Let w_t denote the stroke width value of text, w_t can be defined as:

$$w_t = \max_{l} H(l)\,, \tag{1}$$

where $H(l)$ is the value on the block's stroke width histogram with stroke width l.

From the set of all double-edge pixels, we construct two rough sets: the text set S_t and the background set S_{bg} . The S_t represents the set of all pixels which are predicted as text edge pixels whereas the S_{bg} represents the set of all predicted background edge pixels. S_t and S_{bg} are constructed as follow:

$$S_t = \{i, j \mid i, j \in E, w(i, j) = w_t\}\,, \tag{2}$$

$$S_{bg} = \{i, j \mid i, j \in E, w(i, j) \neq w_t\}\,, \tag{3}$$

where E is the edge map of the block and $w(i, j)$ denotes the stroke width between the double-edge pixels i and j. Note that S_t and S_{bg} are only the rough sets of the text edge pixels and background edge pixels, since only edge pixels with gradient direction in horizontal are considered during the stroke width calculation process.

Thresholds for the simple block and the complex block are determined as follow:

- In the simple block case, the text lies on clear background. Therefore, the threshold is determined as the minimum edge value of all edge pixels belonging to S_t in order to keep text information and simplify the computational process.
- In the complex block case, to determine the suitable threshold for the edge detector is much more difficult. Applying general thresholding methods does not often give

a good result since these methods are used for classifying general problems, not for such a specific problem as separating text from background. In this paper, by discovering the similarity in stroke width of text, we can roughly estimate the text set and background set as S_t and S_{bg} . Therefore, the problem of finding an appropriate threshold in this case can be converted into another but easier problem of finding appropriate threshold to correctly separate the two sets: S_t and S_{bg} .

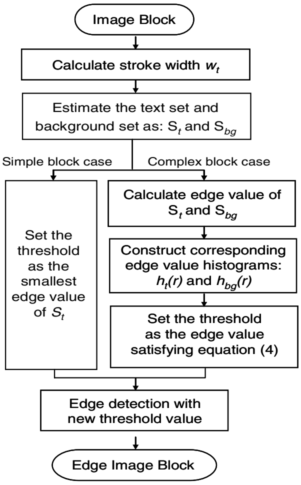


Image Block
Calculate stroke width w_t
Estimate the text set and background set as: S_t and S_{bg}
Simple block case
Complex block case
Set the threshold as the smallest edge value of S_t
Calculate edge value of S_t and S_{bg}
Construct corresponding edge value histograms: $h_t(r)$ and $h_{bg}(r)$
Set the threshold as the edge value satisfying equation (4)
Edge detection with new threshold value
Edge Image Block

Fig. 3. Flowchart of the proposed local adaptive threshold algorithm

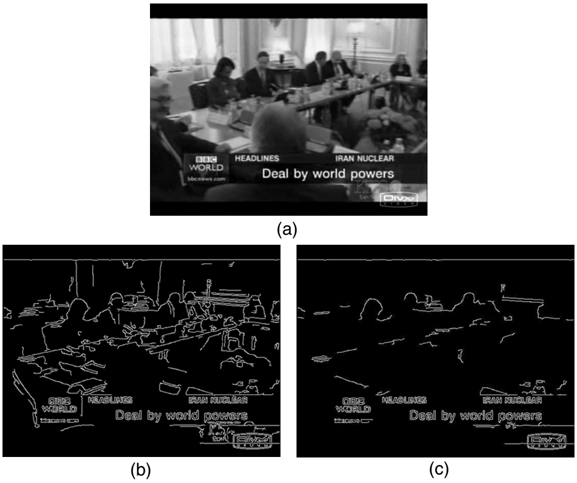


Fig. 4. Edge detection results. (a) Original Image. Edge detection using (b) constant threshold, (c) proposed local adaptive threshold algorithm.

Let r denote the edge value (gradient magnitude) of a pixel in a block, $h_t(r)$ and $h_{bg}(r)$ denote the histograms of the edge values corresponding to the text set S_t and background set S_{bg} , respectively. According to [12], if the form of the two distributions is known or assumed, it is possible to determine an optimal threshold (in term of minimum error) for segmenting the image into the two distinct sets. And the optimal threshold, denoted as *T,* can be revealed as the root of the equation:

$$p_t \times h_t(T) = p_{bg} \times h_{bg}(T), \tag{4}$$

where p_t and p_{bg} ($p_{bg} = 1 - p_t$) are the probabilities of a pixel to be in S_t and S_{bg} sets, respectively. Consequently, the appropriate threshold for the complex block is determined as the value which satisfies or approximately satisfies equation (4). Figure 3 shows flowchart of the local adaptive threshold algorithm. Figure 4 shows the results of edge detection method on video frame in Fig. 4(a) by using only one constant threshold (Fig. 4(b)), in comparison with using the proposed local adaptive thresholds (Fig. 4(c)). The pictures show that the proposed method could eliminate more background pixels while still conserves text pixels.

2.3 Stroke-Based Text Localization

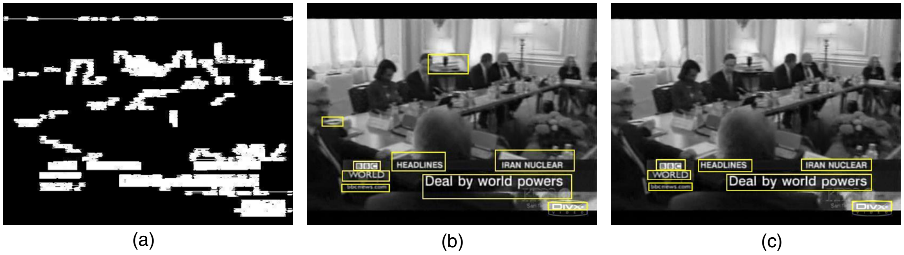
After edge detection process, dilation morphological operator is applied to the edge detected video frame for highlighting text regions. The size of the structuring element is adaptively determined by the stroke width value.

When applying the dilation operator, one of the most important factors that need to be considered is the size of the structuring element. If this size is set too small, the text area cannot be filled wholly. As the result, this area can be regarded as non-text area. In contrast, if this size is set too large, text can be mixed with the surrounding background. This problem results in increasing the number of false alarms. Moreover, using only a fixed size of the structure element, as in Chen et al.'s [9] method, is not applicable for texts of different sizes.

$$\overbrace{\begin{array}{|c|c|c|c|c|}\hline 1 & 1 & \dots & 1 & 1 \\ \hline\end{array}}^{2w_t+1}$$

Fig. 5. Structure element of the dilation operation (w_t is the stroke width value)

In this paper, we determine the size of the structure element based on the stroke width value which is already revealed in the edge detection process. More specifically, for each block, we apply a dilation operator of the size: $((2\times w_t+1)\times 1)$ at which the stroke width is w_t as shown in Fig. 5. This size is satisfactory to wholly fill the character as well as connect neighborhood characters together. Moreover, using block-based dilation with suitable structure element shape makes it applicable for text with different sizes, at different locations in video frame. Figure 6(a) shows the image using the proposed dilation operators.



(a) (b) (c)

Fig. 6. Text localization and refinement process (a) Dilated image (b) Text regions candidates (c) Text regions after being refined by multi-frame refinements

After dilation process, connected component analysis is performed to create text region candidates. Then, based on the characteristic of text, the following simple criteria for filtering out non-text regions are applied: 1) the height of the region is between 8 and 35; 2) the width of the region must be larger than the height; 3) the number of edge pixel must be two times larger than the width based on the observation that text region should have abundant edge pixels. Figure 6(b) shows the text region candidates after applying these criteria.

2.4 Multi-frame Refinement

Multi-frame integration has been used for the purpose of text verification [13], or text enhancement [14]. However, temporal information for the purpose of text refinement in frame, which often plays an important role in increasing the accuracy of text segmentation and recognition steps afterward, has not been utilized so far. In this paper, we propose a multi-frame based method to refine the location of text by further eliminating background pixels in the rough text regions detected in the previous steps. Moreover, the quality of text is also improved by selecting the most suitable frame, i.e. the frame at which text is displayed clearest, in the frame sequence. By using our method, the enhanced text region doesn't cause the blurring problem as in text enhancement of Li et al.'s method [14].

First, a multi-frame verification [13] is applied to reduce the number of false alarms. For each of *m* consecutive frames in a video sequence, a text region candidate is considered as a true text region only if existing at least *n* ($n<m$) similar text regions $T_0, T_1, \ldots, T_{n-1}$ appearing in *n* different frames. T_k (k = 0, 1...n-1) is the region of the corresponding frame received after edge detection process.

Let call *T* the stationary edge image of the corresponding text region candidate. The pixel value at location (*x, y*) of *T* is determined as follows:

$$T(x,y) = \begin{cases} edge\ \ pixel, & if\ \sum_{k=0}^{n\text{-}1} I_k(x,y) > \theta \\ non\ \ edge\ \ pixel, & otherwise, \end{cases} \tag{5}$$

where θ is a specific threshold and $I_k(x, y)$ is defined as:

$$I_k(x,y) = \begin{cases} 1, & if\ \ T_k(x,y)\ is\ edge\ pixel \\ 0, & otherwise. \end{cases} \tag{6}$$

Refer to (5), *T*(*x, y*) is an edge pixel if at the location (*x, y*), an edge pixel appears more than θ times, otherwise, *T*(*x, y*) is a non-edge pixel. In the proposed method, the θ is set equal to $[n\times 3/4]$ in order to reduce the effect of noise. Based on the stationary characteristic of text, almost all background pixels are removed in *T*. However, this integration process may also remove some text edge pixels. In order to recover the lost text edge pixels, a simple edge recovery process is performed. A pixel in *T* is marked as edge pixel if it's two neighborhoods in the horizontal, vertical, or diagonal direction are edge pixels. After the recovery process, *T* can be seen as the edge image of the true text regions. Therefore, the precise text location of the corresponding text region can be obtained by calculating the bounding box of edge pixels contained in *T*.

In order to enhance the quality of the text, we extract the most suitable frame in the frame sequence where text appears clearest. Based on the fact that a text region is clearest if the corresponding edge image contains almost text pixels, the most suitable frame is extracted if the edge image of its text region is the best matching with *T*. In other words, we choose the frame whose edge image T_k (k = 0,.., n-1) is the most similar with T. The MSE (Mean Squared Error) measurement is used to measure the similarity between two regions. The effectiveness of using multi-frame refinement is manifested in Fig. 6(c). Comparing to Fig. 6(b), two false alarms are removed and all of true text regions have more precise bounding boxes.

3 Experimental Result

Due to the lack of a standard database for the problem of text detection in video, in order to evaluate the effectiveness of the proposed method, we have collected a number of videos from various sources for a test database. Text appearance varies with different color, orientation, language, and character font size (from 8pt to 90pt). The video frame formats are 512×384 and 720×480 pixels. The test database can be divided into three main categories: news, sport, and drama. Table 1 shows the video length and the number of ground-truth text regions contained in each video category. Totally, there are 553 ground-truth text regions in the whole video test database.

Table 1. Properties of video categories

	Drama	Sport	News
Video length	15 minute	32 minute	38 minute
Text regions	126	202	225

For quantitative evaluation, the detected text region is considered as the correct one if the intersection of the detected text region (DTR) and the ground-truth text region (GTR) covers more than 90% of this DTR and 90% of this GTR. The efficiency of our detection method is assessed in terms of three measurements (which are defined in [10]): ***Speed***, ***Detection Rate***, and ***Detection Accuracy***.

In order to assess the effectiveness of the proposed method, we compare the performance of the proposed method with that of the typical edge-based method proposed by Lyu et al. [10], and the method using 3 processes: edge detection with a constant threshold, text localization with fixed size dilation operations (similar to the algorithm in [9]), and multi-frame refinement. Let call it "***constant threshold***" method.

Table 2 shows the number of correct and false DTRs for three video categories. It can be seen from the table that not only does the proposed method create the highest number of correct DTRs but it also produces the smallest number of false DTRs in every case. Our method is obviously stronger than the others even in the case of news category (the number of false DTRs is about only a half compared to other methods). It is more difficult to detect text in news video since the background is changing fast and texts have variable sizes with different contrast levels to the background. The proposed method could overcome these problems since it successfully exploits the self characteristic of text (the stroke similarity), which is invariant to the background complexity as well as different font sizes and colors of text.

Table 3 gives a summary of the detection rate and the detection accuracy of the three methods tested on the whole video test database. The proposed method achieved

Table 2. Number of correct and false DTRs

		Drama	Sport	News
Lyu et al. [10]	Correct DTRs	96	154	185
	False DTRs	16	26	38
Constant threshold	Correct DTRs	109	152	189
	False DTRs	19	32	46
Proposed Method	Correct DTRs	114	179	205
	False DTRs	11	20	21

the highest accuracy with the detection rate of 90.1% and the detection accuracy of 90.5%. This encouraging result shows that our proposed method is an effective solution to the background complexity problem of text detection in video. It can be seen from the table that the proposed method is faster than Lyu et al.'s [10] method and a bit lower than using constant threshold method which is obviously clear since we need to scan the frame with different thresholds. Moreover, the processing time of 0.18s per frame meets the requirement for real-time applications.

Figure 7 shows some more examples of the results we got. In these pictures, all the text strings are detected and their bounding boxes are relatively tight and accurate.

Table 3. Text detection accuracy

	Correct DTRs	False DTRs	Detection Rate	Detection Accuracy	Speed (Sec/frame)
Lyu et al. [10]	435	80	78.7 %	84.5 %	0.23s
Constant threshold	450	97	81.4 %	82.3 %	0.16s
Proposed Method	498	52	90.1 %	90.5 %	0.18s

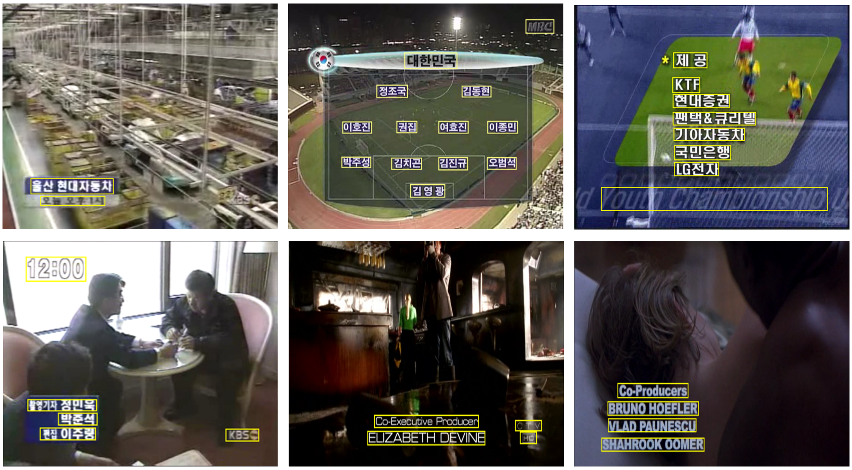


Fig. 7. Some pictures of detected text regions in frames

4 Conclusion

This paper presents a comprehensive method for text detection in video. Based on the similarity in stroke width of text, an effective edge detection method with local adaptive thresholds is applied to reduce the background complexity. The stroke width information is further utilized to determine the structure element size of the dilation operator in the text localization process. To reduce the false alarm as well as refine the text location, a new multi-frame refinement method is applied. Experimental results with a large set of videos demonstrate the efficiency of our method with the *detection rate* of 90.1% and *detection accuracy* of 90.5%. Based on these encouraging results, we plan to continue research on text tracking and recognition for a real time text-based video indexing and retrieval system.

References

1. Zhu, Q., Yeh, M.C., Cheng, K.T.: Multimodal fusion using learned text concepts for image categorization. In: Proc. of ACM Int'l. Conf. on Multimedia, pp. 211–220. ACM Press, New York (2006)
2. Lienhart, R.: Dynamic video summarization of home video. In: Proc. of SPIE, vol. 3972, pp. 378–389 (1999)
3. Fan, J., Luo, H., Elmagarmid, A.K.: Concept-oriented indexing of video databases: toward semantic sensitive retrieval and browsing. IEEE Trans. on Image Processing 13, 974–992 (2004)
4. Zhong, Y., Karu, K., Jain, A.K.: Locating text in complex color images. Pattern Recognition 28, 1523–1536 (1995)
5. Jain, A.K., Yu, B.: Automatic text location in images and video frames. In: Proc. of Int'l. Conf. on Pattern Recognition, vol. 2, pp. 1497–1499 (August 1998)
6. Ohya, J., Shio, A., Akamatsu, S.: Recognition characters in scene images. IEEE Trans. on Pattern Analysis and Machine Intelligence 16, 214–220 (1994)
7. Qiao, Y.L., Li, M., Lu, Z.M., Sun, S.H.: Gabor filter based text extraction from digital document images. In: Proc. of Int'l. Conf. on Intelligent Information Hiding and Multimedia Signal Processing, pp. 297–300 (December 2006)
8. Li, H., Doermann, D., Kia, O.: Automatic text detection and tracking in digital video. IEEE Trans. on Image Processing, 147–156 (2000)
9. Chen, D., Bourlard, H., Thiran, J.P.: Text identification in complex background using SVM. In: Proc. of Int'l. Conf. on Document Analysis and Recognition, vol. 2, pp. 621–626 (December 2001)
10. Lyu, M.R., Song, J., Cai, M.: A comprehensive method for multilingual video text detection, localization, and extraction. IEEE Trans. on Circuits Systems Video Technology, 243–255 (2005)
11. Jung, K.C., Han, J.H., Kim, K.I., Park, S.H.: Support vector machines for text location in news video images. In: Proc. of Int'l. Conf. on System Technology, pp. 176–189 (September 2000)
12. Gonzalez, R.-C., Woods, R.E.: Digital Image Processing, 2nd edn., pp. 602–608. Prentice-Hall, Englewood Cliffs (2002)
13. Lienhart, R., Wernicke, A.: Localizing and segmenting text in images and videos. IEEE Trans. on Circuits Systems Video Technology, 256–268 (2002)
14. Li, H., Doermann, D.: Text enhancement in digital video using multiple frame integration. In: Proc. of ACM Int'l. Conf. on Multimedia, pp. 19–22. ACM Press, New York (1999)