Overlay Text Retrieval From Video Scene

K.Manohar, S.Irfan, K.Sravani
*Lecturer Dept. Of Electrical And Computer Engg. Wollo University
**Lecturer Dept. Of Electrical And Computer Engg. Wollo University
***Assistant Professor Dept. Of Computer Science And Engg. Swarna Bharathi College Of Engineering

Abstract

The rapid growth of video data leads to an urgent demand for efficient and true content-based browsing and retrieving systems. In response to such needs, various video content analysis schemes using one or a combination of image, audio, and text information in videos have been proposed to parse, index, or abstract massive amount of data text in video is a very compact and accurate clue for video indexing and summarization. Most video text detection and extraction methods hold assumptions on text color, background contrast, and font style. Moreover, few methods can handle multilingual text well since different languages may have quite different appearances. In this paper, an efficient overlay text detection and extraction method is implemented which deals with complex backgrounds. Based on our observation that there exist transient colors between inserted text and its adjacent background. It is robust with respect to font size, style text, color, orientation and noise and can be used in a large variety of application fields such as mobile robot navigation vehicle license detection and recognition, object identification , document retrieving, etc.

Index Terms— overlay text , video content analysis, video indexing summarization.

1. Introduction

One powerful indexing cue for retrieval is the text appearing in videos. There are two kinds of text in video scene text and superimposed text. Scene text is relatively less important for video indexing since it appears unpredictably and seldom intended. however superimposed text usually provides important information and its appearance is carefully designed, thus recognition of superimposed text,the process known as OCR (optical character recognition),can help in video retrieval .the main problem in video OCR lies in two aspects one is complexity of background ,and the other is the low resolution of text .

The existing algorithms can be classified into three categories :connected component based ,texture based and edge-based methods the connected component based method assumes that the pixels belonging to the same connected region share some common features such as color or gray intensity extracted text as those connected components of monotonous color that meet certain size and horizontal alignment constraints. However ,these methods are unsuitable for videos whose text are embedded in complex background .the texture methods treat the text region as a special type of texture and conventional texture classification methods to extract text .these methods are robust than connected component based methods in dealing with complex backgrounds another type of method is edge based method and this method is based on the observation that text regions have abundant edges the commonly adopted method is to apply an edge detector to video frame and then identify regions with high edge density and strength and it becomes less reliable as scene contains more edges in its background In this paper, we propose an efficient overlay text detection and extraction method using the transition region between the overlay text and background. First, we generate the transition map based on our observation that there exist transient colors between overlay text and its adjacent background. Then the overlay text regions are roughly detected by computing the density of transition pixels and the consistency of texture around the transition pixels. The detected overlay text regions are localized accurately using the projection of transition map with an improved color-based thresholding method [4] to extract text strings correctly. The rest of this paper is organized as follows. We generate the transition map and refine the detected text regions in Section II. The overlay text extraction from the refined text regions is explained in Section III.

The proposed method is based on our observations that there exist transient colors between overlay text and its adjacent background (see Fig. 1) and overlay texts have high saturation because they are inserted by using graphic components. The overall procedure of proposed overlay text detection method is shown in Fig. 2, where each module is explained in Sections II-1–5. The overlay text extraction method will be clearly explained in Section III.

Fig. 1. Examples of overly text

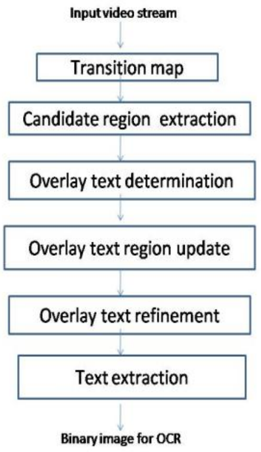

Fig. 2. Overall procedure of the proposed detection method

2. Transition Map and Text Region

2.1. Generation of transition map

As a rule of thumb, if the background of overlay text is dark then the overlay text tends to be bright. On the contrary, the overlay text tends to be dark if the background of overlay text is bright. Therefore, there exists transient colors between overlay text and its adjacent background due to color bleeding, the intensities at the boundary of overlay text are observed to have the logarithmical change As shown in Fig. 3, the intensities of three consecutive pixels are decreasing logarithmically at the boundary of bright overlay text due to color bleeding by the lossy video compression. It is also observed that the intensities of three consecutive pixels increases exponentially at the boundary of dark overlay text. Since the change of intensity at the boundary of overlay text may be small in the low contrast image, to effectively determine whether a pixel is within a transition region [8], the modified saturation is first introduced as a weight value based on the fact that overlay text is in the form of overlay graphics. The modified saturation is defined as follows:

$$S(x,y) = 1 - \frac{3}{(R+G+B)[\min(R,G,B)]} \qquad (1)$$

$$\ddot{S}(x,y) = \frac{S(x,y)}{\max(S(x,y))}$$

$$\text{where } \max(S(x,y)) = \begin{cases} 2\times(0.5-\ddot{I}(x,y)), & \text{if } \ddot{I}(x,y) > 0.5 \\ 2\times\ddot{I}(x,y), & \text{otherwise.} \end{cases} \qquad (2)$$

*S(x,y)*and max(*S(x,y)*) denote the saturation value and

the maximum saturation value at the corresponding intensity

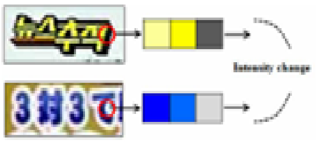

Fig.3. Change of intensities in the transition region.

level, respectively. *I(x,y)* denotes the intensity at the ,(x,y) which is normalized to [0,1]. Based on the conical HSI color model, the maximum value of saturation is normalized in accordance with *I(x,y)* compared to 0.5 in (2). The transition can thus be defined by combination of the change of intensity and the modified saturation as follows:

$$D_L(x,y) = (1+dS_L(x,y)) \times\ I(x-1,y) - I(x,y)|$$

$$D_H(x,y) = (1+dS_H(x,y)) \times |I(x,y) - I(x+1,y)$$

$$\text{where } dS_L(x,y) = |\ddot{S}(x-1,y) - \ddot{S}(x,y) \text{ and}$$

$$dS_H(x,y) = |\ddot{S}(x,y) - \ddot{S}(x+1,y)\,. \qquad (3)$$

Since the weight dS_L(x,y) and dS_H(x,y) can be zero by the achromatic overlay text and background, we add 1 to the weight in (3). If a pixel satisfies the logarithmical change constraint given in (4), three consecutive pixels centered by the current pixel are detected as the transition pixels and the transition map is generated

$$T(x,y) = \begin{cases} 1, & \text{if } D_H > D_L + TH \\ 0, & \text{otherwise.} \end{cases} \qquad (4)$$

The thresholding value *TH* is empirically set to 80 in consideration of the logarithmical change

2. retreival of Candidate Region.

The transition map can be utilized as a useful indicator for the overlay text region. To generate the connected omponents, we first generate a linked map as shown in Fig. 5(b). If a gap of consecutive pixels between two nonzero points in the same row is shorter than 5% of the image width, they are filled with 1s. If the connected components are smaller than the threshold value, they are removed. The threshold value is empirically selected by observing the minimum size of overlay text region. Then each connected component is reshaped to have smooth boundaries. Since it is reasonable to assume that the overlay text regions are generally in rectangular shapes, a rectangular bounding box is generated by linking four points, which correspond to (min_x,min_y),(max_x,min_y),(min_x,max_y),(max_x,max_y) taken from the link

2.2. Overlay text region determination

The next step is to determine the real overlay text region among the boundary smoothed candidate regions by some useful clues, such as the aspect ratio of overlay text region. Since most of overlay texts are placed horizontally in the video, the vertically longer candidates can be easily eliminated. The density of transition pixels is a good criterion as well. Nevertheless, a more refined algorithm is needed to minimize the false detection due to the complex background. In this subsection, we introduce a texture-based approach for overlay text region determination Based on the observation that intensity variation around the transition pixel is big due to complex structure of the overlay text, we employ the local binary pattern (LBP) introduced in

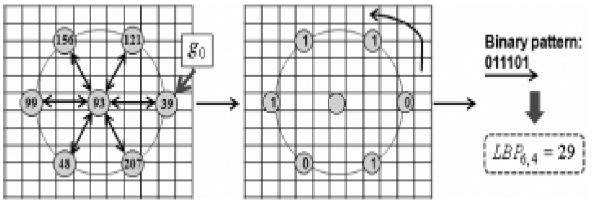

Fig. 4. Example of LBP computation.

to describe the texture around the transition pixel. LBP is a very efficient and simple tool to represent the consistency of texture using only the intensity pattern. LBP forms the binary pattern using current pixel and its all circular neighbor pixels and can be converted into a decimal number as follows:

$$\mathrm{LBP}_{P,R} = \sum_{i=0}^{P-1} s(g_i - g_c)2^i, \text{ where } s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0. \end{cases} \quad (5)$$

P and R denote the user's chosen number of circular neighbor pixels of a specific pixel and the radius of circle, respectively. and denote the intensity of current pixel and circular neighbor pixels, respectively. We can obtain the binary pattern as shown in Fig. 4, and the resulting $\mathrm{LBP}_{6,4}$=29(=$2^4+2^3+2^1+2^0$). Now we define the probability of overlay text (POT) using the operator as follows: The LBP operator is first applied to every transition pixel in each candidate region. Then, we compute the number of different LBPs to consider the intensity variation around the transition pixel. Since we use the 8 neighbor pixels to obtain the LBP value, the total number of potentially different LBPs is 2^8=256 . Although the number of different LBPs is generally increasing as the candidate region includes more transition pixels, it may not be guaranteed since transition pixels can have same local binary pattern. Let ω_i denote the density of transition pixels in each candidate region and can be easily obtained from dividing the number of transition pixels by the size of each candidate region. POT is defined as follows:

$$\mathrm{POT}_i = \omega_i \times \mathrm{NOL}_i, \quad i = 1, \ldots, N \qquad (6)$$

Where N denotes the number of candidate regions as mentioned. NOL_i denotes the number of different LBPs, which is normalized by the maximum of the number of different LBPs (i.e., 256) in each candidate region. If POT of the candidate region is larger than a predefined value, the corresponding region is finally determined as the overlay text region.

2.3. Overlay text region refinement

The overlay text region or the bounding box obtained in the preceding subsection needs to be refined for better accurate text extraction, which will be addressed in Section III. In this subsection, we use a modified projection of transition pixels in the transition map [4] to perform the overlay text region refinement. First, the horizontal projection is performed to accumulate all the transition pixel counts in each row of the detected overlay text region to form a histogram of the number of transition pixels. Then the null points, which denote the pixel row without transition pixels, are removed and separated regions are re-labeled The projection is conducted vertically and null points are removed once again. Compared to the coarse-to-fine projection proposed for edge-based scheme in [4], our projection method is applied to the detected overlay text regions only, making the process simpler.

2.4. Overlay text region update

Once the overlay text regions are detected in the current frame, it is reasonable to take advantage of continuity of overlay text between consecutive frames for the text region detection of the next frame. If the difference, which can be obtained by XOR of current transition map and previous transition map, is smaller than a predefined value, the overlay text regions of previous frame are directly applied as the detection result without further refinement. In order to deal with such changes, we compare the current transition map with the transition map obtained 3 frames earlier and the dissimilarity measure between these maps is defined a

$$d(T_n, T_{n-3}) = \sum_{(x,y)\in T} (T_n(x,y) \otimes T_{n-3}(x,y)) \qquad (7)$$

$$\text{if}(d(T_n, T_{n-3}) < th) TR_n = TR_{n-3}$$
$$\text{otherwise, find new } TR_n \qquad (8)$$

Where T_n and T_{n-3} denote the transition map obtained from The nth frame and the (n-3)th frame, respectively. TR_n andTR_{n-3} denote the detected overlay text regions in the nth frame and the (n-3)th frame, respectively. $\otimes$ denotes the XOR operator. In other words, if the values on the nth frame and the (n-3)th frame transition map are same, the result of $\otimes$ between two values is set to be 0. Otherwise, the result of $\otimes$ between two values is set to be 1. The overlay text region update method can reduce the processing time efficiently.

3. OVERLAY TEXT EXTRACTION

Before applying video OCR application, the refined overlay text regions need to be converted to a binary image, where all pixels belonging to overlay text are highlighted and others suppressed. Since the text color may be either brighter or darker than the background color, an efficient scheme is required to extract the overlay text dealing with complex backgrounds and various text appearances. Among several algorithms proposed to conduct such a task [4], [6], [10], there is one effective method, proposed by Lyu *et al.* [4], consists of color polarity classification and color inversion (if necessary), followed by adaptive thresholding, dam point labeling and inward filling. In this section, we propose a fast and efficient overlay text extraction technique, which is based on Lyu's approach with some modifications for better performance. Fig. 5 shows the overall procedure of extraction method

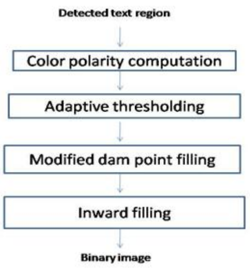

Fig. 5 the overall procedure of extraction method

A. Color Polarity Computation

Fig. 6 shows two opposite scenarios, in which either the overlay text is darker than the surrounding background

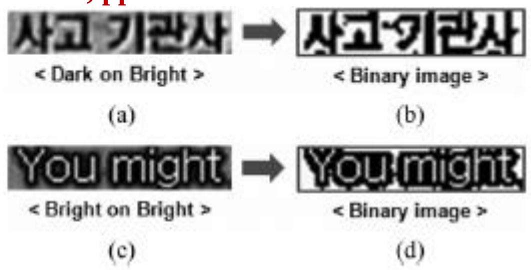

Fig. 6. Examples of binarized image by the average intensity of overlay text
region.

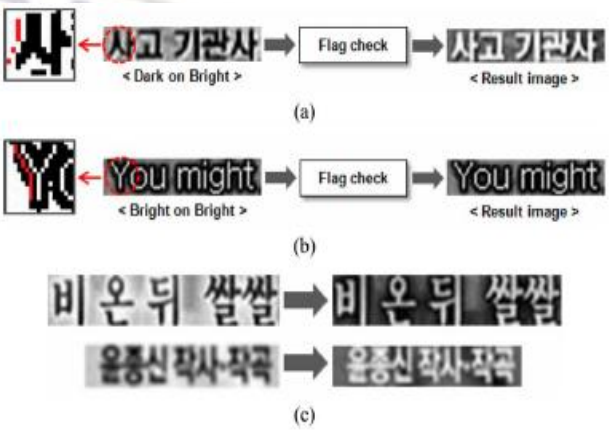

Fig. 7. Process of inversing image by the color polarity. (a) Dark text on bright background. (b) Bright text on bright background. (c) Examples of inverted overlay text by "bright_text_flag".

[Fig. 6(a)], or the text is brighter than its neighbors [Fig. 6(c)]. As shown in Fig. 6(b) and (d), the binarized images obtained by simple thresholding represent the overlay text as either 1 (or " *White*") or 0 (or "*Black*"), respectively. Such inconsistent results must complicate the following text extraction steps Thus, our goal in this subsection is to check the color polarity and inverse the pixel intensities if needed so that the output text
region of the module can always contain bright text compared to its surrounding pixels as shown in Fig. 7.

We observe that this goal can be simply attained owing to the transition map obtained in the preceding section. First of all, the binary image obtained by thresholding with average intensity value can be effectively utilized [see Fig. 6(b) and (d)]. Given the binarized text region, the boundary pixels, which belong to left, right, top, and bottom lines of the text region, are searched and the number of white pixels is counted. If the number of white boundary pixels is less than 50% of the number of boundary pixels, the text region is regarded as "bright text on dark background" scenario, which requires no polarity change In other words, the overlay text is always bright in such scenarios.

If the number of white pixels is greater than that of black pixels, we conduct a task to turn on or off the "bright_text_flag"

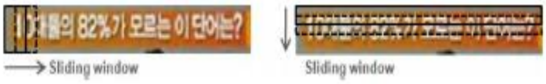

Fig. 8. Process of adaptive thresholding adopted from [7]. Horizontal adaptive thresholding (left). Vertical adaptive thresholding (right).as expressed in (9), shown at the bottom of the page, where denotes the position of the first encountered transition pixel in each row of the text region and denotes the value on the binary image. As shown in Fig. 7, the flag is set to 1 for Fig. 7(a) since the first encountered transition pixel belongs to 1, whereas the pixel apart by two pixel distance belongs to 0 If such case happens at least once, the pixel values in the text region is inverted to make the overlay text brighter than the surrounding background. Note that the inversion is simply done by subtracting the pixel value from the maximum pixel value. The process of color polarity computation is shown in Fig. 7. The first transition pixels in each row on the binary image are represented by red color in Fig. 7. Examples with "bright_flag_text"
are also shown in Fig. 7(c)

$$\text{bright\_text\_flag} = \begin{cases} 1, & \text{if } I_B(x_F, y_F) = 1 \text{ and } I_B(x_F + 2, y_F) = 0 \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Since it is confirmed that the overlay text is always bright in each text region, it is safe to employ Lyu's method to extract characters from each overlay text region. First, each overlay text region is expanded wider by two pixels to utilize the continuity of background. This expanded outer region is denoted as *ER*. Then, the pixels inside the text region are compared to the pixels in *ER* so that pixels connected to the expanded region can be excluded We denote the text region as *TR* and the expanded text region as *ETR*. Next, sliding-window based adaptive thresholding is performed in the horizontal and the vertical directions with different window sizes, respectively. Compared to the Lyu's method, the height of expanded text region is not normalized in our method. Let and*ETR(x,y)* and *B(x,y)*denote gray scale pixels on *ETR* and the resulting binary image, respectively. All *B(x,y)* are initialized as "*White*" The window with the size of 16x*TER_hieght* is moving horizontally with the stepping size 8 and then the window with the size of(*ETR_width*)x(*ETR*_hieght/4) is moving vertically with the stepping size(*ETR*_hieght/8) . If the intensity *ETR(x,y)* of is smaller than the local thresholding value computed by Otsu method in each window the corresponding *B(x,y)*is set to be "*Black*". The process of applying the sliding windows for adaptive thresholding is shown in Fig. 8 Authors of [7] assume that the background pixels in *TR* are generally connected to *ER* in terms of intensity. They use filling from *ER* to the connected pixels in *TR* to remove the background pixels. However, since the text pixels might be connected to the background pixels in *TR*, the unwanted removal can occur when the filling task is conducted. Thus, the "dam points" inside *TR* is defined to prevent the filling from flooding into text pixels. We modify the dam point definition introduced in [7] by adding a simple constraint for transition pixels to improve extraction performance as follows:

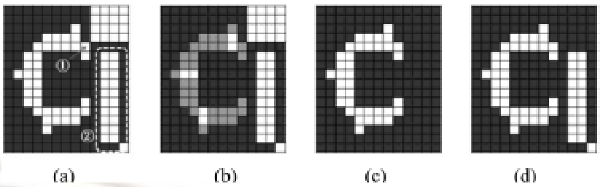

Fig. 9. Text extraction procedure. (a) Adaptive thresholding. (b) Modified dam points labeled in gray. (c) Inward filling. (d) Inward filling when the original Lyu's method is used.

$$\begin{aligned}\text{Dam points} = \{(x,y) | (B(x,y) = \text{“}White\text{”} \\ \wedge N_T(x,y) \neq 0) \\ \wedge (\text{MIN\_}W \leq \min[H_{\text{len}}(x,y) V_{\text{len}}(x,y)] \\ \leq \text{MAX\_}W)\}\end{aligned} \quad (10)$$

whereMIN_W=1,MAX_W=int(*ER_hieght/ 8*) and N_T denotes the number of transition pixels among the horizontally connected pixels with(x,y) .H_{len} and V_{len} denote the length of the connectivity with horizontal and vertical direction at the (x,y), respectively. For example H_{len}, is 6 and V_{len} is 2 on the pixel in Fig. 9(a). Since the height of *ETR* is 16 in the figure MAX_W is set to be 2 by (10). The minimum of these values is 2, which belongs to the range defined in (10). Therefore, the pixel marked as is labeled as a dam point, which is represented in gray in Fig. 9(b). Finally, we can obtain characters correctly from each overlay text region by the inward filling as addressed in [4]. If a pixel is "*White*" during the scanning of binarized pixels in *ER*, all the connected "*White*" pixels including the pixel itself are filled with "*Black*". After the inward filling, all non-"*Black*" pixels are set to be "*White*". The result of inward filling is shown in Fig. 9(c . We see that the background of text is well removed. Note that the condition for the number of transition pixels is added in this paper. If the constraint using transition pixels is not added in the dam point labeling, background region is also labeled as dam points as shown in Fig. 9(d)

4. Experimental Results

In this section, the system is constructed on the PC platform and the development environment is MATLAB 7.5 .The video sequences are taken from internet and movies .Experiment results are shown in Fig 10 and Fig 11.

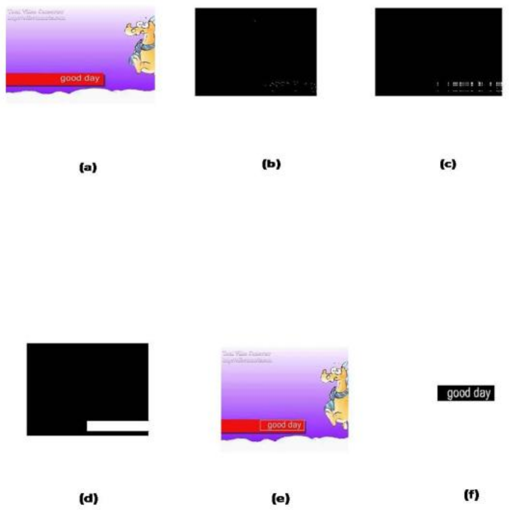

Fig 10. (a) Original image (b) Transition map(c) linked map (d) candidate region (e) text region determination (f) binary text image

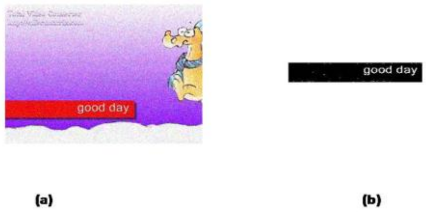

Fig 11. (a) image with a Gaussian noise (b) extracted text

The framework for evaluating performance has been implemented with the image size of 240 X 240 . Since the *TH* value in Section II-1 plays an important role to generate a robust transition map, it is carefully set to 80 The minimum size of overlay text region used to remove small components in Section II-2 is set to be 300. The parameters, such as window size for adaptive thresholding, the minimum size of overlay text region, and the threshold value for the overlay text region update, can be consistently set according to the image width or height The experiments were performed on the low-end PC (Core2Duo 2.1 GHz).

5. Conclusion

An efficient method for overlay text detection and extraction from complex videos is implemented in this paper. Our detection method is based on the observation that there exist transient colors between inserted text and its adjacent background. The transition map is first generated based on logarithmical change of intensity and modified saturation. Linked maps are generated to make connected components for each candidate region and then each connected component is reshaped to have smooth boundaries. We compute the density of transition pixels and the consistency of texture around the transition pixels to distinguish the overlay text regions from other candidate regions. The local binary pattern is used for the intensity variation around the transition pixel in the proposed method. The boundaries of the detected overlay text regions are localized accurately using the projection of overlay text pixels in the transition map. Overlay text region update between frames is also exploited to reduce the processing time. Based on the results of overlay text detection, the overlay texts are extracted based on Lyu's extraction method. We add a simple constraint based on detected transition pixels in the transition map to improve the extraction performance.. The proposed method is very useful even in the presence of noise for the real-time application.

References

[1] C. G. M. Snoek and M. Worring, "Time interval maximum entropy based event indexing in soccer video," in *Proc. Int. Conf. Multimedia and Expo*, Jul. 2003, vol. 3, pp. 481–484.

[2] J. Gllavata, R. Ewerth, and B. Freisleben, "Text detection in images based on unsupervised classification of high-frequency wavelet coefficients," in *Proc. Int. Conf. Pattern Recognition*, Aug. 2004, vol. 1, pp. 425–428.

[3] X. S. Hua, P. Yin, and H. J. Zhang, "Efficient video text recognition using multiple frame integration," in *Proc. Int. Conf. Image Processing*, Sep. 2004, vol. 2, pp. 22–25.

[4] M. R. Lyu, J. Song, and M. Cai, "A comprehensive method for multilingual video text detection, localization, and extraction," *IEEE Trans. Circuit and Systems for Video Technology*, vol. 15, no. 2, pp. 243–255 Feb. 2005.

[5] X. Liu and J. Samarabandu, "Multiscale edge-based text extraction from complex images," in *Proc. Int. Conf. Multimedia and Expo (ICME)*, Jul. 2006, pp. 1721–1724.

[6] T. Sato, T. Kanade, E. K. Hughes, and M. A. Smith, "Video OCR fo digital news archive," in *Proc. IEEE International Workshop on Content- Based Access of*

Image and Video Libraries, Jan. 1998, pp. 52–60.

[7] J. Wu, S. Qu, Q. Zhuo, and W. Wang, "Automatic text detection in complex color image," in *Proc. Int. Conf. Machine Learning and Cybernetics*, Nov. 2002, vol. 3, pp. 1167–1171.

[8] Wonjun Kim and changrick Kim . "A new approach for overlay text detection and extraction from complex video scene" in *IEEE trans on image processing.*

[9] Y. Liu, H. Lu, X. Xue, and Y. P. Tan, "Effective video text detection using line features," in *Proc. Int. Conf. Control, Automation, Robotics and Vision*, Dec. 2004, vol. 2, pp. 1528–1532.

[10] T. Ojala, M. Pierikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[11] S. Antani, D. Crandall, and R. Kasturi, "Robust extraction of text in video," in *Proc. Int. Conf. Pattern Recognition*, Sep. 2000, vol. 1, pp. 831–834.