Morphology Based Text Detection and Extraction from Complex Video Scene

T.Pratheeba , Dr.V.Kavitha , S.Raja Rajeswari

Computer Science Department, Anna University Tirunelveli
Tirunelveli, India

pratheeba32@gmail.com
s.rajaraji@gmail.com

***Abstract*—Text in video provides brief and important content information which is helpful to video scene understanding, annotation and searching. Most of the previous approaches to extracting text from videos are based on low-level features, such as edge, color, and texture information. However, existing methods experience difficulties in handling texts with various contrasts or inserted in a complex background. In this paper, we propose a novel framework to detect and extract the text from the video scene. A morphological binary map is generated by calculating difference between the closing image and the opening image. Then candidate regions are connected by using a morphological dilation operation and the text regions are determined based on the occurrence of text in each candidate. The detected text regions are localized accurately using the projection of text pixels in the morphological binary map and the text extraction is finally conducted. The proposed method is robust to different character size, position, contrast, and color. It is also language independent. Text region update between frames is also employed to reduce the processing time. Experiments are performed on diverse videos to confirm the efficiency of the proposed method.**

I. INTRODUCTION

Images and videos on webs and in databases are increasing. Broadcasters are demonstrating interest in building large digital archives of their assets for reuse of archive materials for TV programs, on-line availability to other companies and the general public. To satisfy this request there is need of systems that are able to provide efficient indexing and retrieval by content of video segments based on the extraction of content level information associated with visual data. While effective content-based retrieval of visual information of images is accomplished by supporting content representation through low-level image features, the same does not apply to content-based retrieval of videos, except for very limited application contexts. Instead, effective retrieval of videos must be based on high-level content descriptors [1].

Most broadcasting videos tend to increase the use of text to convey more direct summary of semantics and deliver better viewing experience. For example, headlines summarize the reports in news videos and subtitles in the documentary drama help viewers understand the content. Sports videos also contain text describing the scores and team or player names [2]. In general, text displayed in the videos can be classified into scene text and overlay text [3]. Scene text occurs naturally in the background as a part of the scene, such as the advertising boards, banners, and so on. In contrast to that, overlay text is superimposed on the video scene and used to help viewers' understanding. Since the overlay text is highly compact and structured, it can be used for video indexing and retrieval [4]. However, text extraction for video optical character recognition (OCR) becomes more challenging, compared to the text extraction for OCR tasks of document images, due to the numerous difficulties resulting from complex background, unknown text color, size and so on.

The rest of this paper is organized as follows. Section II reviews the related work. We generate the morphological binary map and refine the detected text regions in Section III. The text extraction from the refined text regions is explained in Section IV. The experimental results on various videos are shown in Section V, followed by conclusion in Section VI.

II. RELATED WORK

Most of existing video text detection methods have been proposed on the basis of color, edge, and texture-based feature. Color-based approaches assume that the video text is composed of a uniform color. In the approach by Agnihotri and Dimitrova [5] detect and binarizes horizontal white, yellow, and black caption text in video frames. After pre-processing, edge pixels are found using an edge detector with a fixed threshold. Frame regions with very high edge density are considered too noisy for text extraction and are discarded.

Connected component analysis is performed on the edge pixels of remaining regions. Edge components are merged based on spatial heuristics to localize text regions. Binarization is performed by thresholding at the average pixel value of each localized text region. Kim *et al.* [6] cluster colors based on Euclidean distance in the RGB space and use 64 clustered color channels for text detection. However, it is rarely true that the video text consists of a uniform color due to degradation resulting from compression coding and low contrast between text and background.

Edge-based approaches are also considered useful for video text detection since text regions contain rich edge information. The commonly adopted method is to apply an edge detector to the video frame and then identify regions with high edge density and strength. This method performs well if there is no complex background and it becomes less reliable as the scene contains more edges in the background. Lyu *et al.* [7] use a modified edge map with strength for text region detection and localize the detected text regions using coarse-to-fine projection. They also extract text strings based on local thresholding and inward filling generality. Xi et al. [8] propose an edge based method based on an edge map created by Sobel operator followed by smoothing filters, morphological operations and geometrical constraints.

Texture-based approaches, such as the salient point detection and the wavelet transform, have also been used to detect the text regions. Bertini *et al.* [9] detect corner points from the video scene and then detect the text region using similarity of corner points between frames. Zhong et al.[10] detect text in JPEG/MPEG compressed domain using texture features from DCT coefficients. They first detect blocks of high horizontal spatial intensity variation as text candidates, and then refine these candidates into regions by spatial constraints. The potential caption text regions are verified by the vertical spectrum energy. But its robustness in complex background may not be satisfying for the limitation of spatial domain features.

After the text detection step, the text extraction step should be employed before OCR is applied.The text extraction methods can be classified into color-based [11] and stroke-based methods [12], since color of text is generally different from that of background, text strings can be extracted by thresholding. Otsu [11] is a widely used color-based text extraction method due to its simplicity and efficiency of the algorithm. However, Otsu method is not robust to text extraction with similar color of background due to the use of global thresholding. To solve this problem, the detected text regions are divided into several blocks and then Otsu method is applied locally to each block, such as the adaptive thresholding introduced in [7], where a dam point is defined to extract text strings from background. On the other hand, some filters based on the direction of strokes have also been used to extract text in the stroke-based methods. The four-direction character extraction filters [12] are used to enhance the stroke-like shapes and to suppress others. However, since the stroke filter is language-dependent, some characters without obvious stripe shape can also be suppressed.

In this paper, we propose a new text detection and extraction method using the transition region between the text and background. First, we generate the morphological binary map based on our observation that there exist transient colors between text and its adjacent background. Then the text regions are roughly detected by computing the density of transition pixels and the consistency of texture around the transition pixels. The detected text regions are localized accurately using the projection of morphological binary map with an improved color-based thresholding method [7] to extract text strings correctly.

III. Text Region Detection

The proposed method is based on our observations that there exist contrast colors between text and its adjacent background. The relative contrast between texts and their background is an important feature for text region detection. The overall procedure of proposed text detection method is shown in Fig. 1.The text extraction method will be clearly explained in Section IV.

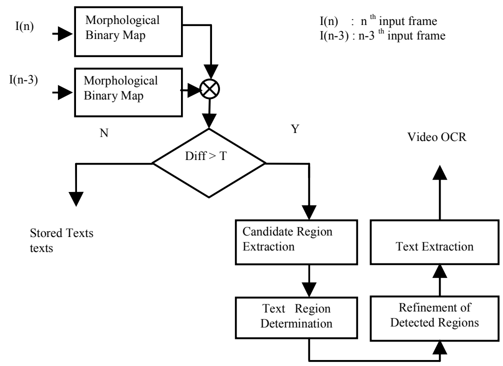

Fig. 1. Overall procedure of the proposed detection method.

A. *Morphological Binary Map*

In order to detect text regions from complex background a morphology based approach is used to extract high-contrast feature [13].

Let $I(x,y)$ denote a gray-level input image. Let $S_{m,n}$ denote a structuring element with size m×n. where m,n are odds and larger than zero. Besides, let $\oplus$ denote a dilation operation, and $\ominus$ denote an erosion operation.

Closing Operation:

$$I(x,y) \bullet S_{m,n}=(I(x,y)\oplus S_{m,n})\ominus S_{m,n} \quad (1)$$

Opening Operation:

$$I(x,y) \circ S_{m,n}=(I(x,y)\ominus S_{m,n})\oplus S_{m,n} \quad (2)$$

Difference:

$$D(I_1,I_2)=|I_1(x,y)-I_2(x,y)| \quad (3)$$

Thresholding:

$$T(I(x,y))=\begin{cases} 255, \text{ if } I(x,y) > T \\ 0 \ , \ otherwise \end{cases} \quad (4)$$

To obtain the morphological binary map, closing (1) and opening (2) operations are performed using a disk structural element $S_{3,3}$. The difference (3) obtained from subtracting both images are the result of the following step. Then, a threshold procedure (4) is applied followed by a labeling process to extract the text segments. In the threshold procedure a parameter T is defined dynamically according to the background of the image. This parameter is responsible to determine the limit value of the binarization operation.

The whole procedure of our morphology-based technique to extract the contrast features is shown in Fig 2.

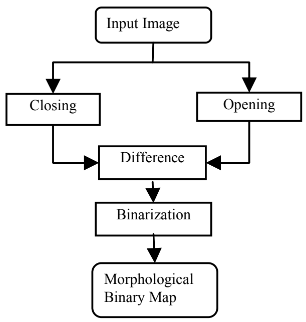

Fig.2. Flowchart of the proposed method to extract contrast features for text region detection.

An example of the result of this process is shown in Fig.3(b).

Fig.3. Generation of morphological binary map (a) Input image (b) morphological binary map

B. Candidate Region Extraction

A morphological dilation operator can easily connect the very close regions together while leaving those whose positions are far away to each other isolated. In our proposed method, we use a morphological dilation operator [14] with a 7× 7 square structuring element to the binary image obtained from the previous step to get joint areas referred to as text blobs. Fig.4 (a) shows the result of feature clustering. 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 text region. Then each connected component is reshaped to have smooth boundaries. Since it is reasonable to assume that the 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 text blobs. The refined candidate regions are shown in Fig. 4(b).

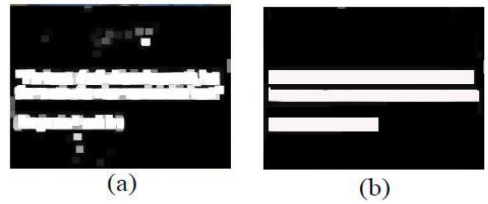

Fig.4. Extraction of candidate regions (a) Connected components through dilation (b) Smoothed candidate regions

C. Text Region Determination

The next step is to determine the real text region among the boundary smoothed candidate regions by some useful clues, such as the aspect ratio of text region. Since most of texts are placed horizontally in the video, the vertically longer candidates can be easily eliminated. Based on the observation that intensity variation around the transition pixel is big due to complex structure of the text, we employ the dominant local binary pattern (DLBP) introduced in [15] to describe the texture around the transition pixel. DLBP effectively capture the dominating patterns in texture images. Unlike the conventional LBP approach, which only exploits the uniform LBP [16], given a texture image, the DLBP approach computes the occurrence frequencies of all rotation invariant patterns defined in the LBP groups. These patterns are then sorted in descending order. The first several most frequently occurring patterns should contain dominating patterns in the image and, therefore, are the dominant patterns.

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:

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

Where, P and R denote the user's chosen number of circular neighbor pixels of a specific pixel and the radius of circle, respectively. g_c and g_i denote the intensity of current pixel and circular neighbor pixels, respectively.

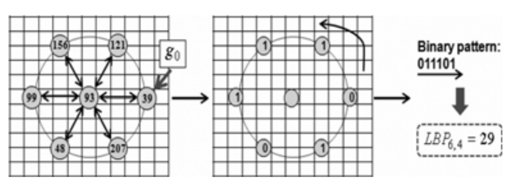

Fig.5. Example of LBP computation

We can obtain the binary pattern as shown in Fig. 5, and the resulting$\mathrm{LBP}_{6,4}$=29(=$2^4+2^3+2^2+2^0$).

DLBP consider the most frequently occurred patterns in an image. It is shown that the DLBP approach is more reliable to represent the dominating pattern information in the images.

It avoids the aforementioned problems encountered by merely using the uniform LBPs or making use of all the possible patterns, as the DLBPs are defined to be the most frequently occurred patterns.

The pseudo codes on determining the number of dominant patterns of DLBP and extracting DLBP feature vectors are presented in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1 Determining the number of dominant patterns of DLBP
Input: Input image, and the parameters P and R for DLBP
Output: The required number of patterns for 15% pattern occurrences
1. Initialize K_{temp}=0.
2. FOR each Candidate region *I* in the image
3. Initialize the pattern histogram, $H[0\ldots(2^m-1)]=0$.
4. FOR each center pixel $g_c \in I$
5. Compute the pattern label of g_c, $LBP_{P,R}$ (1)
6. Increase the corresponding bin by 1, $H[LBP_{P,R}]$++
7. END FOR
8. Sort the histogram in descending order
9. Find the number of patterns *k* for 15% pattern occurrences in *I*.

$$K = \arg\min_k \left(\frac{\sum_{i=0}^{k-1} H[i]}{\sum_{i=0}^{(2m-1)} H[i]} \right) \geq 15\,\%$$

10. K_{temp}+=k
11. END FOR
12. $K_{15\%}$=Number of different Dominant Patterns having occurences more than 15%.
13. Return $K_{15\%}$.

Algorithm 2 Extracting a DLBP feature vector
Input: Input image, the required number of dominant patterns $K_{15\%}$, and the parameters P and R for DLBP
Output: The DLBP feature vector corresponding to image *I*
1. Initialize the pattern histogram, $H[0\ldots(2^m-1)]=0$.
2. FOR each center pixel $g_c \in I$
3. Compute the pattern label of g_c, $LBP_{P,R}$ (1)
4. Increase the corresponding bin by 1, $H[LBP_{P,R}]$++
5. END FOR
6. Sort the histogram in descending order
7. Return $H[0\ldots(K_{15\%}-1)]$ as the feature vector of DLBP.

Now we define the probability of text (POT) using the operator as follows: The LBP operator is first applied to every transition pixel in each candidate region. We use the 8 neighbor pixels to obtain the DLBP value. Then, we compute the number of different DLBPs to consider the intensity variation around the transition pixel by algorithm 1. Thus the total number of potentially different DLBPs is K. Algorithm 2 explains the extraction of DLBP feature vector.

Let w_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:

$$POT_i = w_i \times NOD_i, \qquad i=1\ldots\ldots\ldots N \qquad (6)$$

Where N denotes the number of candidate regions as mentioned. NOD_i denotes the number of different DLBPs, which is normalized by the maximum of the number of different DLBPs (i.e.,$K_{15\%}$) in each candidate region. If POT of the candidate region is larger than a predefined value, the corresponding region is finally determined as the text region. The detection result is shown in Fig. 6. The thresholding value in POT is empirically set to 0.05 based on various experimental results.We can see that the text region is well identified from other candidates.

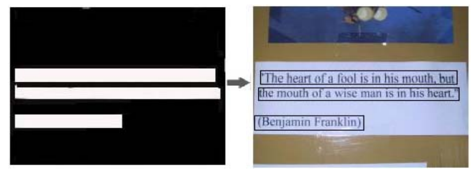

Fig.6.Text Region Determination

D. Section Headings

The text region or the bounding box obtained in the preceding subsection needs to be refined for better accurate text extraction. In this subsection, we use a modified projection of transition pixels [17] in the morphological binary map to perform the text region refinement. First, the horizontal projection is performed to accumulate all the transition pixel counts in each row of the detected 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, our projection method is applied to the detected text regions only, making the process simpler. The result of refinement is shown in Fig. 7.

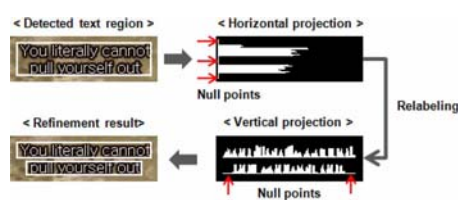

Fig.7.Refinement process of detected text region

E. Text Region Update

Once the text regions are detected in the current frame, it is reasonable to take advantage of continuity of text between consecutive frames for the text region detection of the next frame. If the difference, which can be obtained by XOR of current morphological binary map and previous morphological binary map, is smaller than a predefined value, the 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 morphological binary map with the morphological binary map obtained 3 frames earlier and the dissimilarity measure between these maps is defined as follows:

$$d\,(M_n,M_{n-3}) = \sum_{(x,y)\in T} (M_n(x,y) \otimes M_{n-3}(x,y))$$

$$\text{if } (d(M_n,M_{n-3})<T)\ TR_n=TR_{n-3}$$
$$\text{Otherwise, find new } TR_n \qquad (7)$$

Where M_n and M_{n-3} denote the morphological binary map obtained from *n*th frame and the (*n*-3)th frame, respectively. TR_n and TR_{n-3} denote the detected text regions in the *n*th frame and (*n*-3)th frame, respectively. $\otimes$ denotes the XOR operator. If the values on the *n*th frame and the (*n*-3)th frame morphological binary 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 text region update method can reduce the processing time efficiently.

IV. TEXT EXTRACTION

Before applying video OCR application, the refined text regions need to be converted to a binary image, where all pixels belonging to 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 text dealing with complex backgrounds and various text appearances. In this section, we propose a fast and efficient text extraction technique, which is based on Lyu's approach [7].

A. *Color Polarity Computation*

Color based text extraction technique [17] is proposed for text extraction. The 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. We observe that this goal can be simply attained owing to the morphological binary map obtained in the preceding section. First of all, the binary image obtained by thresholding with average intensity value can be effectively utilized. 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 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" as expressed in (8).

$$\text{Bright\_text\_flag}= \begin{cases} 1, \text{ if } I_B(x_F,y_F) = 1 \\ \qquad \text{and } I_B(x_F+2,y_F) = 0 \\ 0, \text{ Otherwise} \end{cases} \qquad (8)$$

Where (x_F,y_F) denotes the position of the first encountered transition pixel in each row of the text region and I_B denotes the value on the binary image.

The flag is set to 1 if 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 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.8.

Fig.8. Process of inversing image by the color polarity. (a) Dark text on bright background. (b) Bright text on bright background. (c) Examples of inverted text by "bright_text_flag".

As shown in Fig. 8, the flag is set to 1 for Fig. 8(a) since the first encountered transition pixel belongs to 1, whereas the pixel apart by two pixel distance belongs to 0. The first transition pixels in each row on the binary image are represented by red color in Fig. 8. Examples with "bright_flag_text" are also shown in Fig. 8(c).

V. EXPERIMENTAL RESULTS

The proposed approach has been tested on real-life videos. The experiments have been performed on a Pentium PC with 333 MHz CPU. The program is implemented in MATLAB v7.0.As there is no standard database available we created our own dataset consisting of 15 MPEG-1 video sequences with 320 × 240 pixel resolution. There were a total of 5,299 frames (about 75MB of data). There were 156 overlay text events and 144 scene text events in the video data. All text had horizontal orientation and most text events were stationary. The dataset contained a wide variety of video captured from television channels, including television commercials and news broadcasts (domestic and foreign). A wide variety of text fonts, colors, languages, and scripts were represented. Video sequences were captured at 30 frames per second. The proposed method is processed in frames.

(a)

(b)

Fig.9.Experimental results of overlay text and scene text detection and extraction (a) original frames (b) Result

Scene texts are difficult to detect than overlay texts because some of them are irregularly aligned and with different size. Fig.9 gives some result of the proposed algorithm for overlay text and scene text.

A. *Performance Evaluation*

In order to confirm the superiority of our proposed text detection and extraction method, we compare our approach with other methods; Lyu's method [7] and Kim's method [17].The accuracy of text extraction shown in Fig. 8 is evaluated using the probability of error (PE) [18] as follows:

$$PE = P(T)P(B\backslash T) + P(B)P(T\backslash B) \qquad (9)$$

Where P(T) and P(B) denote the probabilities of text pixels and background pixels in the ground-truth image ,respectively. P(B\T) denotes the error probability to classify text pixels as background pixels. P(T\B) denotes the error probability to classify background pixels as text pixels. The comparison of PE for each extraction result and Total Processing Time is shown in Table 1.

TABLE1

COMPARISON OF EFFICIENCY OF EXTRACTION

Images	Kim's Method	Proposed Method
1	0.086	0.072
2	0.076	0.065
3	0.059	0.059
4	0.057	0.055
5	0.074	0.067
6	0.070	0.068
7	0.102	0.087
8	0.132	0.098
9	0.078	0.067
10	0.092	0.088

The comparison of proposed method with Kim's method is shown in Table 1 and Fig 10. It clearly shows that Average PE is reduced.

Fig.10. PE for Each Extraction Result

DLBP is used to describe the texture around the transition pixel. Unlike LBP, which makes use of all possible patterns, DLBP consider only the most frequently occurred patterns in a texture image. The dominant patterns around 80% of the total pattern occurrences, which can effectively captures the image textural information is used to describe the texture. Thus by using DLBP and updation between frames are employed the time complexity is reduced.

VI. CONCLUSION

Text embedded in videos often carries the most important information, such as time, place, name or topics, etc. This information may do great help to video indexing and video content understanding. A novel method for text detection and extraction from complex videos is proposed in this paper. Our detection method is based on the observation that there exist contrast colors between text and its adjacent background. The morphological binary map is first generated by obtaining the difference between closing and opening image. Connected components for each candidate region are generated and then each connected component is reshaped to have smooth boundaries. The dominant local binary pattern is used to find the intensity variation around the transition pixel. The boundaries of the detected text regions are localized accurately using the projection of text pixels in the morphological binary map. Text region update between frames is also exploited to reduce the processing time. Based on the results of text detection, the texts are extracted based on color polarity computation method. To validate the performance of our detection and extraction method, various videos have been tested. The proposed method is very useful for the real-time application. Our future work is to detect and extract the text with different orientations to extend the algorithm for more advanced and intelligent applications.

REFERENCES

[1] A. Del Bimbo, C. Colombo and P. Pala. "Semantics in visual information retrieval". IEEE *Multimedia*, 6(3), 1999.

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

[3] 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*, vol. 1, pp. 425–428 ,Aug. 2004.

[4] J. Cho, S. Jeong, and B. Choi, "News video retrieval using automatic indexing of korean closed-caption," Lecture Notes in Computer Science, vol. 2945, pp. 694–703, Aug. 2004.

[5] L. Agnihotri and N. Dimitrova, "Text detection for video analysis," in *Proc. IEEE Int.Workshop on Content-Based Access of Image and Video Libraries*, Jun. 1999, pp. 109–113.

[6] K. C. K. Kim, H. R. Byun, Y. J. Song, Y. W. Choi, S. Y. Chi, K. K. Kim, Y. K. Chung "Scene text extraction in natural scene images using hierarchical feature combining and verification," in *Proc. Int. Conf. Pattern Recognition*, Aug. 2004, vol.2, pp. 679–682.

[7] 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.

[8] Xi Jie, Xian-Sheng Hua, Xiang-Rong Chen, Liu Wenyin, HongJiang Zhang, "A Video Text Detection And Recognition System", 2001.

[9] M. Bertini, C. Colombo, and A. D. Bimbo, "Automatic caption localization in videos using salient points," in *Proc. Int. Conf. Multimedia and Expo*, Aug. 2001, pp. 68–71.

[10] Y. Zhong, H.J. Zhang, A.K. Jain, Automatic caption localization in compressed video, IEEE Transactions on PAMI 22 (2000) 385–392.

[11] N. Otsu, "A threshold selection method from gray-levl histograms,"*IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, Mar. 1979.

[12] T. Sato, T. Kanade, E. K. Hughes, and M. A. Smith, "Video OCR for digital news archive," in *Proc. IEEE International Workshop on Content Based Access of Image and Video Libraries*, Jan. 1998, pp. 52–60.

[13] Jui-Chen Wu, Jen-Wei Hsieh and Yung-Sheng Chen", Morphology based text line extraction", pp.1195-1200, 2006.

[14] Jagath Samarabandu *and* Xiaoqing Liu," An Edge-based Text Region Extraction Algorithm for Indoor Mobile Robot Navigation", pp.273-280. 2007.

[15] S. Liao, Max W. K. Law, and Albert C. S. Chung," "Dominant Local Binary Patterns for Texture Classification", IEEE *Transactions on Image Processing,* Vol. 18, No. 5, May 2009.

[16] 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.

[17] Wonjun Kim and Changick Kim, "A New Approach for Overlay Text Detection and Extraction from Complex Video Scene", IEEE *Transactions on Image Processing,* Vol. 18, No. 2, February 2009.

[18] S. U. Lee, S. Y. Chung, and R. H. Park, "A comparative performance study of several global thresholding techniques for segmentation," *Comput. Vis., Graph., Image Process.*, vol. 52, pp. 171–190, 1990.