A Fast Caption Detection Method for Low Quality Video Images

Tianyi Gui, Jun Sun, Satoshi Naoi
Fujitsu Research & Development Center CO., LTD.,
Beijing , P. R. China
{guitianyi, sunjun, naoi}@cn.fujitsu.com

Yutaka Katsuyama, Akihiro Minagawa,
Yoshinobu Hotta
Fujitsu Laboratories Ltd,
Kawasaki, Japan
{katsuyama, minagawa.a, y.hotta}@jp.fujitsu.com

***Abstract*—Captions in videos are important and accurate clues for video retrieval. In this paper, we propose a fast and robust video caption detection and localization algorithm to handle low quality video images. First, the stroke response maps from complex background are extracted by a stoke filter. Then, two localization algorithms are used to locate thin stroke and thick stroke caption regions respectively. Finally, a HOG based SVM classifier is carried out on the detected results to further remove noises. Experimental results show the superior performance of our proposed method compared with existing work in terms of accuracy and speed.**

Keywords: Stroke filter, Text localization, Text information extraction

I. INTRODUCTION

With the increasing of digital media resources and the development of Internet, content-based video analysis and retrieval becomes a more and more important research topic. The captions, which include plentiful semantic information, provide useful clues for the video analysis and retrieval. Therefore, the caption recognition becomes a necessary part of those video analysis projects. The precise caption detection for locating the text lines is the first and important step in the caption recognition system.

Our target is to detect the captions in the Internet video. The low quality nature of the Internet videos images introduces some unique problems: Firstly, the video images are typically in low resolution and compressed. Color bleeding between texts and background is very common. Secondly, the captions in the video images usually have low contrast with multi-font and multi-color. Furthermore, as a pre-processing step of the caption recognition system, there are two requirements for our detection algorithm: 1) The speed should be fast. 2) High recall rate are required in the detection stage.

Until now, many effective methods for text detection and localization have been proposed in the last two decades. These methods can be mainly classified into three categories: texture analysis based, edge/corner based and stroke based. However, few of them can address the problems we mentioned above simultaneously.

Texture based methods [1][2][3] are good candidates for low quality video images' captions detection. By using text texture and classifier for sampled windows, it performs well for low quality video images with high recall and precision. However, this kind of methods has two drawbacks: 1) its speed is slow due to the high computational complexity of feature extraction and window classification on multi-scale. 2) Its performance depends on the training data, which will limit the capability for multilingual caption detection.

Inspired by the observation of rich edge and corner information within text areas, edge/corner based caption detection methods [4][5][6][7][8][12] are very popular. However, its precision will be influenced by the non-text region with complex texture. Moreover, to overcome the influence of scale, multi-scale analysis is necessary for these methods. Thus, the computation complexity is high.

The stoke-based method is very reasonable. How to extract the strokes precisely is crucial. The stroke extracting methods, such as those based on color clustering, local binarization, and stroke models, have been used to achieve good performance in past papers [9][10][11]. However, color is not a stable feature for low quality video images and some captions include more than one color. For local binarization, deciding the scale of video texts is difficult, and multi-scale analysis [10] to capture text strokes with different sizes is time consuming. Stroke model based method works well for low quality image, but there is no satisfactory solution of how to estimate the stroke width precisely.

In this paper, a fast stroke-based caption detection method is proposed without multi-scale analysis. Thin and thick strokes are extracted according to the pair-wise characteristic of stroke edge. On extracted strokes images, two different algorithms are designed to detect and locate different size caption regions precisely. A texture-based method is finally used as a verification stage to remove the false detected regions. Section II gives the detail description of our algorithm. Section III shows the experimental results, including both the performance of detection accuracy and the speed.

II. FAST CAPTION DETECTION METHOD

In order to detect the multi-size and multi-color captions from the low quality video images with fast speed, we propose a novel framework as shown in Fig. 1.

In the beginning, we extract the image edge-maps by Sobel operator on four orientations and two polarities. At the first step, to avoid the influence of noise edges and prevent the multi-scale Gauss-pyramid edge analysis, we design an effective method which could extract the strokes with different size in one pass. After that, stroke density analysis and adjacent character grouping are proposed for the accurate text localization on the extracted strokes.

978-0-7695-4661-2 2012

DOI 10.1109/DAS.2012.2

Finally, the detected text blocks will be verified by a texture analysis algorithm. Furthermore, our method can give an index whether the detect text region is normal or inverse text region (Normal text region is defined as a region where the characters have darker grayscale value than that of the background. Inverse text region is opposite to the normal text region [5]).

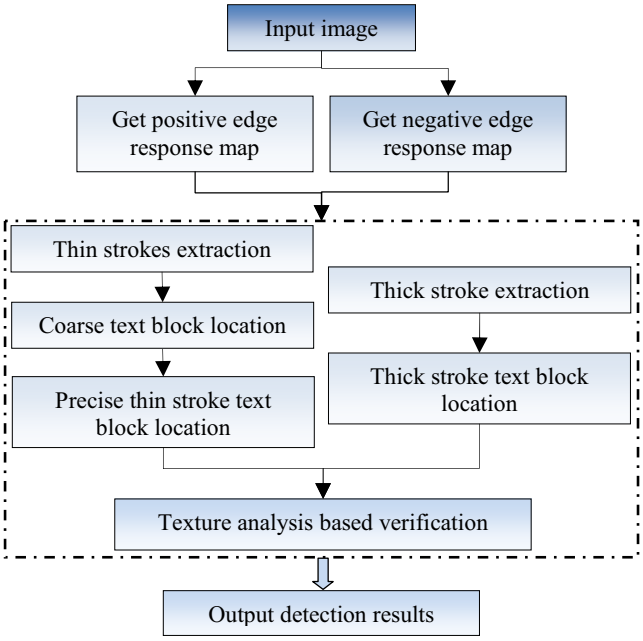


Figure 1: Framework of caption detection system.

A. Preprocessing

At the preprocessing stage, we obtain the positive and negative edge response maps on four orientations and two polarities by Sobel edge detector as shown in Fig. 2.

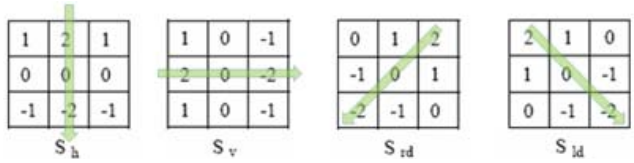


Figure 2: Sobel masks on 4 directions.

The original image and its 8 edge response images are shown in Fig. 3, called $I_{h\text{-}pos}$, $I_{h\text{-}neg}$, $I_{v\text{-}pos}$, $I_{v\text{-}neg}$, $I_{rd\text{-}pos}$, $I_{rd\text{-}neg}$, $I_{ld\text{-}pos}$, $I_{ld\text{-}neg}$ respectively. The subscripts represent their characteristics: 'h' means horizontal directional edge. 'v' means vertical directional edge. 'positive' and 'negative' stand for the positive and negative respone value after convoluted with the sobel mask.

Compared with the commonly used Canny edge detector, our edge response maps not only give a quantittative measurement of the edge, but also provide the orientation and polarity information. The following steps will use this information to judge whether the extracted edges belong to the stroke or background.

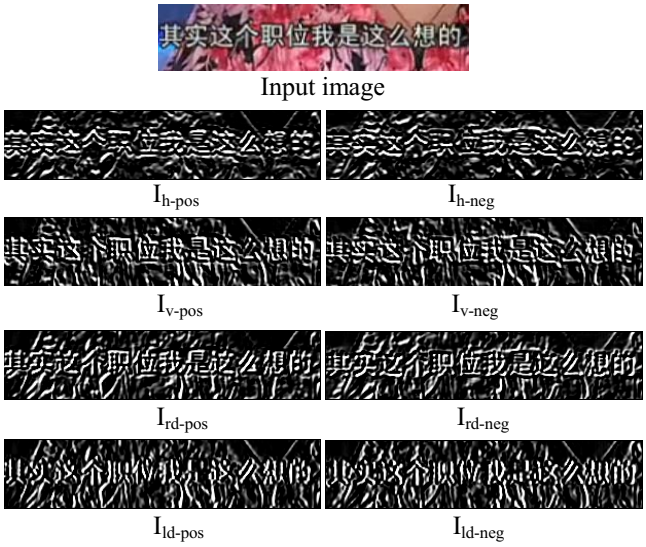


Figure 3: Extract stokes' edges on four orientations.

B. Stroke Extraction Based on Edge Response Map

Generally, we could treat the stroke as a pulse signal with one parameter: stroke width. After convoluted with Sobel operator, its response could be classified into four cases, as shown in Fig.4.

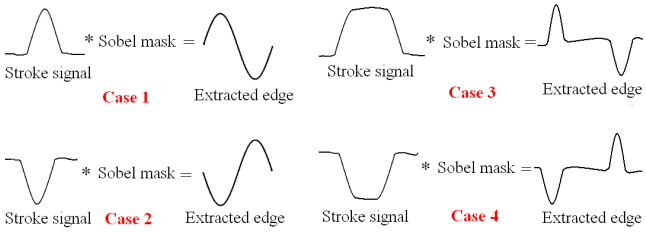


Figure 4: Four cases for edge response of a stroke.

For the thin stroke signals in the case 1 and 2, we can get "continuous" positive/negative responses due to the thin stroke width. For the thick strokes signals in the case 3 case 4, the positive/negative edge responses will have "a little distance" after the convolution.

1) Thin Stroke Extraction

According to the analysis above, we can use the special characteristic of thin stroke edge response to enhance the stroke. The main process is shown in Fig. 5.

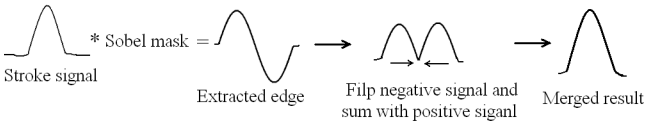


Figure 5: Merge the positive and negative response for thin stroke.

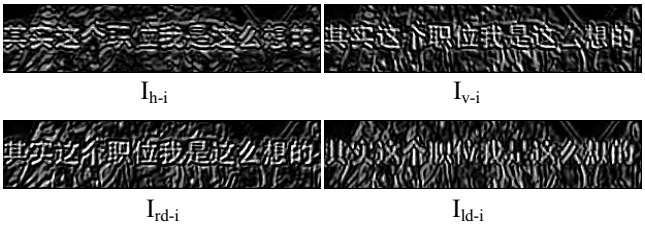
After flipping the negative response and summing with the positive response, the total response of the thin stroke will

be strengthened. Moreover, this operation will suppress the edge responses which do not belong to thin strokes.

Assuming we are dealing with the inverse text, the merging formulas are as follows:

$$I_{h\text{-}i}(x,y) = (I_{h\text{-}pos}(x,y\text{-}w) + I_{h\text{-}neg}(x,y+w))/2;$$
$$I_{v\text{-}i}(x,y) = (I_{v\text{-}pos}(x\text{-}w,y) + I_{v\text{-}neg}(x+w,y))/2;$$
$$I_{rd\text{-}i}(x,y) = (I_{rd\text{-}pos}(x+w,y\text{-}w) + I_{rd\text{-}neg}(x\text{-}w,y+w))/2;$$
$$I_{ld\text{-}i}(x,y) = (I_{ld\text{-}pos}(x\text{-}w,y\text{-}w) + I_{ld\text{-}neg}(x+w,y+w))/2. \quad (1)$$

The subscript 'i' means the text image is the inverse text image. (x, y) is the coordinate of processed point, w is a parameter in our formula and means the edges' response offset step. Processed results are shown in Fig. 6.



$I_{h\text{-}i}$ $I_{v\text{-}i}$

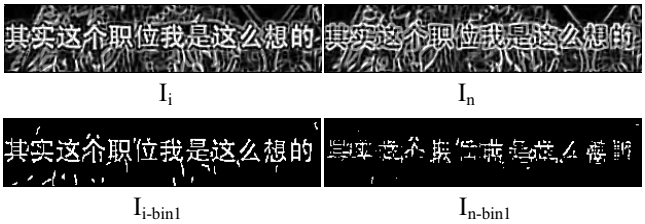
$I_{rd\text{-}i}$ $I_{ld\text{-}i}$

Figure 6: Extract stokes on four orientations.

After merging the positive and negative responses on four orientations, we get four orientation stroke images. And we use Equation (2) to merge them:

$$I_i(i,j) = (I_{h\text{-}i}(i,j) + I_{v\text{-}i}(i,j) + I_{rd\text{-}i}(i,j) + I_{ld\text{-}i}(i,j))/4. \quad (2)$$

The operation on the normal text image is similar to that of the inverse image in Equation (1) and (2). For the two merged edge response maps in normal and inverse modes, we could get a satisfactory binarized stroke image by simple global based binarization with a higher threshold (for example, take Th_{Otsu}+20 as the threshold). The result is shown in Fig. 7.



I_i I_n

$I_{i\text{-}bin1}$ $I_{n\text{-}bin1}$

Figure 7: merged edge response and the binarization result of inverse image (left) and normal image (right).

2) Thick Stroke Extraction

For the thick strokes, unlike the conventional multi-scale analysis based algorithms, a fast and effective stroke extraction method is proposed based on the same merged edge response images.

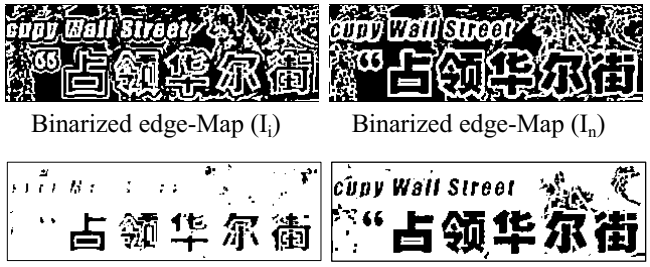
As shown in the edge merged images $I_{n\text{-}bin1}$/$I_{i\text{-}bin1}$, they also could be treated as the processed edge-map images of thick stroke. And these edges will form closed contours around strokes generally, as shown in Fig. 8. Although there are edge responses on background area, they are disordered and could not form close contours.

Input image

Edge-Map (I_i) Edge-Map (I_n)

Figure 8: Edge map of thick stroke.

Based on the analysis above, we could binarize the edge-map images I_i and I_n using a lower threshold (for example, take Th_{Otsu}-20 as the threshold) and treat the CCs in the close contours as the potential strokes, as it shown in Fig. 9:



Binarized edge-Map (I_i) Binarized edge-Map (I_n)

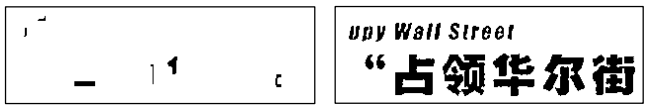
Potential strokes of I_i Potential strokes of I_n

Figure 9: Potential stroke-CCs.

We make some assumptions on the thick strokes to remove the noise strokes in the potential stroke-CCs images:

1) The gradient value on stroke–CCs' external contour should be larger than a threshold.
2) The stroke width should be larger than a threshold.
3) The area and size of the stroke-CCs should be larger than a threshold.

The final extracted strokes images are shown in Fig. 10, and their corresponding binarized images are named as $I_{n\text{-}bin2}(i,j)$ / $I_{i\text{-}bin2}(i,j)$ respectively.



$I_{i\text{-}bin2}$ $I_{n\text{-}bin2}$

Figure 10: Final stroke-CCs of thick stroke image in the normal and inverse image.

C. Caption Localization Based on Extracted Text Strokes

Based on the characteristics of the extracted stroke image, we design two different text localization methods for the thin and thick stroke captions separately.

1) Thin Captions Localization

Since the stoke width has been estimated for thin stroke, we could use stroke density analysis to get the captions'

coarse region and locate them accurately based on projection analysis. Assume the black pixels are background and the white pixels are foreground, the main steps are shown in Table 1 and Fig. 11.

Table 1
Thin stroke localization

Get the captions' coarse region

1) Get the binarized image $I_{bin1}(i, j) = I_{n\text{-}bin1}(i, j) \cup I_{i\text{-}bin1}(i,j)$;
2) Use a sliding window to scan all pixels of image I_{bin1},
   a) Calclulate the white pixles number in the scanning window.
   b) If the number is larger than a threshold, the center point of the window will treated as the text pixl and be labeled as white(255), otherwise it will be labeled as black(0).
3) The white pixl CCs will be treat the candidated text region.

Get the captions' accurate region

1) For every candidate text region, judge whether they belongs to normal or inverse text.
   a) Calcluate the white pixels number of $I_{n\text{-}bin1}$ and $I_{i\text{-}bin1}$ on the candidate text region.
   b) If the white pixels number of $I_{n\text{-}bin1}$ is larger than it on $I_{i\text{-}bin1}$, we treat the candidate text region as normal text, else it is inverse text.
2) Projection analysis
   a) Partition the stroke pixles into row or column according to their horizontal or vertical projection profile.

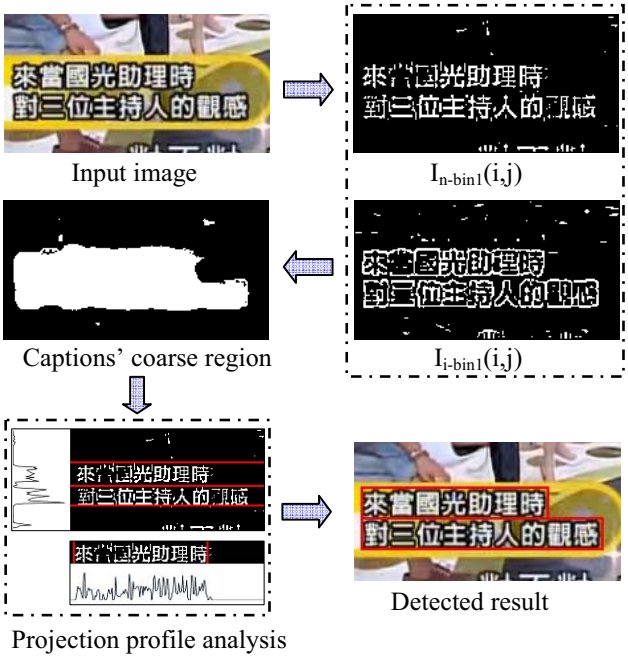


Figure 11: Flowchart of thin captions localization.

2) Thick Stroke Captions Localization

The thick stroke extraction algorithm creates a set of CCs from the input image, including both real strokes and noises, which could not be removed by simple heuristic rules. Assuming that a text string has at least two similar size strokes in alignment, we propose a stroke grouping method to group the strokes into text line and remove the noise. The main steps are shown in Table 2 and Fig. 12.

Table 2
Thick stroke caption localization

Assume we are dealing with the horizontal caption lines, we will treat two stokes belonging to a same text line if their overlap rate between them is larger than 0.9 on vertical directions,

Do{
1) Scan all unprocessed CCs, label their overlap relationships on vertical directions.
2) Find a CC which has the largest number of overlap CCs.
3) Label this CC and its overlap CC or the CC contained by this CC on vertical directions as a text line, and label them as processed CC.

}while(All the CC have benn proceseed or
the remain CC do not own overlap CC)

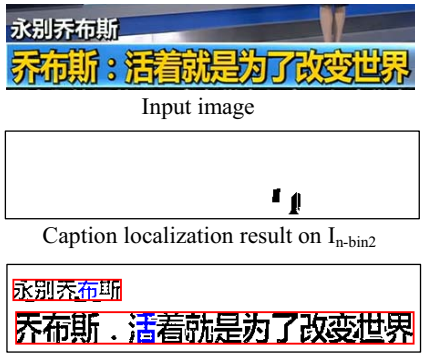


Figure 12: Flowchart of thick stroke captions localization.

D. Verification for the Located Text Blocks

Although we have designed an effective algorithm for caption detection and localization, there are still some noisy blocks remained. A texture analysis and classification based method is used to further verify the true text lines and remove the noises.

1) Text Feature Selection and Classifier design

For the texture feature extraction, we select the HOG feature since its superior performance than other widely-used features (LBP, DCT and Gabor) [3]. Another benefit of using HOG feature is that the feature can be extracted directly from the edge response map obtained in the preprocessing stage.

Linear SVM is selected as the classifier is used due to its good generalization capability and low computational complexity.

2) Text line Verification

Assuming we are verifying a horizontal text block, a moving window is used to scan the input block image on different positions. The window size is the height of input block image and the step size is one third of the height. After extracting their features from sampled windows, we use the linear-kernel SVM classifier to decide whether these sampled windows contain text information or not, as the Fig. 13 shows. If the 20% of the sub-windows of the text block is

judged as containing the text information, we will treat this text block as a correct detected text block to achieve a high recall.

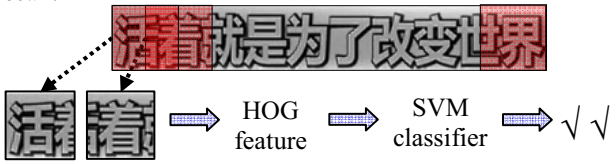


Figure13: The verification process.

III. EXPERIMENTS

A. Dataset and Settings

Our system was coded in C++ and run on a PC with a CPU P8700 on 2.53GHz.

7 videos were collected, including news, movie, advertisement and entertainment. For every fifteen frames we sample one frame image and perform caption detection. The total test images number was 6213.

The SVM classifier was trained with linear kernel using 10000 text and 30000 non-text patterns selected from other video images. Each pattern was normalized to 16*16 images and the feature we used is 8-direction HOG feature.

B. Evaluation Rules

The performance measured on the text region level. If the intersection of the detected text region (DTR) and the ground-truth text region (GTR) covers more than both 90% of the DTR and GTR [4], the detected text line is regarded as a true text line.

The recall and precision rate are thus defined as:

Recall = (number of correct DTRs) / (number of GTRs)

Precision = (number of correct DTRs) / (number of DTRs)

C. Experimental Results

The average processing time for one image with resolution 720*480 is only 92ms. The detailed time distribution is listed in Table 3.

Pre-processing	*Stroke Extraction*	*Text Localization*	*Verification Step*	*All*
55ms	23ms	8ms	6ms	92ms

Table 3: Time cost for each stage.

The performance of our caption detection engine with/without verification step is shown in table 4. From this table, we could find that the verification step could remove about 30% of false detected text regions while preserving the same recall rate.

Methods	*Recall*	*Precision*
Our methods without versitifcation step	96.1%	91.5%
Our methods with versitifcation step	96.1%	94.1%

Table 4: Engine's performance with/without verification.

To evaluate the proposed method, the method in [7] was selected to compare due to its fast and effective performance. As shown in Table 5. It can be observed that both recall and precision rates of our method is much better than the corner based one. The experimental results show that our caption detection method could achieve considerably better performance while keeping a very fast speed.

Video Type	*Number of GTRs*	*Our Method*		*[7]'s method*	
		Recall	*Precision*	*Recall*	*Precision*
News	2424	99.7%	99.5%	92.9%	85.5%
Movie	230	98.1%	94.3%	92.6%	51.5%
Advertisement	181	100%	80.9%	42.0%	54.6%
Entertainment	2422	83.7%	90.1%	52.3%	80.1%
ALL	5052	96.1%	94.1%	72.6%	79.0%

Table 5: Comparison with corner based method [7].

IV. CONCLUSIONS

In this paper, a novel and fast stroke filter is proposed for stroke extraction. Two text line extraction algorithms are used to detect thin and thick stroke text lines. A texture based noise removal function is used as verification. Furthermore, our method could not only give the accurate location of text lines with fast speed, but also give an index whether the detect text region is normal or inverse text.

REFERENCES

[1] D. T. Chen, J. M. Odobez, and H. Bourlard. Text detection and recognition in images and videos frames. Pattern Recognition, vol. 37, no. 3, pp. 595–608, 2004.

[2] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes, IEEE International Conference on Computer Vision and Pattern Recognition, pp. 366–373, 2004.

[3] Y. Pan, C. Liu, X. Hou. Fast scene text localization by learning-based filtering and verification, IEEE International Conference on Image Processing, pp. 2269-2272, 2010.

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

[5] M. Anthimopoulos, B. Gatos, I. Pratikakis. A two-stage scheme for text detection in video images. Image and Vision Computing, vol. 28, no. 9, pp. 1413-1426, 2010.

[6] Hua, X., Chen, X.R., Liu, W., Zhang, H.J., Automatic location of text in video frames, In: Proc. ACM Multimedia Workshop: Multimedia Information Retrieval, pp. 24-27, 2001.

[7] H. Bai, J. Sun, S. Naoi, Y. Katsuyama, Y. Hotta, K. Fujimoto. Video caption duration extraction, International Conference on Pattern Recognition, pp. 1-4, 2008.

[8] X. Zhao, K. Lin, Y. Fu, Y. Hu, Y. Liu and T. S. Huang. Text From Corners: A Novel Approach to Detect Text and Caption in Videos, IEEE Transactions on Image Processing, vol. 20, no. 3, pp. 790-799, 2011.

[9] C. Yi and Y. Tian. Text String Detection from Natural Scenes by Structure-based Partition and Grouping, IEEE Transactions on Image Processing, vol. 19, no. 12, 2011.

[10] Y. Pan, X. Hou, and C. Liu. A hybrid approach to detect and localize texts in natural scene images, IEEE Transactions on Image Processing, vol. 20, no. 3, pp. 800–813, 2011.

[11] X. Ye, M. Cheriet, Ching Y. Suen. Stroke-model-based character extraction from gray-level document images, IEEE Transactions on Image Processing, Vol. 10, no. 8, pp. 1152-1161, 2001.

[12] S. P, T Q. Phan and C L. Tan. A Laplacian approach to multi-oriented text detection in video, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, no. 2, pp. 412-419, 2011.