

Efficient Video Text Detection using Edge Features

Palaiahnakote Shivakumara, Weihua Huang and Chew Lim Tan
School of Computing, National University of Singapore
{shiva, huangwh, tancl}@comp.nus.edu.sg

Abstract

In this paper, we explore new edge features such as straightness for the elimination of non significant edges from the segmented text portion of a video frame to detect accurate boundary of the text lines in video images. To segment the complete text portions, the method introduces candidate text block selection from a given image. Heuristic rules are formed based on combination of filters and edge analysis for identifying a candidate text block in the image. Furthermore, the same rules are extended to grow boundary of candidate text block in order to segment complete text portions in the image. The experimental results of the proposed method show that the method outperforms existing method in terms of a number of metrics.

1. Introduction

With hundreds of thousands of hours of archival videos, there is an urgent demand for tools that allow efficient browsing and retrieving of video data [1]. To achieve fast retrieving, one should have a simple and efficient text detection method [2]. With this intention, many methods have been proposed in literature. Our literature study shows that connected-component based methods are not robust because they assume that text pixels belonging to the same connected region share some common features such as color or grey intensity [3, 4]. On the other hand, texture based methods may be unsuitable for small fonts and poor contrast text [5, 6], while edge based methods give more false alarms and are not robust for complex background images [7- 8]. Hence, there is a great demand for developing a method which gives better detection rate with fewer false alarms irrespective of size, type, font and background of the texts in the video image [9, 10].

2. Proposed Methodology

We assume that text present in images is in horizontal direction with uniform spacing between the words. The proposed method finds candidate text blocks based on heuristic rules and then the same rules are used for segmenting complete text portions in the video image. The method then eliminates edges that are satisfying straightness property. If the centroid of an edge falls on itself, then the edge is said to possess the straightness property. Projection profiles of the edge image and information about text alignment are used for detecting text blocks with fewer false alarms. The method is fast as it uses the edge map of the segmented text portion of a video frame to detect text blocks. This is a great advantage of the proposed method comparing to existing methods.

2.1. Candidate Text Block Selection

We present rules for identifying candidate text block after dividing the whole 256×256 pixel image into 16 equal sized blocks

that are 64×64 pixels, as shown in Figure 1 and Figure 2. We have chosen block size as 64×64 pixels because the block is expected to have some text portions (two words), taking Figure 3 (a) as an example. The method uses Arithmetic mean Filter (AF) and Median Filter (MF) to derive rules. These filters are well known filters to remove noise in an image [11]. The motivation and justification to choose these filters for deriving rules is clearly given in [12]. The AF is defined as

$$\hat{f}_{AF}(x,y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s,t) \quad (1), \text{ where } S_{xy} \text{ represent the set of}$$

coordinates in a rectangular sub-image window of size $m \times n$, centered at point (x, y) and $g(x, y)$ is the pixel intensity. Noise is reduced as a result of blurring. This fact motivates us to derive filter based rules for identifying text blocks.

The MF is defined as

$$\hat{f}_{MF}(x,y) = \text{median}\{g(s,t)\}_{(s,t) \in S_{xy}} \quad (2). \text{ As we know that the AF}$$

attenuates noise but it blurs the image as shown in Figure 3 (b) and the MF attenuates noise pixels but it doesn't blur the block as shown in Figure 3 (c) [11]. The method finds the degree of blurring produced by AF, by subtracting the output of AF from the output of MF (D_{MA}) for both text blocks and non text blocks (refer to Figure 3 (d)). The method computes the number of Sobel edge components for AF block (NS_{AF}) as shown in Figure 3 (e) and the number of Canny edge components (NC) for the differenced block ($NC_{D_{MA}}$) as shown in Figure 3 (f). If the number of Sobel edge components in AF block is greater than the number of Canny edge component in D_{MA} then it is a text block otherwise it is a non-text block. We also believe that Sobel edge detector detects more edges when text is present and it detects fewer edges when non text present especially in case of video images with poor contrast.

Some times, the first rule fails to identify the correct text block as we can expect more Sobel edges in non-text portions also. Therefore, we derive one more rule based on the strong and weak edges of Canny and Sobel. If the number of strong edges in MF block (NST_{MF}) is greater than the number of strong edges in D_{MA} block ($NST_{D_{MA}}$) then it is a text block otherwise it is a non-text block. The strong edges are obtained by subtracting the number of weak edges from the number of Canny edges. The number of weak edges is obtained by subtracting the Canny edge image from the Sobel edge image of the MF (refer Figure 3(g-i)) and D_{MA} (refer Figure 3(j-l))

This procedure gives the number of text and non-text blocks out of 16 blocks in the image. The method finds the difference between NS_{AF} and $NC_{D_{MA}}$ for 16 blocks. From the difference value list, the highest difference (HD) value is chosen and if a block corresponding to HD in rule 2 gives positive difference value then it is considered as a candidate text block. This is illustrated in Figure 4(b) where the 6th block of rule 1 gives HD value but for the same block, rule 2 gives negative value. In this case, the 14th block is chosen as a candidate text block as it satisfies both criteria. If a block does not satisfy this criterion then the method searches the next highest difference in rule 1. It is noticed from Figure 4(a) that the 15th block satisfies both criteria. It is also observed from Figure 14(a) and (b) that blocks 14, 15 and 16 (end of the image in Figure 1) give positive

difference values. This infers that the method gives positive values when there is a text in the blocks.



Figure 1. Grey image Figure 2. 16 Blocks of the image

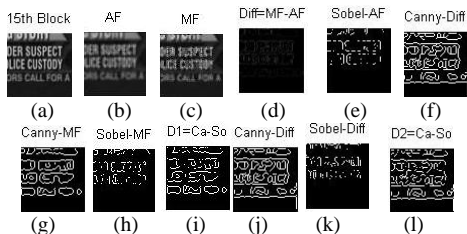


Figure 3. Steps involved for candidate text block selection

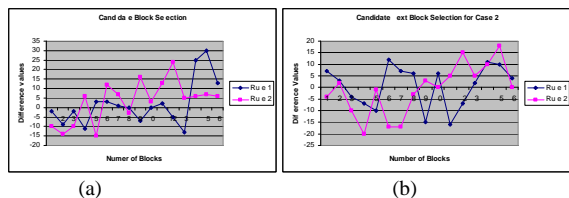


Figure 4. Illustration for candidate block selection

2.2. Segmentation of Text Portion

We present a block growing method using the above rules described in section 2.1 to obtain complete text portion of the image. The method uses rule 2 $NST_{MF} > NST_{D_MA}$ to grow candidate text block boundary. The Rule 1 and rule 2 are used for stopping the boundary growing. First, the boundary grows towards right direction from the candidate text block as shown in Figure 5 (a-e) by incrementing step 16 pixels. Then boundary grows towards left as shown in Figure 6(a-c). Boundary grows upward and downward in a similar way, and the final boundary is shown in Figure 7. The step increment of 16 pixels is estimated based on the space between the characters in the words.

If $NST_{MF} < NST_{D_MA}$ then growing stops, else if the first difference values in rule 1 is positive then the difference value between the first iteration and second iteration is checked. If there is any huge difference with more than 12 then if the difference is decreasing for the next iteration, boundary growing stops. It is illustrated in Figure 8(c), where there is a sudden jump from 3rd to 4th iteration in rule 1. If the first difference value in rule 1 is negative then the difference value in rule 2 is checked and if huge difference is found from iteration to iteration with more than 12 then boundary growing stops. It is illustrated in Figure 8(d), where the first value in rule 1 is negative but there is no jump from iteration to iteration in rule 2. In this case, the boundary grows till end of the image. Illustration given in Figure 8(a) for the right growing, where the boundary stops at the 4th iteration as there is a transition from positive to negative in rule 1. It is revealed from experimental results that the sudden jump occurs when the algorithm crosses from text to non-text area. If all three conditions fail then we consider the whole

image for detecting text. The procedure works for most of the images collected in our dataset.

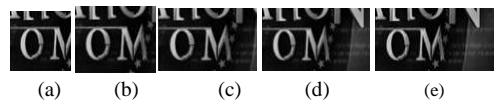


Figure 5. Block growing towards right direction



Figure 6. Block growing towards left direction



Figure 7. Final boundary of text portion

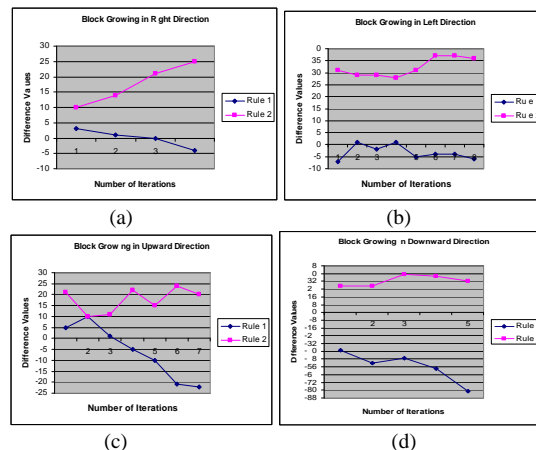


Figure 8. Illustrations for Block growing

2.3. Edge Elimination

The edges are detected using the Canny edge detector and edges containing less than four pixels are eliminated since they are too small to be labeled as straight. Let $X=\{x_1, x_2 \dots x_n\}$ and $Y=\{y_1, y_2 \dots y_n\}$ be the sets of x and y co-ordinates respectively, of the edge pixels. The centroid of an edge is (C_x, C_y) defined as $C_x = \frac{1}{n} \sum_{i=1}^n x_i$

and $C_y = \frac{1}{n} \sum_{i=1}^n y_i$, where n is the number of pixels in the edge.

Using this definition of centroid, the straightness of an edge is defined as

$$Cent - Edge = \begin{cases} 1, & (C_x \in X) \cap (C_y \in Y) \\ 0, & otherwise \end{cases} \quad (3)$$

The edges that satisfy equation (3) are eliminated from the edge image in order to facilitate text detection using projection profiles. Examples of such edges are given in Figure 9. As the centroid of the edge of the text characters does not falls on itself, edges belonging to text are not removed. With this elimination, the performance of the proposed method is improved in terms of a number of metrics considered in this work. For instance, text detection results before edge elimination and after edge elimination are given in Figure 10 where (a) refers to the segmented portion, (b) is the result of Canny

edge detection before edge elimination, (c) is the edge map after edge elimination, (d) is the result of text detection before edge elimination and (e) is the result of text detection after edge elimination. Figure 10(e) shows that text lines are properly detected with four bounding boxes for the given image and Figure 10(d) shows that text lines are detected with two big bounding boxes. Therefore, edge elimination helps in improving the performance of the proposed method. The effect of edge elimination can be seen in Figure 10(c) by comparing to Figure 10(b). However, some times this eliminates text characters when characters are connected to each other. This leads to more misdetection rate. This can be noticed from Figure 10(e) where some characters are not covered by the bounding boxes at the bottom text line.



Figure 9. Sample edges that satisfies the condition

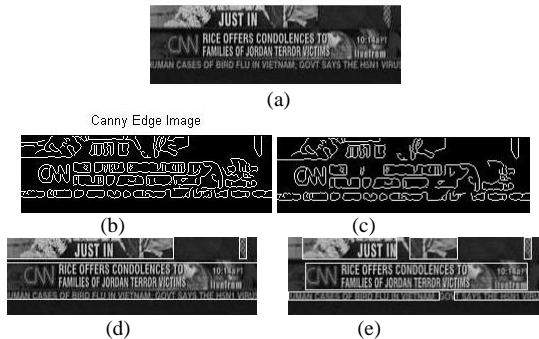


Figure 10. Bounding boxes for text lines with and without edge elimination

2.4. Projection Profile Analysis for Text Detection

The method obtains projection profiles of edge pixels after edge elimination for the segmented text portion. The method uses high peaks no less than 10 pixels and distance between the characters no less than 13 pixels to fix the boundary for the text lines. These values are estimated through experiments. This procedure is common for most text detection methods. Finally, a block is eliminated if the block is too tall, too small, or height to width aspect ratio is greater than 1.

3. Experimental Results

For experimental purpose, we created our own dataset as there is no standard dataset available in the literature. In this dataset, we included a variety of video images, including 101 frames taken from movies, news clips, sports videos and music videos. There are both graphic text and scene text in the video images.

3.1. Experimental Results for Candidate Text Block Selection

We evaluate the performance of the algorithm of candidate text block selection by considering accuracy as the metric. The accuracy is defined as the number of images for which candidate text block has been correctly chosen divided by the total number of images. The method successfully identifies candidate text blocks for 93 images out of 101 images. Therefore, the accuracy is 92%. Sometimes, there is a necessity of choosing two candidate text blocks when image

contains text in different parts as shown one example in Figure 11 where (b) and (c) are the two candidate text blocks for the image in Figure 11(a). Furthermore, we also present a sample of wrong choice of candidate blocks in Figure 11 where 11(d) refers to actual image and 11(e) refers wrong candidate text block choice. For images in Figure 11 (f) and 11(g), the method fails to choose any candidate text blocks.

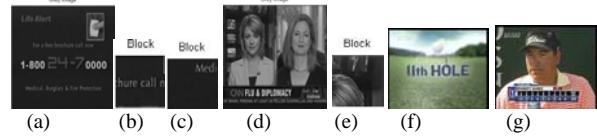


Figure 11. Performance of the candidate text block selection algorithm

3.2. Experimental Results for Segmentation

Sometimes, the method fails to segment the complete portion because of the complex background and isolated texts. For example, Figure 12(a) and (c) refer to actual images, and (b) and (d) refer to the incomplete segmented text portions for corresponding to images in 12(a) and 12(c).



Figure 12. Poor results of segmentation

3.3. Experimental Results for Text Detection

The detected text blocks are represented by their bounding boxes. To judge the correctness of the text blocks detected, we manually count true text blocks existing in the images in the dataset. Also we manually label each of the detected blocks as one of the following categories:

- **Correctly detected text block:** a detected block that contains text.
- **Falsely detected text block:** a detected block that does not contain text.
- **Text block with missing data:** a detected text block that does not include some characters.
- **Text block with inaccurate boundary:** a detected text block whose boundary is wider than the true bounding box of the text block.

Based on the number of blocks in each of the categories mentioned above, the following metrics are calculated to evaluate the performance of the method:

- **Detection rate** = Number of correctly detected text blocks / Existing number of text blocks
- **False positive rate** = Number of falsely detected text blocks / Number of detected text blocks
- **Misdetection rate** = Number of text blocks with missing data / Number of correctly detected text blocks
- **Inaccurate boundary rate** = Number of text blocks with inaccurate boundary / Number of correctly detected text blocks.

The experimental results of the proposed and existing methods are given in Figure 13 where (a) is the result of text detection, (b) is the result of an existing method [6]. It is noticed from Figure 13(b) that the existing method fails to detect small font and text with poor contrast. On the other hand, the proposed method detects them successfully. Similarly, Figure 13(c) is the result of the text detection, (d) is the result of the existing method for another video image. It is observed from Figure 13(d) that the existing method gives more false alarms, misdetection and inaccurate boundary

comparing to Figure 13(c). On the other hand, the proposed method gives better results with more accurate boundary.



Figure 13. Results of the proposed and existing method

The proposed method still has some limitations. The boundary adjustment step in the detection process is affected by complex environment surrounding the text blocks, causing some boundaries to be inaccurately specified, as shown in Figure 14(a) and (b).



Figure 14. Poor results given by the proposed method

Table 1. Result of the proposed and existing methods

	Text blocks	Detected	False alarm	Misdetction	Inaccurate boundary
ICDAR 2005 [6]	479	383	84	77	83
Proposed Method	479	429	48	69	42

Table 2. Performance of the proposed and existing methods

	Detected	False alarm	Misdetction	Inaccurate boundary	Time in minute
ICDAR 2005[6]	79.9%	17.9%	20.1%	21.6%	72
Proposed Method	89.5%	10.6%	17.1%	10.4%	17

4. Comparative Study

In order to evaluate the performance of the proposed method, we compare the results of the proposed method with the results of the existing method [6] using the set of metrics mentioned above. The results of this existing method (denoted as “ICDAR 2005”) using the same dataset are also summarized in Table 1 and Table 2.

We can see that the number of blocks detected by the proposed method is greater than the existing method, because the proposed method detects small font and text with poor contrast, thus resulting in a higher detection rate. However, the existing method ignores text lines with small font and poor contrast. Furthermore, the proposed method detects more accurate boundary for detected text blocks. Thus both the misdetection rate and the inaccurate boundary rate of

the proposed method are lower than those of the existing method. Furthermore, the proposed method gives lower false positive rate, because the projection profile of the edges after edge elimination provides a good guidance for finding the text lines. In addition, the proposed method takes very less time comparing to the existing method because the proposed method uses the segmented text portions instead of the whole image. However, processing time depends on data structure and platform used by the method. Here we used MATLAB for implementation purpose. Therefore the method takes minutes to detect text in the whole dataset.

5. Conclusion

In this paper, we proposed a new edge based method for detecting both graphic text and scene text in video images efficiently. The proposed method is based on the candidate text block selection and segmentation of text portion of the image. The experimental results reveal that the edge elimination step improves detection rate, reduces false alarms, misdetection, and inaccurate boundary rates compared to an existing method. Hence, the proposed method outperforms the existing method in terms of the above metrics. The method can be extended further to detect text lines with arbitrary direction. The robustness of the method handling complex backgrounds is to be improved as well.

Acknowledgment

This research is supported in part by IDM R&D grant R252-000-325-279.

References

- [1] K. Jung, K. I. Kim and A. K. Jain. Text information extraction in images and video: a survey. *Pattern Recognition*, 37(5): 977-997, 2004.
- [2] Y. Zhong, H. Zhang and A. K. Jain. Automatic Caption Localization in Compressed Video. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(4): 385-392, April 2000.
- [3] A. K. Jain and B. Yu. Automatic Text Location in Images and Video Frames. *Pattern Recognition*, 31(12): 2055-2076, 1998.
- [4] C. W. Lee, K. Jung and H. J. Kim. Automatic text detection and removal in video sequences. *Pattern Recognition Letters*, 24: 2607-2623, 2003.
- [5] Q. Ye, Q. Huang, W. Gao and D. Zhao. Fast and robust text detection in images and video frames. *Image and Vision Computing*, 23: 565-576, 2005.
- [6] C. Liu, C. Wang and R. Dai. Text Detection in Images Based on Unsupervised Classification of Edge-based Features. *IEEE ICDAR*, pp. 610-614, 2005.
- [7] E. K. Wong and M. Chen, “A new robust algorithm for video text extraction”, *Pattern Recognition*, 36: 1397-1406, 2004.
- [8] Q. Ye, W. Gao, W. Wang and W. Zeng. A Robust Text Detection Algorithm in Images and Video Frames. *IEEE ICICS-PCM*, pp. 802-806, 2003.
- [9] V. Y. Mariano and R. Kasturi. Locating Uniform-Colored Text in Video Frames. *IEEE 15th ICPR*, 4:539-542, 2000.
- [10] S. Antani, D. Crandall and R. Kasturi. Robust Extraction of Text in Video. *IEEE 15th ICPR*, 1: 831-834, 2000.
- [11] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Person Education, 2002.
- [12] P. Shivakumara, S. Noushat and G. H. Kumar. New Filter Based Unsupervised Rules for Boolean Blur Metric. *Intl. Conf. on Computing Theory and Applications (ICCTA)*, Kolkata, India, pp 611-617, March 2007.