2009 10th International Conference on Document Analysis and Recognition

A Laplacian Method for Video Text Detection

Trung Quy Phan, Palaiahnakote Shivakumara and Chew Lim Tan
School of Computing, National University of Singapore
{phanquyt, shiva, tancl}@comp.nus.edu.sg

Abstract

In this paper, we propose an efficient text detection method based on the Laplacian operator. The maximum gradient difference value is computed for each pixel in the Laplacian-filtered image. K-means is then used to classify all the pixels into two clusters: text and non-text. For each candidate text region, the corresponding region in the Sobel edge map of the input image undergoes projection profile analysis to determine the boundary of the text blocks. Finally, we employ empirical rules to eliminate false positives based on geometrical properties. Experimental results show that the proposed method is able to detect text of different fonts, contrast and backgrounds. Moreover, it outperforms three existing methods in terms of detection and false positive rates.

1. Introduction

There is an increasing number of video databases on the Internet and a reliable source of information for searching and retrieval is the text that appears in the videos. Video text consists of two types: graphic text and scene text. Graphic text is artificially added to the video during the editing process. Scene text appears naturally in the scenes captured by the camera. Although many methods have been proposed over the past years, text detection is still a challenging problem because videos often have low resolution and complex backgrounds and text can be of different sizes, styles and alignments. In addition, scene text is usually affected by lighting conditions and perspective distortions [1 – 3].

Text detection methods can be classified into three approaches: connected component-based [4], edge-based [5 – 9] and texture-based [10 – 14]. The first approach does not work well for all video images because it assumes that text pixels in the same region have similar colors or grayscale intensities. The second approach requires text to have a reasonably high contrast to the background in order to detect the edges. So these methods often encounter problems with complex backgrounds and produce many false positives. Finally, the third approach considers text as a special texture and thus, uses fast Fourier transform, discrete cosine transform, wavelet decomposition and Gabor filters for feature extraction. However, these methods require extensive training and are computationally expensive for large databases.

In this paper, we consider three existing methods [7, 8, 15] for comparative study. Liu et al. [7] extract edge features by using the Sobel operator. This method is able to determine the accurate boundary of each text block. However, it is sensitive to the threshold values for edge detection. Wong et al. [8] compute the maximum gradient difference values to identify candidate text regions. This method has a low false positive rate but uses many threshold values and heuristic rules. Therefore, it may only work well for specific datasets. Finally, Mariano et al. [15] perform clustering in the L*a*b* color space to locate uniform-colored text. Although it is good at detecting low contrast text and scene text, this method is extremely slow and produces many false positives.

We propose a text detection method which consists of three steps: text detection, boundary refinement and false positive elimination. In the first step, we identify candidate text regions by using the Laplacian operator. The second step uses projection profile analysis to determine the accurate boundary of each text block. Finally, false positives are removed based on geometrical properties. Experimental results show that the proposed method outperforms the above three methods in terms of detection and false positive rates.

2. Proposed method

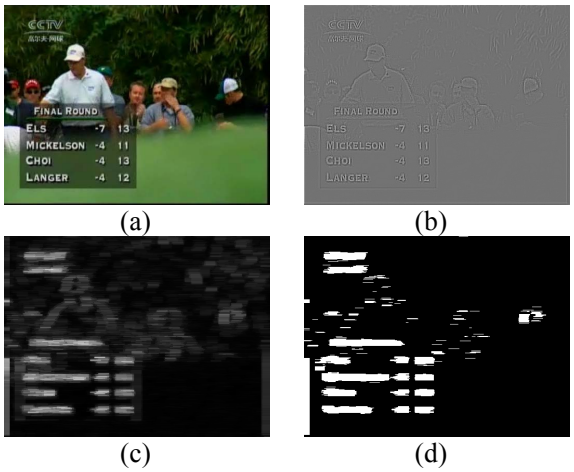
2.1. Text detection

Text regions typically have a large number of discontinuities, e.g. transitions between text and

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.153

1	1	1
1	-8	1
1	1	1

Figure 1. The 3 × 3 Laplacian mask.



(a) (b) (c) (d)

Figure 2. The text detection step. (a) Input image. (b) Laplacian-filtered image. (c) Maximum gradient difference map. (d) Text cluster.

background. Therefore, the input image is converted to grayscale and filtered by a 3 × 3 Laplacian mask to detect the discontinuities in four directions: horizontal, vertical, up-left and up-right (Figure 1).

Because the mask produces two values for every edge, the Laplacian-filtered image contains both positive and negative values. The transitions between these values (the zero crossings) correspond to the transitions between text and background. In order to capture the relationship between positive and negative values, we use the maximum gradient difference (MGD), defined as the difference between the maximum and minimum values within a local 1 × N window [8]. The MGD value at pixel (i, j) is computed from the Laplacian-filtered image f as follows.

$$MGD(i,j) = \max(f(i,j-t)) - \min(f(i,j-t)) \quad (1)$$

where $t \in \left[-\frac{N-1}{2}, \frac{N-1}{2}\right]$. The MGD map is obtained by moving the window over the image. In Figure 2c, brighter colors represent larger MGD values.

Text regions typically have larger MGD values than non-text regions because they have many positive and negative peaks (Figure 3). Therefore, we normalize the MGD map to the range [0, 1] and use K-means to classify all the pixels into two clusters, text and non-text, based on the Euclidean distance between MGD

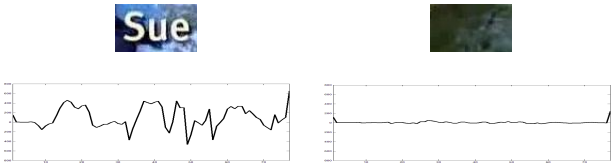


Figure 3. Sample profiles of text and non-text regions. Each graph shows the positive and negative values of the middle row of the corresponding Laplacian-filtered image (not shown here).

values. Let the two clusters returned by K-means be C_1 (cluster mean M_1) and C_2 (cluster mean M_2). Since the cluster order varies for different runs, we have the following rule to identify the text cluster. If $M_1 > M_2$, C_1 is the text cluster; otherwise, C_2 is the text cluster. This is because it is expected that text regions have larger MGD values than non-text regions. At the end of this step, each connected component in the text cluster is a candidate text region (Figure 2d).

2.2 Boundary refinement

It is difficult to determine the boundary of each text block directly from the text cluster because of false positives and connected text lines (Figure 4b). Therefore, we compute the binary Sobel edge map SM of the input image (only for text regions) (Figure 4c). The horizontal projection profile is defined as follows.

$$HP(i) = \sum_{j} SM(i,j) \quad (2)$$

If $HP(i)$ is greater than a certain threshold, row i is part of a text line; otherwise, it is part of the gap between different text lines. From this rule, we can determine the top row i_1 and bottom row i_2 of each text line. The vertical projection profile is then defined as follows.

$$VP(j) = \sum_{i=i_1}^{i_2} SM(i,j) \quad (3)$$

Similarly, if $VP(j)$ is greater than a certain threshold, column j is part of a text line; otherwise, it is part of the gap between different words. Finally, different words on the same text line are merged if they are close to each other.

By applying this step recursively, we can determine the accurate boundary of each text block, even when the text blocks are not well-aligned or when one candidate text region contains multiple text lines. At the end of this step, each detected block is a candidate text block (Figure 4d).

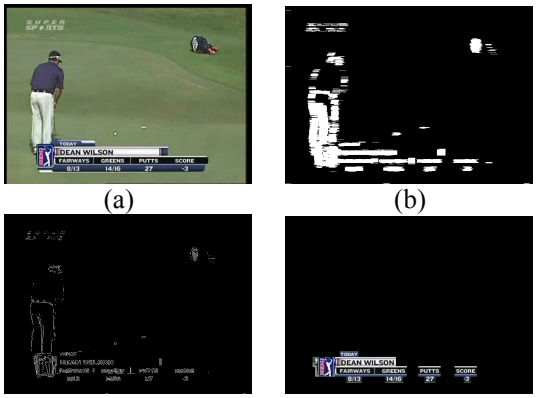


Figure 4. The boundary refinement step. (a) Input image. (b) Text cluster. (c) Sobel edge map. (d) Text blocks.

2.3. False positive elimination

We eliminate false positives based on geometrical properties. Let W, H, AR, A and EA be the width, height, aspect ratio, area and edge area of text block B.

$$AR = W / H \tag{4}$$

$$A = W \times H \tag{5}$$

$$EA = \sum_{(i,j) \in B} SM(i, j) \tag{6}$$

If $AR < T_1$ or $EA / A < T_2$, the candidate text block is considered as a false positive; otherwise, it is accepted as a text block. The first rule checks whether the aspect ratio is below a certain threshold. The second rule assumes that a text block has a high edge density due to the transitions between text and background.

3. Experimental results

As there is no standard dataset available, we have selected 101 video images, extracted from news programmes, sports videos and movie clips, for our own dataset. There are both graphic text and scene text of different languages, e.g. English, Chinese and Korean, in the dataset. The image sizes range from 320 × 240 to 816 × 448. The parameter values are empirically determined: $N = 5$, $T_1 = 0.5$ and $T_2 = 0.1$.

For comparison purpose, we have implemented three existing methods [7, 8, 15]. Method [7], denoted as *edge-based method*, extracts edge features by using the Sobel operator. Method [8], denoted as *gradient-based method*, computes the MGD values to identify candidate text regions. Finally, method [15], denoted as *uniform-colored method*, performs clustering in the L*a*b* color space to locate the text lines.

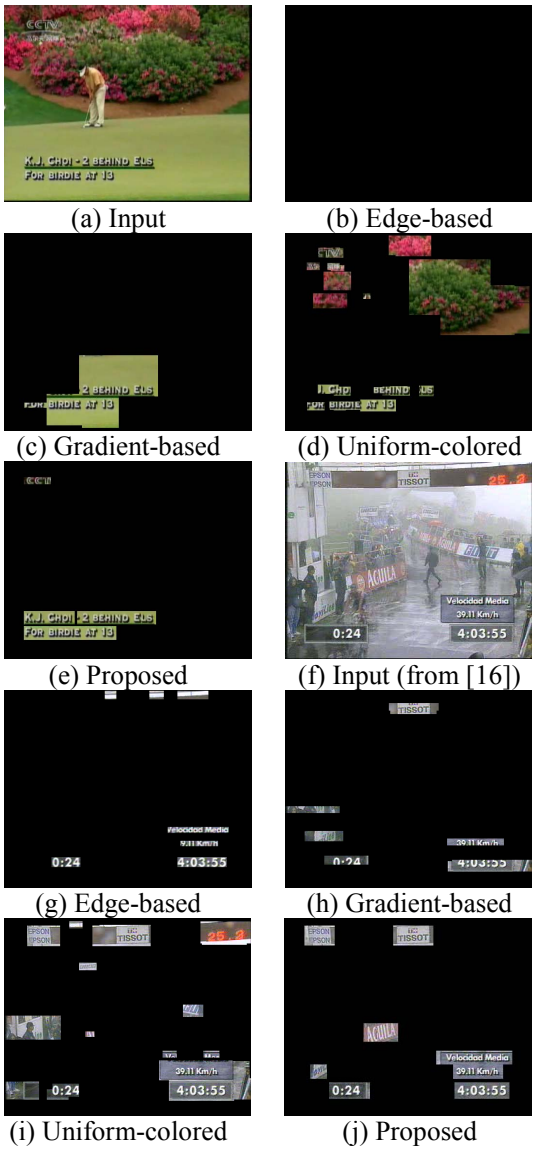


Figure 5. The detected text blocks of the three existing methods and the proposed method for input images (a) and (f).

3.1 Sample results

Figure 5 shows some sample results of the three existing methods and the proposed method. Image (a) has two low contrast text blocks. The edge-based method fails to detect any text block because of the problem of fixing threshold values for edge detection. The gradient-based method detects the text blocks with missing characters and inaccurate boundary. This method uses many threshold values and heuristic rules and thus, may only work well for specific datasets. The

uniform-colored method detects the text blocks with missing characters and produces many false positives due to the problem of color bleeding. The proposed method detects all the blocks correctly and even picks up the low contrast logo of the television channel.

Image (f) has both graphic text (at the bottom left and right corners) and scene text (at the top). The edge-based method detects the graphic text but misses the scene text. The gradient-based method also misses some graphic text (the first graphic text line) and scene text (the two scene text blocks at the top left and right corners). The uniform-colored method produces many false positives. The proposed method detects all the text blocks correctly, except one at the top right corner. One of the billboards on the road is also detected.

Figure 6 shows an image where the proposed method fails to detect some text blocks. The red text on the blue background is not detected because there is very low contrast between these two colors in the grayscale domain. The edge-based method and the gradient-based method have the same problem because they also use the grayscale image. By using the color information, the uniform-colored method is able to detect one of the two red text lines ("Life Alert").

Figure 7 shows the results of the proposed method for two different window sizes. A small window size gives a low false positive rate but might miss some low contrast characters (on the third line) (image (b)). On the other hand, a large window size helps to recover missing characters but also includes more false positives (image (c)). In our experiment, N is set to 5.

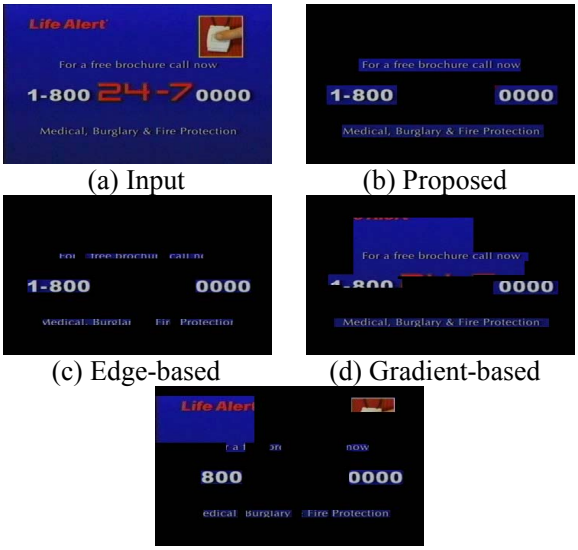
3.2 Results on the dataset

We define the following categories for each detected block by a text detection method.

- *Truly Detected Block (TDB)*: A detected block that contains a text line, partially or fully.
- *False Detected Block (FDB)*: A detected block that does not contain text.
- *Text Block with Missing Data (MDB)*: A detected block that misses some characters of a text line (MDB is a subset of TDB).

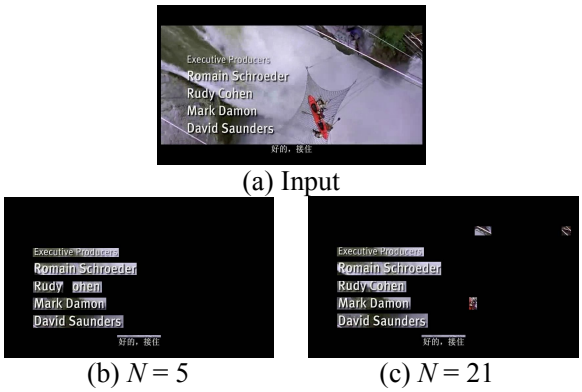
For each image in the dataset, we manually count the *Actual Text Blocks (ATB)*, i.e. ground truth data. The performance measures are defined as follows.

- *Detection Rate (DR)* = TDB / ADB
- *False Positive Rate (FPR)* = FDB / (TDB + FDB)
- *Misdetection Rate (MDR)* = MDB / TDB



(a) Input (b) Proposed (c) Edge-based (d) Gradient-based (e) Uniform-colored

Figure 6. The proposed method fails to detect some text blocks because the contrast is too low.



(a) Input

(b) $N = 5$ (c) $N = 21$

Figure 7. Results of different window sizes.

Tables 1 and 2 show the performance of the three existing methods and the proposed method on the dataset. The proposed method has the highest DR and lowest FPR. It outperforms the edge-based method and the uniform-colored method in all the performance measures.

Compared to the gradient-based method, the proposed method has better DR and FPR but worse MDR. However, the slightly higher MDR might be compensated by the significant difference in DR between the two methods. If we consider the number of fully detected text blocks, i.e. text blocks which do not have any missing character, the proposed method detects 458 – 55 = 403 blocks while the gradient-based method only detects 349 – 35 = 314 blocks.

Therefore, the proposed method has achieved better detection results than the three existing methods on the dataset.

Table 1. Results on the dataset.

Method	ATB	TDB	FDB	MDB
Edge-based [7]	491	393	86	79
Gradient-based [8]	491	349	48	35
Uniform-colored [15]	491	252	95	94
Proposed	491	458	39	55

Table 2. Performance on the dataset.

Method	DR	FPR	MDR
Edge-based [7]	80.0	18.0	20.1
Gradient-based [8]	71.1	12.1	**10.0**
Uniform-colored [15]	51.3	27.4	37.3
Proposed	**93.3**	**7.9**	12.0

4. Conclusion and future work

We have proposed an efficient method for text detection based on the Laplacian operator. The gradient information helps to identify the candidate text regions and the edge information serves to determine the accurate boundary of each text block. Experimental results show that the proposed method outperforms the three existing methods in terms of detection and false positive rates.

In the future, we plan to extend this method to text of arbitrary orientation. Currently, the text detection step can show white patches even for non-horizontal text (Figure 8). However, the refinement step is only able to detect the boundary for horizontal text because of the use of horizontal and vertical projection profiles.

Figure 8. The text detection step is able to show white patches for non-horizontal text.

5. Acknowledgment

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

6. References

[1] J. Zang and R. Kasturi, "Extraction of Text Objects in Video Documents: Recent Progress", *The Eighth IAPR Workshop on Document Analysis Systems* (DAS2008), Nara, Japan, September 2008, pp 5-17.

[2] J. Zhang, D. Goldgof and R. Kasturi, "A New Edge-Based Text Verification Approach for Video", *ICPR*, December 2008, pp 1-4.

[3] K. Jung, K.I. Kim and A.K. Jain, "Text information extraction in images and video: a survey", *Pattern Recognition*, 37, 2004, pp. 977-997.

[4] A.K. Jain and B. Yu, "Automatic Text Location in Images and Video Frames", *Pattern Recognition*, Vol. 31(12), 1998, pp. 2055-2076.

[5] M. Anthimopoulos, B. Gatos and I. Pratikakis, "A Hybrid System for Text Detection in Video Frames", *The Eighth IAPR Workshop on Document Analysis Systems* (DAS2008), Nara, Japan, September 2008, pp 286-293.

[6] 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, February 2005, pp 243-255.

[7] C. Liu, C. Wang and R. Dai, "Text Detection in Images Based on Unsupervised Classification of Edge-based Features", *ICDAR* 2005, pp. 610-614.

[8] E. K. Wong and M. Chen, "A new robust algorithm for video text extraction", *Pattern Recognition* 36, 2003, pp. 1397-1406.

[9] P. Shivakumara, W. Huang and C. L. Tan, "An Efficient Edge based Technique for Text Detection in Video Frames", *The Eighth IAPR Workshop on Document Analysis Systems* (DAS2008), Nara, Japan, September 2008, pp 307-314.

[10] Y. Zhong, H. Zhang and A.K. Jain, "Automatic Caption Localization in Compressed Video", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 22, No. 4, 2000, pp. 385-392.

[11] K. L Kim, K. Jung and J. H. Kim, "Texture-Based Approach for Text Detection in Images using Support Vector Machines and Continuous Adaptive Mean Shift Algorithm", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 12, December 2003, pp 1631-1639.

[12] Q. Ye, Q. Huang, W. Gao and D. Zhao, "Fast and robust text detection in images and video frames", *Image and Vision Computing* 23, 2005, pp. 565-576.

[13] H. Li, D. Doermann and O. Kia, "Automatic Text Detection and Tracking in Digital Video", *IEEE Transactions on Image Processing*, Vol. 9, No. 1, January 2000, pp 147-156.

[14] W. Mao, F. Chung, K. K. M. Lam and W. Siu, "Hybrid Chinese/English Text Detection in Images and Video Frames", *ICPR*, Volume 3, 2002, pp 1015- 1018.

[15] V. Y. Mariano and R. Kasturi, "Locating Uniform-Colored Text in Video Frames", *15th ICPR*, Volume 4, 2000, pp 539-542.

[16] X. S. Hua, W. Liu and H. J. Zhang, "Automatic Performance Evaluation for Video Text Detection", *ICDAR*, 2001, pp 545-550.