Graph-based Background Suppression For Scene Text Detection

Cunzhao Shi, Baihua Xiao, Chunheng Wang, Yang Zhang
State Key Laboratory of Intelligent Control and Management of Complex Systems
Institute of Automation, Chinese Academy of Sciences
Beijing, China
{cunzhao.shi, baihua.xiao, chunheng.wang, yang.zhang}@ia.ac.cn

***Abstract*—Detecting text in video or natural scene image is quite challenging due to the complex background, various fonts and illumination conditions. The preprocessing period, which suppresses the nontext areas so as to highlight the text areas, is the basis for further text detection. In this paper, a novel graph-based background suppression method for scene text detection is proposed. Considering each pixel as a node in the graph, our approach incorporates pixel-level and context-level features into a graph. Various factors contribute to the unary and pairwise cost function which is optimized via max-flow/min-cut algorithm [16] to get a binary image whose nontext pixels are suppressed so that text pixels are highlighted. Furthermore, the proposed background suppression method could be easily combined with other detection methods to improve the performance. Experimental results on ICDAR 2011 competition dataset show promising performance.**

Keywords-background suppression; text detection; graph; edge detection; region-based classifier.

I. INTRODUCTION

With the widely use of various digital image capturing devises, efficient content-based image analysis techniques are necessary in applications such as web image searching and retrieval, license plate recognition, sign reading and so on. As text in images or video could provide exact and unique information about the content, detecting, extracting and recognizing text is receiving more and more attention in the recent years as surveyed in [5], [6], [7].

As shown in Fig. 1, text detection process includes the preprocessing, the connected components (CCs) analysis and text components grouping stages. In this paper, we focus on the preprocessing stage which suppresses most of the background pixels so as to highlight text pixels. Most of the

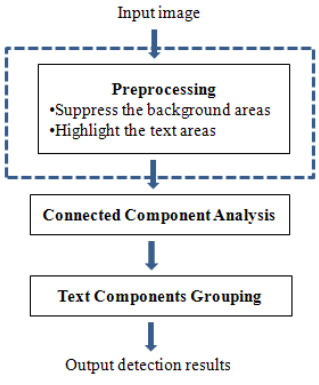


Figure 1. Illustration of detection process.

preprocessing methods for text detection could be classified into three categories: region-based, color-based and edge-based.

Region-based methods consider that text has distinct textural features, and apply various approaches such as Fast Fourier Transform and wavelet decomposition to exact textural features which are then fed into a classifier to suppress nontext areas. Ye et al. [12] compute features from wavelet decomposition coefficients at different scales and apply adaptive threshold to suppress background pixels. Lee et al. [14] use neighboring grayscale values as features for each pixel which are classified by SVM as text or nontext. Chen and Yuille [13] propose a fast text detector based on a cascade AdaBoost classifier to exclude nontext areas. Pan et al. [1] propose to preprocess the image based on the text confidence map and scale map and the hybrid detection method performs competitively on ICDAR 2005 competition dataset [15]. However, region-based methods require a large number of training set of text and nontext samples and it is especially hard to make sure the nontext samples are representative.

Color-based methods assume characters in the same text region have uniform color and employ color quantization or color clustering to group pixels of the similar colors into connected components (CCs) [8], [9] so as to suppress the background pixels. However, color-based methods might fail to perform well when text and background have similar color.

Edge-based methods are based on the assumption that text has a high contrast to its background for reading. Liu et al. [10] exact features for each pixel from Sobel edge maps of four directions and use K-means to suppress background pixels. Lyu et al. [4] use a background-complexity-adaptive local threshold algorithm to suppress the background edges and design a text-like area recovery filter to recover the text edges. However, it might not generalize well on other dataset due to the various parameters. Recently, Epshtein et al. [11] propose to use the stroke width transform to calculate the stroke width of each pixel and suppress those

As text does have distinct textural features, relatively uniform color and high contrast with the background, we propose a background suppression method for text detection integrating various factors into a graph. To some extent, the proposed method combines the advantages of region-based, color-based and edge-based methods. To this end, the background suppression process is formulized as a cost function minimization problem for all the pixels. The cost function is composed of unary and pairwise cost. The unary cost is defined as the tradeoff between the gradient value and the text classification map, whereas the pariwise cost is the

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.40

IEEE computer society

feature distance between neighboring pixels which reflects the connectivity constraints. By optimizing the cost function via max-flow/min-cut algorithm [16], a binary image whose nontext pixels are suppressed and text pixels are highlighted, is acquired. Furthermore, the approach could be easily combined with various text detection methods to further suppress nontext pixels.

The rest of the paper is organized as follows. Section Ⅱ details the proposed method. Experiments and results are presented in Section Ⅲ and conclusions are drawn in Section Ⅳ.

II. THE PROPOSED METHOD

The flowchart of the method is shown in Fig. 2. The proposed method focuses on the preprocessing stage which suppresses the nontext pixels so that text pixels are highlighted. First, the gradient map and the text classification map are calculated based on which we define the cost function for all the pixels whose unary and pairwise cost function not only reflect the pixel-level and context-level features but also combine edge, color and texture features. By optimizing the cost function via max-flow/min-cut algorithm [16], we get a binary image ready for text detection. Furthermore, by changing the parameter ε in (5), the proposed method could be combined with various text detection methods to further suppress the nontext areas.

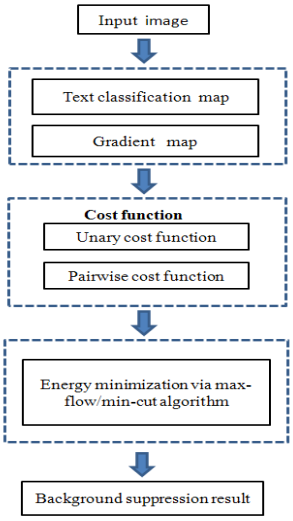


Figure 2. Flowchart of the proposed system.

A. Problem Formulation

In the preprocessing stage, most text detection methods either apply edge detector (Sobel, Canny[17]), text region classifier, or color clustering to get the candidate text components. While in this paper, we aim to integrate all the factors into a graph. An undirected graph $G=\{V,E\}$ is composed of nodes (vertices V) and undirected edges (E) that connect these nodes [18]. Each pixel in the image is considered as a node in the graph, and edges are composed of the standard 8 neighborhood system. Thus the preprocessing stage could be formulized as a segmentation problem by labeling the interested text areas as 1 (foreground) and other areas as 0 (background). Let P be all the pixels in the image and N be a set of neighboring pairs $\{p,q\}$ in P. $L=\{L_1,L_2...,L_p...\}$ is a binary vector whose components L_p specify the labels of pixel p in P. Each L_p is either 1 (foreground) or 0 (background). The cost function $E(L)$ for each segmentation L is defined as [18]:

$$E(L)=\lambda U(L)+B(L), \tag{1}$$

where

$$U(L)=\sum_{p\in P}U_p(L_p) \tag{2}$$

$$B(L)=\sum_{\{p,q\}\in N}B_{\{p,q\}}\cdot\delta(L_p,L_q) \tag{3}$$

and

$$\delta(L_p,L_q)=\begin{cases}1 & if\ L_p\neq L_q\\ 0 & otherwise.\end{cases} \tag{4}$$

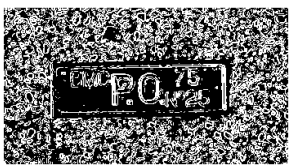
The coefficient $\lambda>=0$ is a trade-off parameter for the unary cost $U(L)$ and the pairwise cost $B(L)$. Thus, the target of background suppression is to find a segmentation that minimizes the cost function. In the following sections, we will give details about the unary and pairwise cost functions which integrate different features.

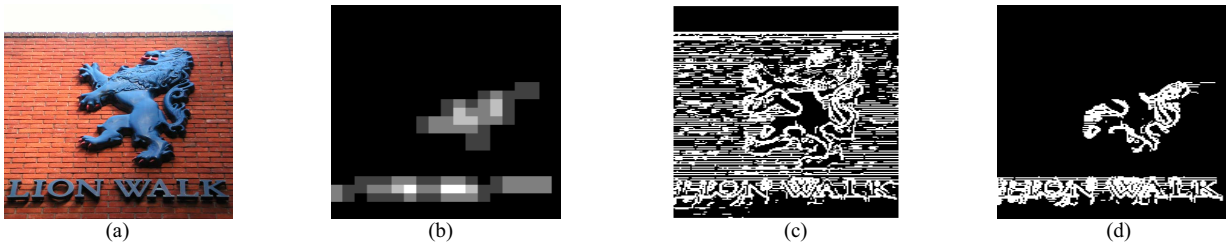
B. Unary Cost Function

Unary cost function $U(L)$ measures the individual penalties for labeling pixel p as foreground or background and each pixel has two cost weights $U_p(1)$ and $U_p(0)$, corresponding to linking cost to foreground and background respectively. In this paper, we define cost function as the weighted sum of gradient-based function and the text classification map. For each pixel p, large gradient value or text classification result should correspond to small linking cost $U_p(1)$ whereas small gradient or prediction result should correspond to large $U_p(0)$.

1) Gradient Map: The gradient map reflets the edge-based feature for each pixel. The gradient value of each





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

Figure 3. Examples of the background suppression process: (a) Input. (b) Text classification map. (c) Edge map by Otsu threshold of the Sobel map. (d) Background suppression result by the proposed method.

pixel is the bigger one computed by the Sobel operator in two directions (horizontal and vertical).

2) Text Classification Map: The value for each pixel in the classification map represents the probabilities of the pixel being text. As the classification map together with the gradient map contributes to the unary cost, we don't need to classify each pixel precisely. Thus our method is region-based classification rather than pixel-based which is time consuming. Next, we will give specific descriptions of the overall scheme, the features and classifiers.

a) Overall scheme: We scan the image with a fixed window of size 16-by-16 and a step of 8-by-8 at multiple scales. Each window is classified as text or nontext. If the region is classified as text, add 1 on the classification map for all the pixels in the scanned window. Then, divide the classification map by the times of the pixels being classified to get the normalized classification map for each scale. Finally, the final value of each pixel in the classification map is the largest one of all the scales. Images in the second chum in Fig. 3 are examples of the text classification maps.

b) Features: 8-orientation histograms of oriented gradients (HOG) [19] features are exacted from each scanned region. As each region is partitioned into 2-by-2 blocks, the total feature dimensions are 32.

c) Classifier: We choose Random Forests [23] as the region-based classifier due to its fast speed and relatively better generalization performance. For text and nontext classification, the main problem is the difficulty to make the nontext samples representative. To address this problem, in addition to the ICDAR 2011 training dataset [20], we collect nontext samples from fifteen scene categories dataset [22] which contain a large number of natural scenes.

Given the pixel-level gradient map and reigon-level text classification map which represents the context information, the unary cost funciton for each pxiel is definded as

$$U_p(1)=\begin{cases} Max\_Cost, & if \quad G_p <= \varepsilon \\ \beta\cdot(\exp(\frac{-G_p^{\,2}}{2*\sigma_u^{\,2}}))+\alpha\cdot(1-\Pr_p), & otherwise, \end{cases} \tag{5}$$

$$U_p(0)=\begin{cases} 0, & if \quad G_p <= \varepsilon \\ \beta\cdot(\exp(\frac{-(1-G_p)^2}{2*\sigma_u^{\,2}}))+\alpha\cdot\Pr_p, & otherwise. \end{cases} \tag{6}$$

where Max_Cost is the maximum cost used in the graph, ε is the minimum acceptable gradient value for text, α, β are tradeoff parameters between the gradient value G_p and the text classification map $\Pr_p$, and σ_u is the precision factor. The gradient value and the text classification map are normalized in the range from 0 to 1. α, β and σ_u are set to 0.5, 0.5 and 0.25 by cross-validation while Max_Cost are set to 1000. The use of gradient threshold ε could greatly reduce computation expense as the cost of those pixels whose gradients are smaller than ε are directly set to a value without further computing. In the experiment, different edge detection methods will be combined with our methods by changing the threshold ε and the improvement of background suppression results will be shown.

C. Pairwise Cost Function

The pairwise cost function $B(L)$ reflects penalties for discontinuity between neighboring pixels. $B(L)$ could be defined as a decreasing function of the feature distance between the neighboring pixels p and q, which means if the features of p and q are similar, the penalty $B_{\{p,q\}}$ for assigning different labels to the neighboring pixels p and q should be large and if the features are different, the penalty should be small. We use color and gradient features for each pixel and the cost function for each pair of neighboring pixels are defined as

$$B_{\{p,q\}}=\exp(-\frac{(Fea_p-Fea_q)^2}{2\sigma_B^{\,2}}), \tag{7}$$

where Fea_p and Fea_q are features for pixel p and q respectively. σ_B is the precision factor set to 0.25 by cross-validation.

D. Cost Function Minimization

Now that we have defined the unary and pairwise cost function for each pixel, given an input image, the total cost function for labeling all the pixels as foreground or background could be represented. The cost function could be minimized by finding the minimum cut of a graph whose nodes are the pixels and edges are the standard 8 neighboring system. The max-flow/min-cut algorithm [16] is used to optimize the cost function to get a binary image

whose background areas are suppressed while text areas are preserved. Fig. 3 shows the process.

Given the binary background suppression image, various methods could be used to group CCs into regions and heuristic rules or classifiers could be used to exclude nontext regions. However, as we focus on the background suppression period which plays an important role for text detection, the following stages in Fig. 1 is not our main concern in this paper. We only use the text detection result to evaluate the performance of the background suppression method. In fact, better preprocessing results will lead to better text detection results as most of the nontext areas are suppressed.

III. EXPERIMENTS

A. Dataset and Evaluation Methods

In order to evaluate the performance of the proposed background suppression method, 40 images from ICDAR 2011 scene text localization competition test dataset [20] are chosen as our test set. The texts in these images have different fonts, sizes, colors and illumination conditions.

We use the pixel-level and region-level text detection rates as the evaluation methods. For pixel-level evaluation, pixel classification rate (PCR) is defined as

$$PCR = N_{text} / N \tag{8}$$

where N_{text} and N are the numbers of detected text pixels and the total detected pixels respectively. As better background suppression result should correspond to bigger N_{text} and fewer nontext pixels, larger PCR means better performance. For region-level evaluation, we use the following measures to evaluate the performance:

$$\mathrm{Re}call(R) = DTB / TB, \tag{9}$$

$$\mathrm{Pr}ecisoin(P) = DTB / (DTB + DNB), \tag{10}$$

$$F = 2 * P * R / (P + R), \tag{11}$$

where TB, DTB and DNB are the numbers of actual text regions, the detected text regions and the detected nontext regions.

The proposed method is compared with the preprocessing stages of two edge-based text detection methods. Lyu et al. [4] proposed to use a local threshold algorithm to suppress the background edges and a text-like filter to recover the text edges. Liu et al. [10] extracted features from four edge maps and used k-means to cluster the pixels into text or nontext areas. Although the two methods have their own methods for the following text detection, in order to compare the background suppression performance, similar rules as that in [10] are used to group the CCs in the preprocessed image to text regions and exclude nontext regions.

B. Comparing Results with Other Methods

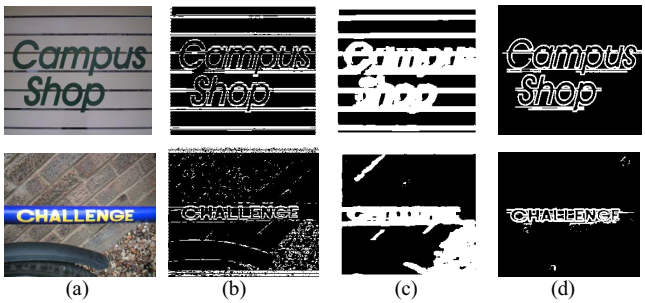
For computation efficiency, each input image is normalized to a height of 300 while maintaining the ratio between the width and height. The parameters in [4] and [10] are set to the best ones according to their paper. Three scales are used for method [4] and the best one is chosen. The threshold ε is set to 4/5 of the global Otsu [21] threshold of the gradient map. Results are shown in Table Ⅰ, where the PCR are the average PCR for the 40 images. We should point out that as our main purpose is to evaluate the background suppression performance and only part of the ICDAR 2011 test dataset are used, it's inappropriate to compare the results with the public competition results.

TABLE I. RESULTS OF DIFFERENT METHODS

Evaluation Methods	The Preprocessing Methods		
	Lyu's method [4]	*Liu's method [10]*	*Our method*
PCR (%)	17.7	17.4	**59.2**
R (%)	75.3	72.5	**88.7**
P (%)	60.5	80.4	**95.6**
F (%)	67.1	76.2	**92.0**

Our method takes 1 second on average for each image, a little slower than Lyu' method whereas much faster than Liu's method. From the results we can see that the proposed method performs better than the other two methods both in the pixel-level and region-level detection rates. This is quite reasonable as 1) the proposed method incorporates different factors such as gradient, color features and context information from a trained region-based classifier; and 2) better background suppression result removes most of the nontext area, making the text detection much easier.

Some of the preprocessing results of the three methods are shown in Fig. 4. As we can see, as Lyu's method only use heuristic rules, it fails to remove nontext areas with complex backgrounds. For Liu's method, as the statistical features are sensitive to the window size and the features might not be representative enough for text, nontext areas with strong textures are also detected in the second image. In contrast, the proposed method removes most of the nontext pixels while also maintaining the text pixels.



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

Figure 4. Background suppression results of three methods: (a) Input. (b) Lyu's method. (c) Liu's method. (d) The proposed method.

C. Results Combined with Edge Detection Methods

The proposed methods could be easily combined with other detection methods to further remove the nontext pixels. Here we combine it with two edge detection methods, Sobel edge detector with local threshold and Canny edge detector [7]. ε is set to the edge binarization threshold. The PCR results are shown in Table Ⅱ where BS is short for

background suppression. The result demonstrates that PCR increases a lot for both of the two edge detection methods when combined with the proposed background suppression method. Some of the results are shown in Fig. 5. As we can see, our method could successfully remove most of the nontext edges while also preserve the text edges.

TABLE II. RESULTS COMBINED WITH EDGE DETECTION METHODS

Evaluation Methods	Edge Detection Methods			
	Canny	*BS of Canny*	*Local Sobel*	*BS of Sobel*
PCR (%)	22.90	**50.14**	22.18	**56.15**

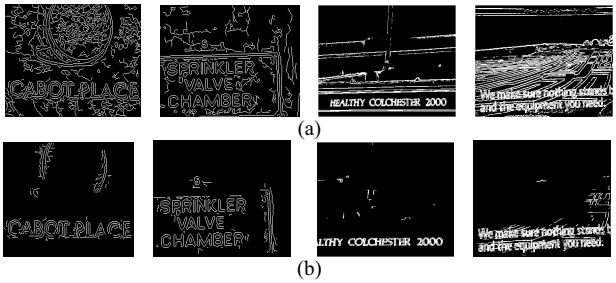


Figure 5. Background suppression with edge detection methods: (a) Canny edge map (the left two images) and Sobel edge map by local threshold (the right two images). (b) Background suppression results.

IV. CONCLUSION

In this paper, we propose a background suppression method for scene text detection. The main contribution of this paper is the framework we propose to integrate edge-based, color-based and region-based methods into one framework so as to take advantages of different methods. Experimental results demonstrate the proposed method could successfully remove most of the nontext pixels and also preserve the text pixels, making the text detection much easier. Furthermore, the approach could be easily combined with other detection methods to further suppress nontext areas.

Although the proposed method performs well, it still needs further improvements. In the future, we would like to get scale information from the text classification map and make the unary and pairwise cost function adaptive with context information for better background suppression performance so as to detect text with higher recall and precision.

ACKNOWLEDGMENT

This work is supported in part by the National Natural Science Foundation of China under Grant No. 60802055, No. 60933010 and No. 60835001.

REFERENCES

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

[2] X. Chen and A. Yuille, "Detecting and reading text in natural scenes," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'04), vol. 2,2004.

[3] P. Shivakumara, T. Q. Phan, and C. L. Tan, "A laplacian approach to multi-oriented text detection in video," IEEE Trans. Pattern Anal. Mach. Intell, vol. 33, no. 2, pp. 412 –419, feb. 2011.

[4] M. Lyu, J. Q. Song, and M. Cai, "A comprehensive method for multilingual video text detection, localization, and extraction," IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 2, pp. 243–255, 2005.

[5] K. Jung, K. I. Kim, and A. K. Jain, "Text information extraction in images and video: A survey," Pattern Recogniton, vol. 37, no. 5, pp. 977–997, 2004.

[6] J. Liang, D. Doermann, and H. P. Li, "Camera-based analysis of text and documents: A survey," Int. J. Document Anal. Recogn., vol. 7, no. 2-3, pp. 84–104, 2005.

[7] J. Zhang and R. Kasturi, "Extraction of text objects in video documents:Recent progress," in Proc. 8th IAPR Workshop on Document Analysis Syetems (DAS'08), Nara, Japan, 2008, pp. 1–13.

[8] Y. Zhong, K. Karu and A.K. Jain, "Locating Text in complex color Images," Pattern Recognition, vol. 28, no. 10, pp. 1,523-1,535, 1995..

[9] H.K. Kim, "Efficient automatic text location method and content-based indexing and structuring of video database," J. Visual Commun. Image Representation, 7 4 (1996), pp. 336–344.

[10] C. Liu , C. Wang and R. Dai, "Text detection in images based on unsupervised classification of edge-based features," in Proc.8th Int. Conf. Document Analysis and Recognition (ICDAR'05), Seoul, South Korea, 2005, pp. 610–614.

[11] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'10), 2010, pp. 2963 –2970.

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

[13] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'04), 2004, vol. 2, pp. II–366 – II–373 Vol.2.

[14] C.W. Lee, K. Jung and H.J. Kim, "Automatic text detection and removal in video sequences," Pattern Recognition Letters 24, 2003, pp. 2607-2623.

[15] S. M. Lucas, "ICDAR 2005 text locating competition results," in Proc.8th Int. Conf. Document Analysis and Recognition (ICDAR'05), Seoul, South Korea, 2005, pp. 80–84.

[16] Boykov, Y., Kolmogorov, V., "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," IEEE Trans. Pattern Anal. Mach. Intell, vol. 26, no. 9, pp. 1124 - 1137, Sept. 2004.

[17] J. Canny. A computational approach to edge detection. Readings in computer vision: issues, problems, principles, and paradigms, 184:87–116, 1987.

[18] Y.Y. Boykov and M.P. Jolly, "Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in n-d Images," Proc. IEEE Int. Conf. Computer Vision, pp. 105-112, 2001.

[19] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, 2005, pp. 886–893.

[20] http://robustreading.opendfki.de/wiki/SceneText.

[21] N. Otsu. A threshold selection method from gray-level histograms. Automatica, 11: 285–296, 1975.

[22] L. Fei-Fei and P.Perona, "A Bayesian hierarchical model for learning natural scene categories," In Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'05), 2005.

[23] L. Breiman. Random forests. Machine Learning, 45:5-32, 2001.