SVM OUTPUT SCORE BASED TEXT LINE REFINEMENT FOR ACCURATE TEXT LOCALIZATION

Cheolkon Jung, Qifeng Liu, Joongkyu Kim

School of Information and Communication Engineering, Sungkyunkwan University,
Suwon 440-746, Republic of Korea

ABSTRACT

In this paper, we propose the text line refinement method based on the SVM (support vector machine) output score for accurate text localization. In general, SVM output scores for the verification of text candidates provide a measure of the closeness to the text. Up to the present, most researchers used the score for the verification of the text candidate region. However, we use the output score for refining the initial text localization results. By means of the proposed approach, we can obtain more accurate text localization results. The effectiveness and efficiency of the proposed method is validated by extensive experiments on a complex database containing 435 images.

Index Terms— Text processing, text line refinement, the SVM output score, accurate text localization

1. INTRODUCTION

Text in images and video always carries rich useful information, which can help a computer to understand their content. So text localization is very important for many fields of automatic annotation, indexing and parsing of images and video [1].

Up to the present, many significant achievements have been made by researchers in the field of text localization [10, 11, 13]. In the early stages of the research in this area, the methods were comparatively simple, because the texts were detected and localized based on much heuristic information [2, 3, 12]. Although these methods were very fast, a large number of false alarms inevitably occurred. Recently, many researchers have focused their attention on the application of pattern classification to text localization based on elaborately selected features [4]. However, when we carefully observe the text localization results, we find that there are a lot of problems. Some of the text lines are not accurate as shown in Fig. 1.

For example, (1) some text candidates contain too much background, (2) some texts are missed by text candidate detection, and (3) some characters are divided into two text line boxes. These problems will have a detrimental effect on the recognition results.

(1) Included background (2) A missed text

(3) Text box fractions

Fig. 1. Problems of initial localization results.

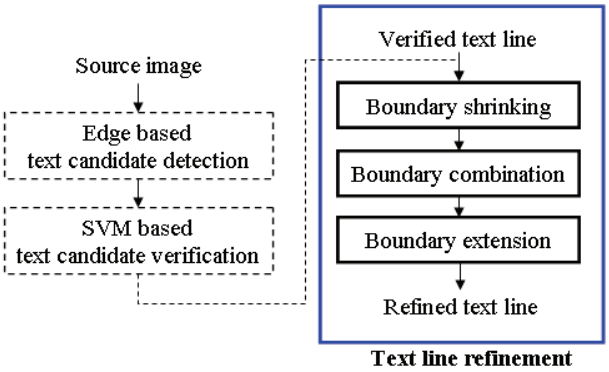

Fig. 2. Flow chart of text localization. The right part in the blue box is the focus of this paper.

In this paper, we propose a novel approach to solve these problems. Basically, we believe that the detected text boundaries should be refined by some strategies. The strategies may rely on some measurement, which can tell us how likely a given sub-image is text. Thereby, we use the SVM output score [5] and image similarity measurement [9] for the text line refinement, which have been successfully applied in the fields of computer vision and pattern recognition.

The input of the text line refinement is initial text localization results, which could be obtained by some methods such as [6] and [7]. As shown in Fig. 2, give a source image, text candidates are detected by edge extraction, morphologic operation and connected component analysis. Then these text candidates are verified by a SVM classifier trained in advance.

The remainder of this paper is organized as follows. The three modules of text boundary shrinking, combination and extension are described in Section 2, 3 and 4, respectively.

Fig. 3. Text boundary shrinking.

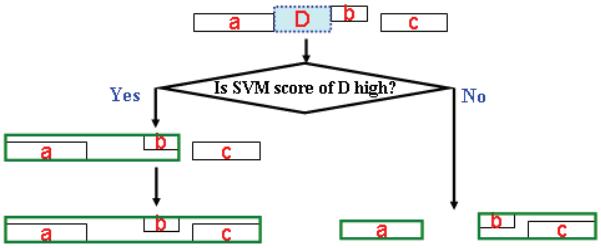

Fig. 4. Text boundary combination.

We give experiments in Section 5 and conclude this paper in Section 6.

2. TEXT BOUNDARY SHRINKING

For each initial text boundary, we used a sliding window to generate some samples. These samples are verified by a SVM classifier which was trained in advance [6]. Approximately, the output score of SVM can provide the quantitative measure of the closeness to text. If the score is larger than a certain threshold, the corresponding sub-image will be accepted as text. Based on these SVM scores, we get rid of the non-text parts in the right and left sides of input sub-image. Thus, the initial text boundary shrinks to the right position. Fig. 3 is a good reference for understanding this method, and some related results are shown in Fig. 6 (a) ~ (d).

3. TEXT BOUNDARY COMBINATION

In general, the initial text localization results contain many text fractions, which should be combined together. To avoid accepting non-text image parts falsely, we propose to use the SVM output scores for the purpose of boundary combination. We choose two of initial localized text lines, which will be combined together when: 1) they are near or 2) not near each other, but the image part between them is accepted by the SVM. This method is illustrated in Fig. 4, where boxes a, b and c denote the three initial text boundary boxes before boundary combination. Box D denotes the image part between a and b, and the green boxes denote the results after boundary combination. If the SVM score of box D is high, boxes a and b are combined. Then, boxes a, b, and c are combined, because boxes b and c are near. However, if the score of D is low, boxes a and b can not be combined. Some related results are shown in Fig. 6 (e) ~ (j). In these results, the missed text regions are correctly recalled by the text boundary combination module

4. TEXT BOUNDARY EXTENSION

Sometimes, the initial text localization results are not complete due to complex illumination and background clutter. Therefore, some parts of the text can be missed.

In order to recall the missed text, we add a text boundary extension moudule based on the SVM output score and image similarity measurement, which are combined together to improve the performance. Our purpose is to determine whether the outer sub-image near the initial text boundary is accepted by the score and image similarity between the central part of the initially detected results and the outer sub-image. This method contains the following four steps. Fig. 5 gives a clear illustration of the text boundary extension module.

Step 1: Non-parametric PDF estimation

From the central part of each initial text line sub-image, we estimate the color PDF (Probability Density Function) p_1 in HS (Hue and Saturation) color space, using Parzen window based method [8]:

$$p_1(x) = \frac{1}{nh^d}\sum K(x - x_i / h), \quad (1)$$

where $K(\bullet)$ is Gaussian kernel function with scale h, x_i the HS color vector of the ith pixel in the detected text line sub-image and d the dimension of x (=2). Note that we choose HS color space because HSV is very similar to human color perception, and "V" is not used here because it represents the brightness of the color, which should be ignored in our case.

Step 2: Image similarity measurement

Measuring the similarity between images (or their PDFs) is an essential issue in low-level computer vision. Recently, Vasconelos et al. give a unifying view of image similarity [9]. Here, we use Bhattacharyya coefficient D_{Bhat}, which is nearly optimal since it is the upper bound of the Bayesian classification error. Comaniciu et al. demonstrate that D_{Bhat} has impressive performance in texture retrieval and non-rigid tracking [8].

Bhattacharyya coefficient D_{Bhat} could be regarded as the probability distance between PDFs p_1 and p_2, which is formulated as:

$$D_{Bhat}(p_1, p_2) = -\ln \int \sqrt{p_1(x) \cdot p_1(x)}\, dx. \quad (2)$$

Intuitively, the smaller D_{Bhat} is, the more similar the two sub-images or their PDFs are.

Step 3: SVM classification

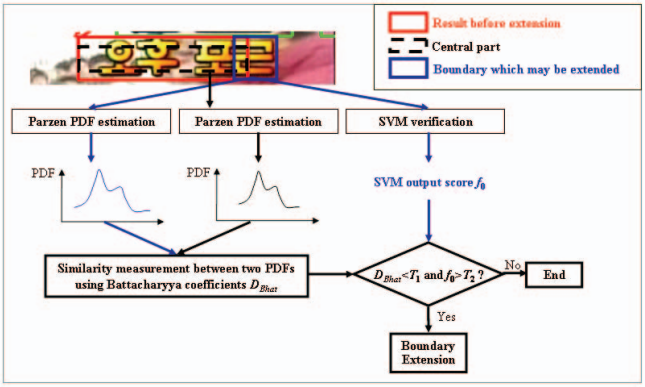

Fig. 5. Text boundary extension.

Compute the SVM output score f_0 of the boundary sub-image.

Step 4: Boundary extension

If $D_{Bhat} < T_1$ and $f_0 > T_2$, the initial text boundary is extended to include the boundary sub-image, where T_1 and T_2 are two threshold values. Then, we go to Step 1 to test the next boundary sub-image until no new text region is found. And we set the threshold values, T_1 and T_2, to 0.5 and 0.1, respectively, using the grid search method.

Fig. 6(f) ~ (n) show some related results. The last character is falsely rejected by the SVM classifier before boundary extension in Fig. 6 (m), because of its low contrast. However, it has similar appearance in HS space with the other characters. Finally, the last character is correctly recalled after the boundary extension.

5. EXPERIMENTS

We performed the experiments using a PC with one CPU (Intel Pentium 4 1.79GHz) and 256M RAM with Windows 2000 using Visual C++6.0. In our ground truth database, there are news images from South Korea TV channels, KBS, MBC and SBS, whose size are 720*480 (text height is from 10 to 150 pixels). In these images, 435 images are for testing and 357 images for training. In this paper, the initial text localization results are obtained by our previous work [6].

Ground truth is marked semi-automatically by a tool named MDTDS (Making Database for Text Detection and Segmentation), implemented by ourselves from the test data sets. There are totally 15437 text lines in the three test sets for testing. The size of our test set is 200 times larger than that of [13]. The recall rate, precision rate and time cost for a performance evaluation are defined as:

(1) Recall Rate = (Number of recalled GT characters) / (Number of characters in GT);

(2) Precision Rate = (Number of validate detected characters) / (Number of detected text characters);

(3) Time Cost(s/frame) = (Total processing time) / (Number of frames)

(4) Number of Characters = Text line length / (Text height * 1.0125).

As listed in Table. 1, 96.64% recall rate and 96.89% precision rate are reported on our test sets. And processing time is 1.08s/frame. The benefit of the text line refinement is to increase the accuracy of text localization in two aspects:

(1) The recall rate increases about 2.4% and precision rate about 0.5%, as show in Table 1. At the same time, the computational cost only increases a little.

(2) Sometimes, the accuracy of text localization can not be objectively evaluated by the recall rate and precision rate, since the two criteria are intrinsically based on overlapping area estimation. For example, for the first sub-image in Fig. 6 (e), the initial text localization boxes almost cover the whole text region, and thus the result of the performance evaluation will be that the text is recalled correctly. However, this localization result will seriously lead to wrong text recognition. Fortunately, after the text line refinement procedure, this problem is solved well, as shown in the second sub-image in Fig. 6 (k).

Table 1. Improvement by the text line refinement module.

	Recall Rate	Precision Rate	Time Cost (s/frame)
Without text refinement	0.9421	0.9639	0.92
With text Refinement	0.9664	0.9689	1.08

More results of text line refinement are shown in Fig. 6, from which we can find the proposed method works well in various complex situations. Of course, our method is not always satisfying, as shown in Fig. 6 (o) ~ (q). In Fig. 6 (o), the boundary extension stops too early, and in Fig. 6 (p) and (q), the text line refinement results are not better than before. Fortunately, this will not result in wrong text recognition.

Fig. 6. Results of the text line refinement. For the pair of sub-images in each sub-figure, the white dashed boxes on the first (second) sub-image indicate text localization result before (after) the text line refinement.

6. CONCLUSIONS

In this paper, we propose the text line refinement method for accurate text localization in images and videos. The proposed method contains text boundary shrinking, combination and extension, based on the SVM output score and image similarity measurement. The text line refinement module enables 2.4% more of the texts that are missed to be recalled. Since the text localization results will be used as the input of OCR (optical character recognition), the proposed method will significantly improve the recognition results. Although the proposed method is designed mainly for localizing the superimposed text in an image, it can also be used for the accurate localization of general texts, including video text, scene text, and so on.

7. REFERENCES

[1] K. Jung, K.I. Kim, and A.K. Jain, "Text Information Extraction in Images and Video: A Survey," Pattern Recognition, vol. 37, pp. 977-997, 2004.

[2] M. Lyu, J. Song, and M. Cai, "A Comprehensive Method for Multilingual Video Text Detection, Localization, and Extraction," IEEE Trans. on Circuits and Systems for Video Technology, vol. 15, no. 2, 2005.

[3] X. Hua, X. Chen, L. Wenyin and H.J. Zhang, "Automatic Location of Text in Video Frames," Proc. ACM Multimedia 2001 Workshop: Multimedia Information Retrieval, Canada, 2001.

[4] Y. Zheng, H. Li, and D. Doermann, "Machine Printed Text and Handwriting Identification in Noisy Document Images," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 26, no. 3, pp. 337-353, 2004.

[5] V. Vapnick, "The Nature of Statistical Learning theory," New York, NY: Spinger-Verlag, 1995.

[6] C. Jung, Q. F. Liu, and J.K. Kim, "Accurate Text Localization Based On the SVM Output Score," Submitted to Image and Vision Computing, 2007.

[7] D. Chen, O. Jean-Marc, and B. Herve, "Text Detection and Recognition in Images and Video Frames," Pattern Recognition, pp. 595-608, 2004.

[8] D. Comaniciu, V. Ramesh, and P. Meer, "Real-Time Tracking of Non-Rigid Objects using Mean Shift," IEEE Conf. Computer Vision and Pattern Recognition, vol. 2, pp. 142-149, South Carolina, 2000.

[9] N. Vasconcelos and A. Lippman, "A Unifying View of Image Similarity," Int'l Conf. Pattern Recognition, pp. 1038-1041, Spain, 2000.

[10] K. Kim, K. Jung, and J.H. Kim, "Texture-Based Approach for Text Detection in Image Using Support Vector Machines and Continuously Adaptive Mean Shift Algorithm," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, no. 12, pp. 1631-529, 2003.

[11] H. Li, D. Doermann, and O. Kia, "Automatic Text Detection and Tracking in Digital Video," IEEE Trans. on Image Processing, vol. 9, no. 1, 2000.

[12] V. Wu, R. Manmatha, and E.M. Riseman, "TextFinder: An Automatic System To Detect And Recognize Text In Images," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 21, no. 11, pp. 1224-1229, 1997.

[13] Q. Ye, Q. Huang, W. Gao, and D. Zhao, "Fast and Robust Text Detection in Images and Video Frames," Image and Vision Computing, vol. 23, pp. 565-576, 2005.