2014 22nd International Conference on Pattern Recognition

Graphics and Scene Text Classification in Video

Jiamin Xu[1],Palaiahnakote Shivakumara[2], Tong Lu[1], Trung Quy Phan[3] and Chew Lim Tan[3]

[1] National Key Lab for Novel Software Technology, Nanjing University, Nanjing, China

[2] Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia

[3] School of Computing, National University of Singapore

superxjm@yeah.net, hudempsk@yahoo.com, lutong@nju.edu.cn, phanquyt@comp.nus.edu.sg and tancl@comp.nus.edu.sg

***Abstract*—Achieving good accuracy for text detection and recognition is a challenging and interesting problem in the field of video document analysis because of the presences of both graphics text that has good clarity and scene text that is unpredictable in video frames. Therefore, in this paper, we present a novel method for classifying graphics texts and scene texts by exploiting temporal information and finding the relationship between them in video. The method proposes an iterative procedure to identify Probable Graphics Text Candidates (PGTC) and Probable Scene Text Candidates (PSTC) in video based on the fact that graphics texts in general do not have large movements especially compared to scene texts which are usually embedded on background. In addition to PGTC and PSTC, the iterative process automatically identifies the number of video frames with the help of a converging criterion. The method further explores the symmetry between intra and inter character components to identify graphics text candidates and scene text candidates. Boundary growing method is employed to restore the complete text line. For each segmented text line, we finally introduce Eigen value analysis to classify graphics and scene text lines based on the distribution of respective Eigen values. Experimental results with the existing methods show that the proposed method is effective and useful to improve the accuracy of text detection and recognition.**

Keywords*—*Temporal frames, Error estimation, K-means clustering, Video text segmentation, Eigen value analysis, Graphics and scene text classification

I. Introduction

Text detection and recognition is a topic of great interest to many systems such as Google Street View and iTowns, which have generated a huge amount of daily life images and videos [1]. Further, text detection and recognition in videos is often used for video information indexing and retrieval, since text can provide a concise and direct description of the objects or stories presented in videos, especially in news or sport videos. It is helpful for people to get more information about the video [2]. Therefore, in content-based information retrieval, video text detection and recognition has attracted much attention of many researchers. At high level, text in digital video can be divided into two types, namely, graphics text and scene text. Graphics texts are artificially added to video frames to supplement visual or audio content. Scene texts, on the other hand, appear within natural scenes and are directly captured by a camera. Examples of scene texts include street signs, billboards, texts on truck, and the writing on shirts. Since graphics texts are purposefully added, they are often more structured and closely related to the subject than scene texts. In some domains such as sports and map navigation systems, however, scene texts can be used to uniquely identify objects. Though scene texts are difficult to detect and extract due to their virtually unlimited range of poses, sizes, shapes and colors, they are still important in the applications such as navigation, surveillance, video classification, or analysis of sporting events [3]. Due to the presence of both graphics and scene texts in the same video frame, developing a general method which can detect both the two types of texts with a good accuracy in terms of detection rate and recognition rate has become an urgent but challenging task. Besides, it is noted from the above definition of graphics text and scene text that these two types share different characteristics to represent text information.

Current video text detection approaches can be classified into two categories. One category is detecting text regions in individual frames independently. The other category is utilizing the temporality of the video sequences [2]. The first category can be further divided into three kinds: connected component based methods [4], which may have difficulties when texts are embedded in a complex background or potentially touch other scene objects, texture-analysis-based methods [5], which can be very sensitive to font sizes or styles, and accurate boundaries of text areas are hard to find, and gradient-edge based methods [6], which are sensitive to background and thus in general produce more false positives. Most related previous work has focused on the extraction of graphics texts but not both graphics and scene texts simultaneously. To address this issue, several methods have been proposed in the past years [6, 7]. Although the methods solve the problems such as multi-orientations and curved scene text detection, the accuracy of text detection and recognition is still not consistent when the dataset changes due to the unpredictable characteristics of scene texts in video. Therefore, this work aims to achieve a good accuracy of video text detection by first separating graphics and scene texts from video frames, and then respectively selecting appropriate processing strategies for the two types of video texts. This is inspired by the work proposed in [8] for the identification of both handwritten and machine printed texts in document images to improve the recognition rate. It is shown that the identification of handwritten and machine printed before recognition in general improves the recognition rate. In addition, it is also shown in [9] that by proposing two dynamic thresholds to classify the low contrast and the high contrast individual frames before applying text detection methods, the accuracy of text detection can be improved compared to the methods without classification.

Text detection and recognition by integrating temporal information has been proposed by many methods [10-14] in the literature based on the fact that caption texts in general stay at the same position for a few seconds for human reading. A few methods detect dynamic caption texts in video by proposing spatio-temporal information. However, most of the methods exploit temporal information for improving the contrast of text, and it is noted that their focus is on detecting either graphics texts or scene texts but not detecting both of them. Besides, there is no proper criterion proposed in the existing methods to adaptively select the number of video frames for either text detection or text recognition. Instead, the methods use a fixed number of frames or

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.806

all the 30 frames per second. It is true that the fixed number of frames may affect the performance of the method because the contrast, resolution etc. in heterogeneous videos may vary as stated in [10]. Bouaziz et al. [13]. proposed a similarity criterion to find text appearance based on frame differences. However, the similarity criterion requires a threshold to identify the sudden difference. Therefore, it may not work for different types of videos. In addition, the focus of the method is only on graphics text detection but not scene text detection. Similarly, Huang et al. [14] proposed a method for text detection by using temporal information, edge density and texture information. These features are sensitive to background. Hence, in this work, we explore temporal information together with the intra and inter symmetries of character components to identify the potential graphics and scene text candidates. Boundary growing is used to restore the complete text line of each potential text candidate. Eigen value analysis over text lines is finally explored to classify graphics texts and scene texts in video frames. This idea works based on the fact that graphics texts in general do not have large movements compared to scene texts (background), in order to be readable.

II. The Proposed Method

For a video sequence, the method first performs an iterative process on successive frames. In this work, we consider video in which text appearance in the first frame since the input for this work is video containing text. We hypothesize graphics texts stay almost at the same location for a few seconds in several frames and do not have large movements compared to the background containing scene texts. Therefore, we can expect a low deviation for graphics text pixels and a high deviation for the background which potentially contains scene texts. Since it is a two-class problem, we apply k-means clustering with k=2 to classify the low deviation cluster and the high deviation cluster. The cluster which gives a low deviation is considered as the Probable Graphics Text Candidates (PGTC) cluster and the other is considered as the Probable Scene Text Candidates (PSTC) cluster. The condition based on the number of edges components in the PSTC cluster is defined to stop the iterative process as a converging criterion (this will be discussed in sub-section). For the edge components in the two clusters, we estimate stroke width [15] to extract the intra and inter symmetries of character components as stated in [16], which proposes such symmetry features for scene text detection in natural scene images, for distinguishing potential graphics texts and scene texts. Next, the boundary growing method as described in [6] is proposed to restore the missing text information from the potential text candidates by referring the Sobel edge image of the input frame. This results in text line segmentation. For each segmented text region, we introduce Eigen value analysis to study the distribution of Eigen values for classifying graphics and scene text lines. As motivated from the work presented in [17] for text detection using Eigen value analysis in which it is observed that Eigen values help in sharpening the edges of graphics texts, we consider scene texts can also be enhanced due to their background variations and non-uniform colors. Since graphics text has uniform color values with the plain background, the number of the pixels with either low or high Eigen values is less compared to other Eigen values. On the other hand, scene texts can have any background with color due to their unpredictable background variations.

The logical flow of the method can be seen in Fig. 1, where PTC denotes potential text candidates of both PGTC and PSTC clusters.

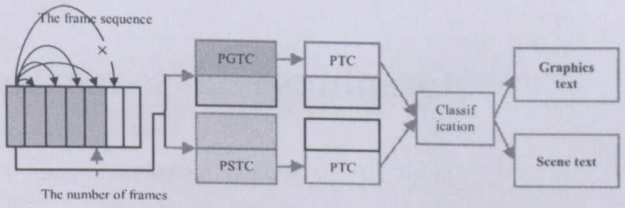


Fig. 1. Block diagram of the proposed classification method

A. Iterative Process to Identify Probable Text Candidates

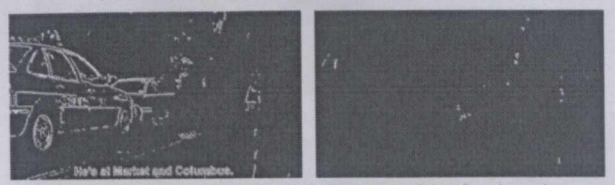
Let t, t+1, t+2…tn be the video sequence as shown in Fig. 2(a). Here n is rate of frames per second. The method initially considers the first two consecutive frames, say t and t+1 as shown in Fig. 2(b) for iteration. Each frame is divided into equally sized grids of 8×8 pixels. The Euclidean distance for each grid is computed using the intensity values of both the two frames. Then the distances are compared to find the deviations between the two frames. Next, we propose to use k-means clustering algorithm with k=2 to obtain PGTC cluster-1 and PSTC cluster-1 as shown in Fig. 2(c), where we can see that a few non-graphics text components are classified into PSTC cluster-1, and PGTC cluster-1 contains both graphics and non-graphics text components. We thereby count the numbers of the edge components in PGTC cluster-1 and PSTC clusters-1 for this iteration, say TC and NC, respectively. Next, for the second iteration, we restore the gray patches which correspond to the edges in PGTC cluster-1 (the left one in Fig. 2(c)) from the input frame as shown in Fig. 2(d) and it is considered for deviation estimation with the t+2 frame, namely, the deviation is estimated between only the gray patches in PGTC cluster-1 and the corresponding gray patches in the t+2 frame. We can see from Fig. 2(d) that the black patches in PGTC cluster-1 have been successively removed as non-graphics text components. In the second iteration, PGTC cluster-1 is considered as the input of k-means clustering, the two results of which are respectively subtracted by PGTC cluster-1 as PGTC cluster-2 and added into PSTC cluster-1 as PSTC cluster-2. The results of k-means clustering after the second iteration can be seen in Fig. 2(e). In the same way, the number of non-graphics text components is decreasing in PGTC cluster-2 compared to PGTC cluster-1. As a result, the counts TC and NC in the PGTC cluster and the PSTC cluster decreases and increases, respectively.

This iterative process continues until a condition is reached such that that the number of component (NC) in the PSTC cluster decreases, or stays constant as the number of iterations increases. This is valid because as iteration increases, in the PGTC cluster, the number of non-graphic text components that will be classified into the PSTC cluster correspondingly increases. Namely, only graphics text components are finally left in the PGTC cluster after iterations. At the some point, after classifying most non-graphics text components from the PGTC cluster into the PSTC cluster, the count of classifying new non-graphics text components in the next iteration either decreases or remains constant as there are no more non-graphics text components existing in the PGTC cluster. This can be seen in Fig. 2(f) where the final PGTC cluster contains only graphics text information and the final PSTC cluster contains few non-graphics text components after 18 iterations. To extract this observation, we draw the graph on the number of the non-graphics text components in the PSTC cluster for each iteration as shown in Fig. 3. One can see from Fig. 3 that the number of non-graphics text components in PSTC remains constant after 18 iterations. Therefore, the iterative number at which the number of edge components in PSTC remains constant

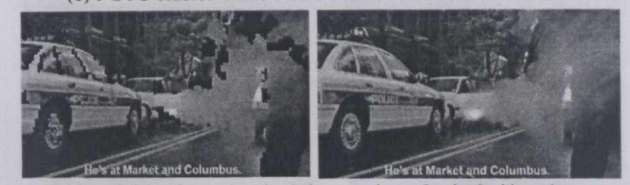
is considered as the converging criterion to stop the iterative process. If graphics text disappears after few frames, iterative process terminates quickly with two cluster results. Thus, we can conclude that this iterative process respectively gives graphics candidates in the final PGTC cluster and scene text candidates by the final PSTC cluster after iterations. In addition, this converging criterion overcomes the problem of choosing a fixed number of video frames for processing further as the existing methods [10-14]. This is actually the advantage and the contribution of this step.

(a) Input : Video sequence starting from t, t+1, t+2 to tn

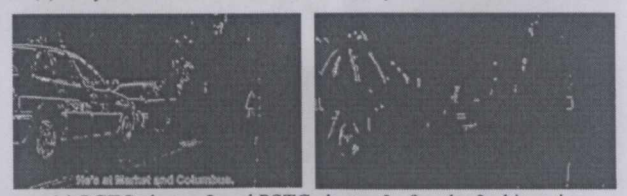
(b) t and t+1 frames as input for the first iteration



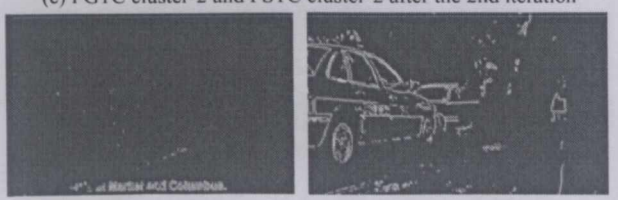
(c) PGTC cluster-1 and PSTC cluster-1 of the first iteration



(d) Gray PGTC cluster-1 and t+2 frame as input for the 2nd iteration



(e) PGTC cluster-2 and PSTC cluster-2 after the 2nd iteration



(f) Finally, PGTC cluster-18 and PSTC cluster-18 at the 18th iteration

Fig. 2. Probable candidates of graphics and scene texts. Note that we shade the background of the cluster results to make visible

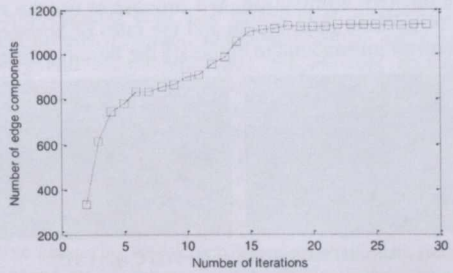


Fig. 3. The number of iterations vs the number of edge components in the PSTC cluster

B. Text Line Segmentation based on Symmetry and Boundary Growing

Fig. 2 shows that the iterative process outputs graphics text candidates by the final PGTC and scene text candidates by the final PSTC after each iteration. To further identify potential text candidates from the probable text candidates, we propose to explore the stroke width distance as presented in [15] by considering the shape characteristics of text characters. The method in [15] moves along the gradient direction to calculate stoke width and finds texts using the assumption that stroke width distances remain constant throughout a character, while non-text components may not have constant stroke width distances. However, the gradient direction may not always be inside a character and can also be outside the character due to being affected by the gray intensity of text and background. It is true that there exists symmetry both inside-outside a stroke and inter-intra characters [16]. With this observation, we propose a new symmetry feature which considers the gradient direction and the inverse gradient direction in both the graphics text candidates and the scene text candidates. We consider in this way potential text candidates of graphics and scene texts can be well outputted. The new symmetry is defined as follows.

For each edge pixel p, we move along its gradient direction as in [15] and inverse gradient direction until it meets other edge pixels, say q_1, q_2 respectively. We then define the symmetry of the center pixel p:

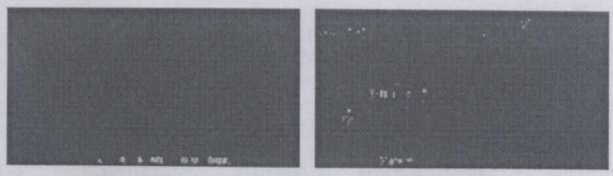
$$dist_{1,2} = \begin{cases} \|p - q_{1,2}\|, & if\ q_{1,2}\ exists \\ Inf, & otherwise \end{cases} \quad (1)$$

$$symmetry = \begin{cases} 1, & if\ \|dist_1 - dist_2\| < K \\ 0, & otherwise \end{cases} \quad (2)$$

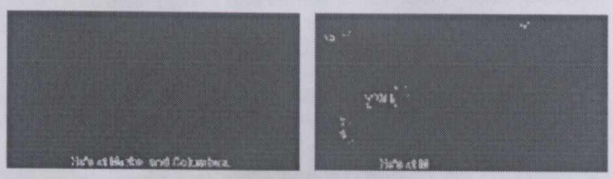
K is determined empirically to be 5.

Equation (1) calculates the distance between the center pixel and the symmetry pixels, while equation (2) produces a symmetry map containing the pixels from potential text candidates, which satisfy equation (2) as shown in Fig. 4(a), where it is found that most of the non-text pixels are eliminated from both PGTC and PSTC. However, the symmetry alone may not remove all the non-graphics text pixels as we can see still non-graphics text components in Fig. 4(a). We can also notice from Fig. 4(a) that both the potential text candidates of PGTC and PSTC do not provide complete text information. Therefore, we restore the edge components corresponding to potential text candidates from the Sobel edge image of the first input frame as shown in Fig. 4(b). The boundary growing is used as stated in [6] to segment text lines by referring again the Sobel edge image as shown in Fig. 4(c), where it can be seen that the graphics and the scene texts are well segmented. Fig 4(c) shows the output

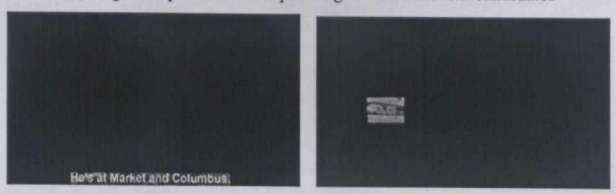
after false positive elimination. We propose to use the standard deviation of gradient information and the ratio of horizontal and vertical gradient information to eliminate the false positives.



(a) Potential text candidates for PGTC and PSTC



(b) Edge components corresponding to Potential text candidates



(c)Outputs of boundary growing and false positive elimination for potential text candidates of both graphics and scene text

Fig. 4.The text line segmentation: Note that we shade the background of cluster results to make visible

C. Graphic and Scene Texts Classification

The previous section gives text lines of both graphics and scene text in video. Since the method presented in Section B works based on unsupervised k-means clustering, it is not sure whether the method always classifies graphics text pixel into one cluster and scene text pixel into another cluster because of text movements and distortion effects. Therefore, we propose a method to classify each segmented text line as either a graphics text or a scene text correctly. Hence, in this section, we are inspired by the work presented in [17] for text detection in video images, where it is stated that Eigen values are useful for differentiating text and non-text pixels. It is noted that usually Eigen value represents the horizontal and vertical variance in gray intensity level for each grid. In the same way, the global contrast and clarity for text can also be measured by the Eigen value distribution for classification. We propose Eigen values analysis for the gray text lines that are classified by Section B. It is found that low and high Eigen values do not contribute much for graphics texts, while low Eigen values contributes much for scene texts. This is valid because graphics texts generally have plain background and uniform texts. On the other hand, scene texts have unconstrained background and non-uniform colors. For instance, we can notice from Fig. 5(a) that Eigen value enhances text pixels and suppresses non-text pixels for the segmented graphics text line. In addition, we can also see more Eigen values contribute to sharpening texts. On the other hand, we can see the Eigen image for the segmented scene text image shown in Fig. 5(b), where Eigen value enhances both the texts, as well as non-text due to unconstrained background. Thus, we get bell curves for graphics text lines and non-bell curves for scene text lines as shown in Fig. 5(c). The Eigen value estimation is done more formally as follows.

First, we calculate the dominant stroke width distance for the text block that is most frequency stroke width distance obtained from the stroke width histogram. The stroke widths here are obtained along gradient direction and inverse gradient direction as discussed in Section B. Let the dominant stroke width be w. Second, slide a $w * w$ window W_{ij} in the text blocks with step size equals one pixel. Third, let M_{ij} be the gray intensity matrix of W_{ij}, μ_{ij} be the mean value for M_{ij}, calculate the Eigen values for $(M_{ij}^T - \mu_{ij}) * (M_{ij} - \mu_{ij}) + (M_{ij} - \mu_{ij}) * (M_{ij}^T - \mu_{ij})$ and let λ_{ij} be the maximum Eigen value for window W_{ij}. For each $row * col$ text block, we get a set of Eigen values $E = \{\lambda_{ij} | i \in 1 \ldots row + 1 - w, j \in 1 \ldots col + 1 - w\}$. After sorting E from small to large, normalizing to [0,1] and eliminating the Eigen values near zero, we plot a histogram H for these Eigen values $E$with 20 bins as the blue line graph shown in Fig. 5(c).

For each text block, H is calculated and fit with single Gaussian curve (3) using MATLAB cftool toolbox.

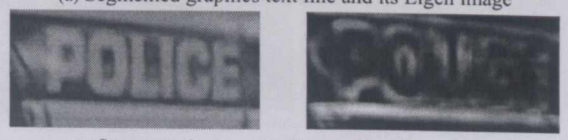
Gaussian curve formular:

$$f(x) = \frac{1}{k} exp\left\{-\frac{(x-\mu)^2}{m}\right\} \tag{3}$$

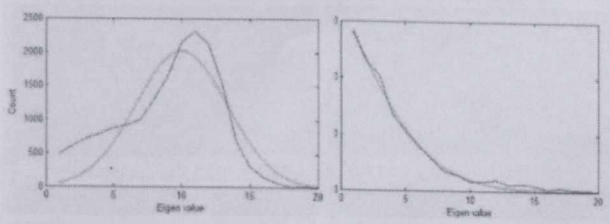
μ is the expectation, k and m are the parameters.

After fitting the histogram curve for H with the Gaussian curve as shown in Fig. 5(c), the expectation μ is calculated. If the fitting curve exists and μ is in the first $1/5$ of the x axis, less than 4 (can be negative) in this situation, the text block is considered as a scene text block. Otherwise, it is a graphics text block.

(a) Segmented graphics text line and its Eigen image



(b) Segmented scene text line and its Eigen image



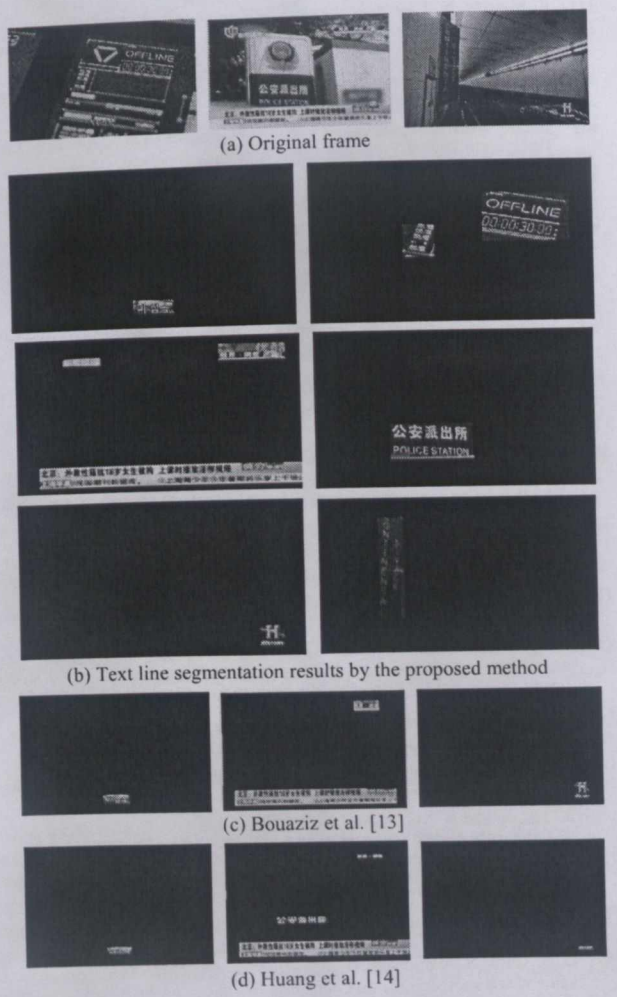
(c) The curves for graphics texts and for scene texts (blue for the histogram curve, red for the fitted Gaussian curve)

Fig. 5. Eigen value analysis for studying distribution of text pixels of both graphics and scene text

III. Experimental Results

Since there is no benchmark dataset for classification of graphics and scene text using temporal frames, we collect video data comprising 500 video clips containing text, which includes 693 graphics and 586 scene texts. Each video clip may last less than 2 seconds. These videos contain both graphics and scene text of different scripts, orientation, contrast, resolution etc. The text detection method is evaluated in terms of Recall (R), Precision (P)

and F-measure (F) and Average Processing Time (APT). In order to show that the proposed method is effective, the method is compared with the existing algorithms [13] and [14]. The reasons to choose these two existing algorithms are as follows. First, they use temporal information for text detection. Second, their goal is to detect text regions but not exact text lines fixed by closed bounding boxes. However, the algorithms explore edge density, similarity measures and stroke information to detect texts, while the proposed method presents an iterative procedure to select the number of frames and explores deviation estimation between consecutive frames to classify text regions without any threshold. We present the confusion matrix for classification of graphics and scene text in Section III.B and the recognition rate to show that how classification is effective in Section III. C.



(a) Original frame

(b) Text line segmentation results by the proposed method

(c) Bouaziz et al. [13]

(d) Huang et al. [14]

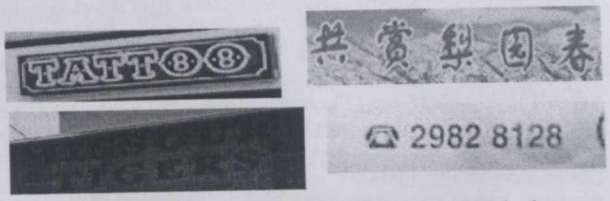
Fig. 6. Sample results of the proposed and the existing methods for text segmentation

A. *Experiments for Text Line Segmentation*

Sample results of the proposed and the existing methods are shown in Fig. 6, where (a) shows the first input frame containing both graphic and scene texts, (b) shows the text line segmentation results by the proposed method, (c) and (d) respectively show the results of Bouaziz et al. and Huang et al. methods. Fig. 6 shows that the proposed method segments graphics and scene text lines properly and separately, while the existing methods segment graphics texts well but fail to segment scene texts. Fig. 6 also shows that the proposed method is good for segmenting different scripts. The quantitative results of the proposed and the existing methods are reported in Table I, where the proposed method gives high recall, precision and F-measure compared to the existing methods. However, the proposed method requires more time for text detection compared to the existing methods due to the iterative clustering and Eigen value analysis. The main reason for poor accuracy of the existing methods is that the methods use threshold for finding the similarity between consecutive frames based on edge density and binarization as these two are good for graphics text but not for scene text which may not preserve similarity and edge density. In addition, the methods may not work well when the frame contains both graphics texts and scene texts because of thresholds and the variations of texts. On the other hand, the proposed method separates graphics and scene text and then segment text lines and hence gives a better accuracy.

TABLE I. PERFORMANCE OF THE PROPOSED AND EXISTING METHODS FOR TEXT DETECTION (IN %)

Method	*R*	*P*	*F*	*APT*
Proposed Method	**91**	**82**	**86**	6 second
Bouaziz et al. [12]	45	56	49	4 second
Huang et al. [16]	67	60	63	1 second



(a). Sample graphics text classified by the proposed method

(b) Sample scene text classified by the proposed method

Fig. 7. Sample graphics and scene text blocks from our database

B. *Experiments for Classification of Graphics and Scene Texts*

We choose 500 graphics text blocks and 400 scene text blocks segmented by the method as the samples respectively shown in Fig. 7 (a) and (b) to evaluate the classification method in terms of the classification rate. It is observed from Fig. 7that the proposed method classifies both graphics texts and scene texts correctly, even text blocks are suffering from illumination, orientation, different fonts, different contrast and different scripts. The quantitative results of the classification method are reported in Table II, where one can see the classification rate for scene texts is somewhat higher than that of graphics texts because graphics texts must satisfy a bell distribution, while scene texts do not.

Therefore, some of the graphics texts are classified as scene texts. Thus, we can conclude that the proposed method is good for classifying graphics texts and scene texts in video.

TABLE II. CONFUSION MATRIX FOR CLASSIFICATION OF GRAPHICS AND SCENE TEXT (IN%)

Type	*Graphics text*	*Scene text*
Graphics text	**80.2**	19.8
Scene text	17.5	**82.5**

C. Validating classification by for Recognition

To show that the proposed method is effective and useful, we evaluate classification in terms of recognition rate for the classified graphics and scene texts before classification and after classification. We implement two baseline thresholding binarization methods [18, 19] and the video text binarization method [20] to binarize the segmented text lines. Then we use Tesseract (Google) OCR, which is freely available to calculate character recognition rate. We calculate the recognition rates before classification which accepts both graphics and scene text as input and after classification which accepts graphics and scene text separately. The recognition results before and after classification of the binarization methods are reported in Table III. Table III shows that all the three methods give good recognition rates for graphics texts and before classification compared to scene texts after classification. The method in [20] gives better results than the other two methods because it is developed for video text binarization, while the other two were developed for scanned document images. It is clearly noticed from the recognition rates of graphics texts in Table III that the presence of scene texts in video causes the main reason to get a poor accuracy for text recognition. Therefore, we can assert that the classification of graphics and scene texts is essential and it improves recognition rate.

TABLE III. CHARACTER RECOGNITION OF THE BINARIZATION METHODS BEFORE AND AFTER CLASSIFICATION (IN%)

Methods	*Before classification*	*After classification*	
	Graphics+Scene	*Graphics*	*Scene*
WGF[20]	67.6	82.3	49.6
Niblack[18]	52.4	71.5	29.1
Souvola[19]	27.3	38.0	14.4

IV. Conclusion and Future Work

This paper presents a novel method for classifying graphics and scene text in video. The method explores temporal information in an iterative way to find probable graphics and scene text candidates. The iterative process also helps in identifying the exact number of frames automatically by satisfying a converging criterion. The potential graphics and scene text candidates are identified with the help of stroke width symmetry of character components. The boundary growing method is used to segment text lines. We further introduce Eigen value analysis to study the graphics text pixel distribution and scene text pixel distribution in order to discriminate graphic and scene texts. Experimental results with the existing methods show that the proposed method is effective and useful for both text detection and recognition. To the best of our knowledge, this is the first work on classification of graphics texts and scene texts by using temporal information. We are planning to extend this method to solve the situation where both graphics and scene text are moving arbitrarily.

Acknowledgements

The work described in this paper was supported by the Natural Science Foundation of China under Grant No. 61272218 and No. 61321491, the 973 Program of China under Grant No. 2010CB327903, and the Program for New Century Excellent Talents under NCET-11-0232.

References

[1] R. Minetto, N. Thome, M. Cord, N. J. Leite and J. Stolf, "Snoopertrack: Text Detection and Tracking for Outdoor Videos", In Proc. ICIP, pp 505-508, 2011.

[2] R. Wang, W. Jin and L. Wu, "A Novel Video Caption Detection Approach using Multi-Frame Integration", In Proc. ICPR, 2004.

[3] H. Li, D. Doermann and O. Kia, "Automatic Text Detection and Tracking in Digital Video", IEEE Trans. IP, pp 147-156, 2000.

[4] A. K. Jain and B. Yu, "Automatic Text Location in Images and Video Frames", PR, pp. 2055-2076, 1998.

[5] 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 Trans. on PAMI, pp. 1631-1639, 2003.

[6] P. Shivakumara, H. T. Basavaraj, D. S. Guru and C. L. Tan, "Detection of Curved Text in Video: Quad Tree based Method", In Proc. ICDAR, pp 594-598, 2013.

[7] Y. Liu, Y. Song, Y. Zhang and Q. Meng, "A Novel Multi-Oriented Chinese Text Extraction Approach from Videos", In Proc. ICDAR, pp 1387-1391, 2013.

[8] Y. Zheng, H. Li and D. Doermann, "Machine Printed Text and Handwriting Identification in Noisy Document Images", IEEE Trans. PAMI, pp 337-353, 2004.

[9] P. Shivakumara, W. Huang, T. Q. Phan and C. L. Tan, Accurate Video Text Detection Through Classification of Low and High Contrast Images, PR, pp 2165-2185, 2010.

[10] D. Chen and J. M. Odobez, "Video text recognition using sequential Monte Carlo and error voting methods", PRL, pp 1386-1403, 2005.

[11] T. Q. Phan, P. Shivakumara, T. Lu and C. L. Tan, "Recognition of Video Through Temporal Integration", In Proc. ICDAR, pp 589-593, 2013.

[12] B. Bouaziz, W. Mahdi and A. B. Hamadou, "Automatic text regions location in video frames", In Proc. SITIS, pp 2-9, 2005.

[13] B. Bouaziz, T. Zlitni and W. Mahdi, "AViTExt: Automatic Video Text Extraction" CoRR abs/1301.2173, 2013.

[14] X. Huang and H. Ma: Automatic Detection and Localization of Natural Scene Text in Video. ICPR 2010:3216-3219.

[15] B. Epshtein, E. Ofek, Y. Wexler, "Detecting text in natural scenes with stroke width transform," In: Proc. CVPR, 2010, pp. 2963-2970.

[16] T. Q. Phan, P. Shivakumara and C. L. Tan, "Detecting Text in the Real World", In Proc. ACM Multimeida (ACMMM), 2012, pp 765-768.

[17] D. S. Guru, S. Manjunath, P. Shivakumara, and C. L. Tan, "An eigen value based approach for text detection in video," In: Proc. DAS, 2010, pp. 501-506.

[18] W. Niblack, "An Introduction to Digital Image Processing", Prentice Hall, Englewood Cliffs, 1986.

[19] J. Sauvola, T. Seeppanen, S. Haapakoski and M. Pietikainen,"Adaptive Document Binarization", In Proc. ICDAR, 1997, pp 147-152.

[20] S. Roy, P. Shivakumara, P. Roy and C. L. Tan, "Wavelet-Gradient-Fusion for Video Text Binarization", In Proc. ICPR, 2012, pp 3300-3303.