Multimed Tools Appl
DOI 10.1007/s11042-015-2690-6

An automatic video text detection method based on BP-adaboost

Hui Wu [1,2] · **Bei-ji Zou** [1,2] · **Yu-qian Zhao** [1,2,3] · **Hong-pu Fu** [1,2]

Received: 21 October 2014 / Revised: 20 March 2015 / Accepted: 12 May 2015

Abstract Video text usually provides us a lot of useful information that is important for video analysis, indexing and retrieval. However, it is still a challenging work to detect text from video images due to variation of text patterns and complexity of background. In this paper, an automatic video text detection method is proposed. Firstly, K-means is utilized to classify pixels in gradient images into text and non-text regions. Subsequently, morphological operations are performed on text regions to form connected candidate text components, followed by projection profile boundary refinement. Finally, the detection results are verified by geometry and BP-Adaboost identifications. The experimental results on our manually selected dataset and the publicly available Microsoft Asia dataset show the effectiveness and feasibility of the proposed method.

Keywords Video text detection · Morphological operations · Projection profile · BP-Adaboost

1 Introduction

Video is now becoming one of the most popular media via TV, Internet, micro-blogging, and many other ways. In order to manage massive video data and realize efficient video retrieval, indexing and summarization, a large number of content-based video analysis approaches have been proposed over the past years. The current video analysis methods are usually based on the information of audio, video, text and so on. Text information plays an important role in video analysis, since text is easy for us to recognize and can provide many significant clues such as

✉ Yu-qian Zhao
zyq@csu.edu.cn

[1] School of Information Science and Engineering, Central South University, Changsha 410083, China

[2] Mobile Health Ministry of Education, China Mobile Joint Laboratory, Changsha, Hunan 410012, China

[3] School of Geosciences and Info-Physics, Central South University, Changsha 410083, China

Published online: 14 August 2015

scene locations, speakers' names, program introductions, scores, time, etc. Generally, video text is categorized into two kinds, graphics and scene text. The former is artificially added to the video during the editing process and the latter is captured by the camera which exists in the image naturally [13]. Among these two kinds of text, graphics text seems to be more important for video content understanding. For instance, graphics text in news programs usually describes information of relative people and events; subtitles in sport programs often provide information of players or scores [19]. In order to extract text information in video images, lots of text detection methods have been proposed.

Generally speaking, the existing video text detection methods can be classified into four categories: edge based [1, 9, 11, 17, 19], texture based [16, 20], *Connected Components* (CCs) based [13, 14, 18] and temporal information based methods [7, 8, 12]. Edge based methods usually apply edge information for rough text localization, and then perform text verification procedures to obtain refined text regions. These methods are generally effective for simple background images, while for complex ones, some false alarms may generate and a set of heuristic constraints have to be applied for non-text elimination, making them lack of generality. Texture-based methods are based on the observation that text in video images is rich in texture information, which can be used to distinguish text from background. These methods usually use texture analysis approaches such as wavelet transform, Gabor filter, *Fast Fourier Transform* (FFT), *Discrete Cosine Transform* (DCT) etc. to extract texture features and employ a trained classifier to eliminate non-text. These methods can distinguish text from background when they have different texture features and fail when the background contains some non-text blocks with abundant texture features such as bricks, windows, handrails and so on. Since text in one frame usually has similar color and satisfies certain size, shape and spatial alignment constraints, CCs based methods firstly achieve small text components by edge detection or similar color clustering and then group them into successive larger ones according to their similar properties. A procedure of non-text components exclusion is usually needed in the follow-up stages. These methods can extract multi-oriented text but cannot detect text accurately without prior knowledge of text position and scale, and sometimes are ineffective on irregular text. Unlike the aforementioned three kinds of methods detecting video text at an individual image level, temporal information based methods detect text by applying features between neighboring video frames, which can help generate more text and exclude non-text at the same time.

Although many research efforts have been made to detect text from video images, this topic is far from being solved due to the variation of font sizes, colors, styles and the complexity of the video images. Thus more robust and effective methods are still expected. In this paper, we propose an automatic video text detection method based on BP-Adaboost, which focuses on detecting horizontal graphics text at image level and the main contributions include: 1) The combination of non-text elimination at pixel level and text verification by BP-Adaboost at block level makes our method receive high accuracy. 2) Accurate text block boundary is obtained by performing boundary refinement on corresponding region in Sobel map of each labeled block. 3) A video image dataset is built, containing text images from different programs in different languages, and the proposed method receives satisfactory results on different datasets.

The remainder of this paper is organized as follows. Section 2 reviews state-of-the-art video text detection methods. Section 3 describes the proposed automatic video text detection method. In Section 4, the experimental results and discussion are presented. The conclusions and future work are included in Section 5.

2 Related works

As mentioned above, it is an important and challenging work to detect text from video images, and many related works have been done, some of which will be reviewed in this section.

Text detection performance is usually influenced by environment such as low contrast, illumination change and so on. To decrease these influences, edge-based methods have been applied. Yang et al. [19] proposed an edge-based multi-scale detector for text localization and adopted image entropy-based filter together with classifier-based strategy for non-text elimination. This method takes both text detection and recognition into account, while various assumptions make its application limited. Liu et al. [9] detected initial text candidates by classifying pixels in four Sobel maps, followed by projection profile refinement, which works well for images with complex background, but fails to detect text in small sizes. Cai et al. [1] filtered out non-text pixels in the edge map and refined it by two text-area enhancement operators. Phan et al. [11] proposed a Laplace-based video text detection method, which locates text region by computing *Maximum Gradient Difference* (MGD) for each pixel in the Laplacian-filtered image. Both methods in [1] and [11] can detect text under low contrast. Wong et al. [17] computed the MGD to locate candidate text regions and used color information to remove non-text blocks. This method receives high precision rate while too many heuristic rules are adopted.

In addition to environment influences, complex background also increases text detection difficulty, and many texture based methods are proposed to decrease influences of background. Zhao et al. [20] proposed a classification-based algorithm to detect text using a sparse representation with discriminative dictionaries. The method uses wavelet transform to detect edges and divides them into patches. The classification procedure is performed on the patches to obtain candidate text regions, which are then refined by an adaptive run-length smoothing algorithm and projection profile analysis. This method can detect text in various fonts, sizes, colors etc. and works well for both graphics and natural scene text to some extent. Wei et al. [16] combined gradient and texture features to detect text in video images using pyramidal scheme. The method is performed on three images including the original and its two downsized images generated by bilinear interpolation, and text regions are obtained by classifying pixels in these three images according to their gradient values, which will be final verified by a trained classifier based on texture features. This method combines gradient and texture features for text detection and receives good text detection results.

Many existing video text detection methods mainly focus on detecting text in horizontal direction. To extract text in arbitrary orientations, some algorithms are proposed. Shivakumara et al. [13] proposed a multi-oriented video text detection method, which detects text by skeleton extraction on Fourier-Laplacian filtered image, and some false positives may generate due to simple text verification procedures. They also presented an algorithm to detect multi-oriented video text through Bayesian classification and boundary growing [14]. The algorithm uses the product of Laplace and Sobel to enhance text pixels and gets true ones by a Bayesian classifier. Multi-oriented video text is obtained by connecting small text components into larger ones based on boundary growing method. Wu et al. [18] presented a new thinning based multi-oriented video character extraction method, which firstly applies *Ring Radius Transform* (RRT) to generate a radius map, then detects arbitrary orientated text by identifying medial axes of strokes, and finally performs false positives identification by analyzing color histograms of medial axes. This method can detect arbitrary oriented text while sometimes sensitive to low contrast and illumination change.

In contrast to the aforementioned methods which detect text at an individual image level, some methods use features between frames for text detection. Li et al. [7] proposed an effective video text extraction method using *Key Text Points* (KTPs) acquired by wavelet transform on original images. The differences at the KTPs between neighboring frames are used to reduce influence of dramatic background variation. This method takes features between frames into account and works well for complex and low contrast images to some extent. For this method is mainly based on characters' KTPs, it therefore results in unsuccessful detection of small size text with less KTPs. Qian et al. [12] adopted the integer DCT coefficients of frames for graphics text detection and localization. This method considers text blocks in different sizes, sets adaptive thresholds and applies characteristics in video frames, making it effective in most situations. Liu et al. [8] proposed a stroke-like edge detection method to extract text and utilized inter-frame features together with spatial analysis to improve detection accuracy.

3 Methodology

In this paper, we propose an automatic video text detection method including three major steps: 1) locate text regions by K-means and morphological operations on the gradient images; 2) refine text candidates according to the projection profile on the *Sobel Map* (SM) of the original image; 3) eliminate the false positives by geometry and BP-Adaboost identifications. The flowchart of the proposed method is illustrated in Fig. 1

3.1 Text region localization

Image gradient reflects the change of intensity, and there is always big gradient difference between text regions and background. In this paper, we use gradient feature to separate text

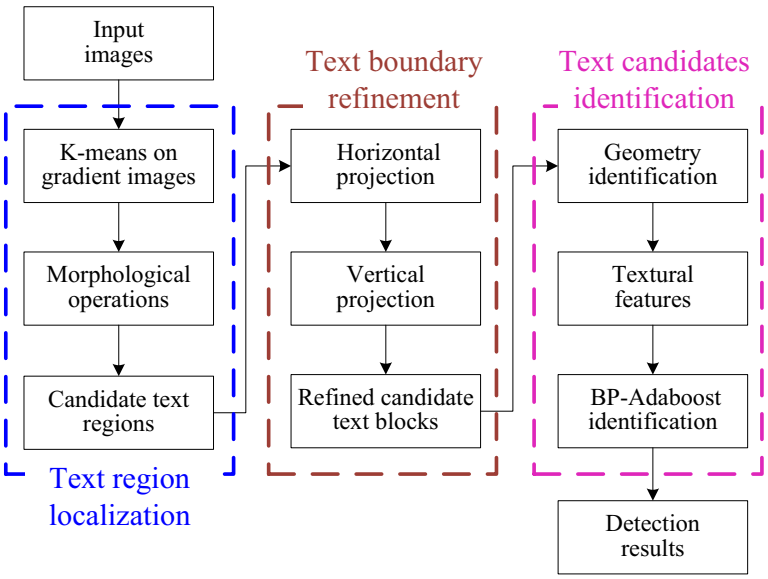


Fig. 1 Flowchart of the proposed method

regions from background. Firstly, we obtain the horizontal and vertical gradient maps of original image, which are denoted as f_h and f_v, respectively. Then at each pixel location, the MGD between the maximum and minimum gradient values within a local window of size $1 \times N$ is obtained based on f_h and f_v by

$$\mathrm{MGD}(i,j) = \max_{c\in\{h,v\}} (f_c(i,j+k)) - \min_{d\in\{h,v\}} (f_d(i,j+k)) \tag{1}$$

where $k \in \left[-\frac{N-1}{2}, \frac{N-1}{2}\right]$ and N is dependent on the maximum text size to be detected, which will be elaborated in Section 4.3.

Figure 2a shows an original grayscale image, the f_h, f_v and MGD of which are shown in Figs. 2b-d, respectively. It can be observed that the gradient images include many scattered pixels, among which the brighter ones correspond to the larger gradient values. In order to extract these brighter pixels for text localization and obtain binary gradient images, we adopt K-means clustering method for binarization which is simpler than setting thresholding values. Pixels in f_h are classified into two clusters C_1 and C_2 by K-means based on the Euclidean distance of their corresponding gradient values, and we define M_1 and M_2 as the average gradient values of clusters C_1 and C_2, respectively. Since the cluster order varies for different images and the text regions usually have larger gradient values than the non-text regions, we use the following rule to identify the text cluster:

$$Text\_Cluster = \begin{cases} C_1 & \text{if } M_1 > M_2, \\ C_2 & \text{otherwise} \quad . \end{cases} \tag{2}$$

In this way, we obtain a binary image of f_h, as shown in Fig. 2e. The same operation is performed on the MGD, and the result is shown in Fig. 2f. It can be observed from Figs. 2e and f that some non-text pixels are classified into the text regions because of their larger gradient values. In order to decrease the number of misclassified pixels in the text regions, we compute the intersection of Figs. 2e and f to form a final clustering result, which is called *Candidate Text Pixel Map* (CTPM), as shown in Fig. 2g. Thus, some non-text information is eliminated at pixel level.

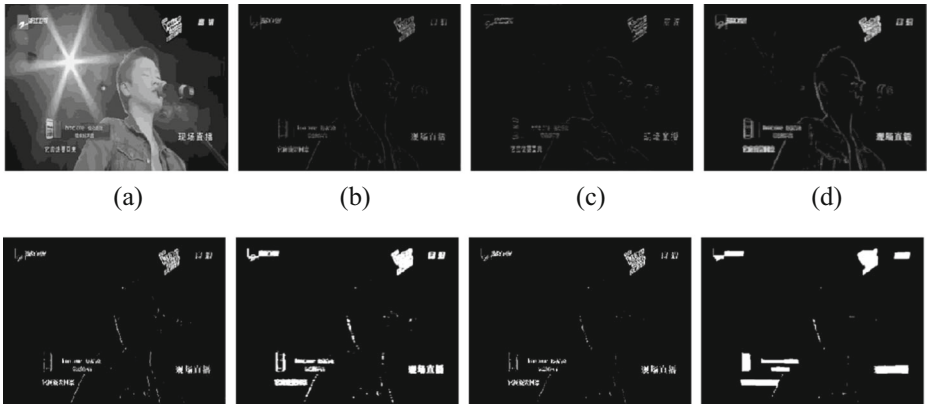


Fig. 2 Text localization. (**a**) Grayscale image, (**b**) f_h, (**c**) f_v, (**d**) *Maximum Gradient Difference* (MGD) (**e**) binary image of f_h and (**f**) MGD, (**g**) *Candidate Text Pixel Map* (CTPM), and (**h**) *Candidate Text Regions Map* (CTRM)

From Fig. 2g, we can see that the CTPM is composed of the scattered pixels. In order to generate text regions from these scattered pixels, a series of morphological operations is applied on the CTPM. First, horizontal morphological operations 'close' with a horizontal line structuring element of length 30 and 'open' with that of length 2 are applied to obtain a horizontal text region image I_h. In the same way, a vertical text region image I_v is obtained. The union of I_h and I_v is denoted by I_s and calculated as following:

$$I_s(i,\ j) = \begin{cases} 1, & \text{if } I_h(i,j) = 1 \text{ or } I_v(i,j) = 1 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Then the morphological 'close' operation is performed on I_s, with a square structuring element of size 6×6, and a *Candidate Text Regions Map* (CTRM) composed of several text blocks is obtained, as shown in Fig. 2h.

Figure 3 gives two other examples of the text region localization according to the aforementioned operations. It can easily observe that some boundaries of text blocks are incorrect. In order to achieve the exact text boundaries, we introduce text boundary refinement in next section.

3.2 Text boundary refinement

We use 4-adjacency type [15] to label each candidate block in CTRM and refine it on the SM of the original image. A projection profile is applied on the corresponding region in SM (Fig. 4a) to each labeled block. In the horizontal direction, we calculate the total pixel number of the rth row in the corresponding SM of the i^{th} labeled block, denoted by $N_i(r)$. If $N_i(r)$ is smaller than a threshold T_1, the row r in the i^{th} labeled block will be set as a gap, otherwise set as a text line. Considering that different images have different Sobel information, we normalize $N_i(r)$ to [0, 1], so that we can set a fixed T_1 for most of the detected images. The horizontal

Fig. 3 Text region localization. (**a**) Original color images, (**b**) *Candidate Text Regions Maps* (CTRMs)

projection profile of Fig. 4a is shown in Fig. 4b, in which the horizontal and vertical axes indicate the normalized pixel number N_i and the corresponding row r, respectively. In the same way, we apply vertical projection profile, and the result is shown in Fig. 4c. We perform the aforementioned projection profile procedures on each labeled text block in CTRM, respectively, and obtain the final text boundary result, as shown in Fig. 4d, from which we can see that after the boundary refinement operations, the text blocks can get exact boundaries.

We use CTRM for text boundary refinement instead of using CTPM directly mainly because text in CTRM is refined on each block alone and therefore is not influenced with each other, however, text in CTPM is refined on the whole image and the boundaries of text blocks in it may influence mutually. In order to further illustrate the advantages of using CTRM over CTPM, we also perform projection profile operations on CTPM, and the result of horizontal projection profile on the corresponding region in SM (Fig. 4f) to CTPM (Fig. 4e) is shown in Fig. 4g. From Figs. 4f and g we can see when there exist multiple text blocks in the same rows, text with smaller height will be influenced by that with larger one, just as the text between the two red dashed lines. Figure 4h shows the final results of text boundary refinement on CTPM. Comparing text boundary refinement results on CTRM (Fig. 4d) and CTPM (Fig. 4h), we can see that the former shows more accurate text boundary results than the latter, which can be observed from the top detected text blocks.

Figure 5 gives the text boundary refinement results of the original color images in Fig. 3, which indicates that the candidate text blocks achieve exact boundaries. From Figs. 4d and 5, it can also be clearly found that the detected candidate blocks contain both the text blocks and the non-text blocks.

3.3 Text candidate identification

In order to obtain better video text detection results, we use geometry and BP-Adaboost identifications to remove the non-text blocks.

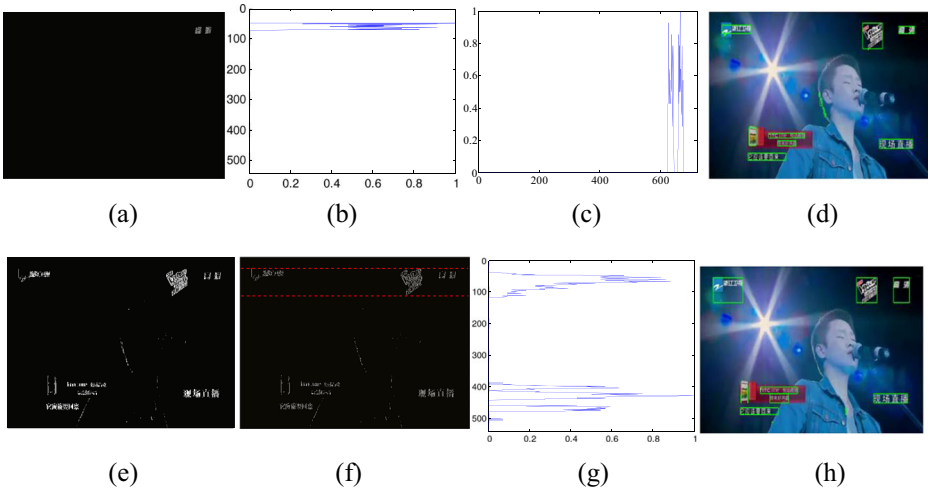


Fig. 4 Text boundary refinement. (**a**) Corresponding region in SM to the labeled block, (**b**) horizontal projection profile, (**c**) vertical projection profile, (**d**) text boundary refinement of CTRM, (**e**) *Candidate Text Pixel Map* (CTPM), (f) SM corresponding to CTPM, (**g**) horizontal projection profile, and (**h**) text boundary refinement of CTPM

Fig. 5 Text boundary refinement results of Fig. 3

3.3.1 Geometry identification

If a text block satisfies one of the following conditions, it will be regarded as a false positive:

$$W < T_2 \tag{4}$$

$$H < T_3 \tag{5}$$

$$R_a < T_4 \tag{6}$$

where W and H denote the width and height of each candidate block, respectively. R_a is the ratio of W to H. Figure 6 shows the geometry identification results of Figs. 4d and 5. Obviously, some non-text blocks in them are removed.

3.3.2 BP-Adaboost identification

Though geometry identification can eliminate some distinct non-text blocks, the results are still not satisfying. For example, there still exist some non-text blocks in Fig. 6. Therefore, we add classifier identification to decrease the number of the non-text blocks, and select BP-Adaboost as the classifier.

BP-Adaboost is a classifier composed of several BP networks whose weights are adjusted each iteration according to the error rate. The training process of BP-Adaboost classifier is shown in Table 1, where T is the maximum number of iterations, and is set to 12 in this paper.

To train the BP-Adaboost classifier and obtain a text/non-text model, we construct a training set which will be elaborated in Section 4.2. For each image in the training set, we perform the aforementioned text localization and boundary refinement procedures and obtain a

Fig. 6 Geometry identification results of Figs. 4d and 5

Table 1 Training process of BP-Adaboost classifier

Algorithm 1. BP-Adaboost

Input: A set of training samples $S=\{(x_i,y_i)\}$, where $i=1,\cdots,m$, x_i is the feature vector of the i^{th} sample, and $y_i \in \{-1,1\}$.

Output: A text/non-text classification model

1: Initialize the weights of training samples: $\omega_0(i)=1/m$.

2: **for** t = 1 to T **do**

3: Train the weak classifier h_t

4: Calculate the error rate of h_t: $e_t=\frac{1}{2}\sum_{i=1}^{m}\omega_{t-1}(i)\,|\,h_t(x_i)-y_i\,|$

5: Set the coefficient: $\alpha_t=\frac{1}{2}\ln[(1-e_t)/e_t]$

6: Update and normalize the weights, where Z_t is a normalization factor.

$$\omega_t(i)=\frac{\omega_{t-1}(i)}{Z_t}\exp\{-\alpha_t y_i h_t(x_i)\}$$

7: **end for**

8: **Return** $H(x)=\mathrm{Sign}[\sum_{t=1}^{T}{}_t h_t(x)]$, where Sign is a symbolic function.

series of candidate blocks, among which those containing text will be chosen as positive samples, otherwise negative ones. For each sample we extract features by *Local Binary Pattern* (LBP), Haar wavelet transform and *Gray Level Co-occurrence Matrix* (GLCM), respectively, and input them into the BP-Adaboost classifier. The methods of feature extraction are described as follows.

(1) LBP features

LBP is used to analyze the text texture, which is computed using current pixel and its all circular neighbor pixels and can be converted into a decimal number. The LBPs are computed clockwise by:

$$\mathrm{LBP}_{p,\,r}=\sum_{i=1}^{8}s(g_i-g_c)\times 2^{i-1} \tag{7}$$

where $s(x)=\{0,x<0\ \ 1,x\geq 0$, p denotes the number of circular neighbor pixels of the specified pixel and r is the radius of the circle, g_c and g_i are the intensities of the current pixel and its circular neighbor pixels, respectively. Figure 7 gives an example of computing LBP with $\mathrm{LBP}_{8\ ,4}=01101001=1\times 2^0+1\times 2^3+1\times 2^5+1\times 2^6=105$

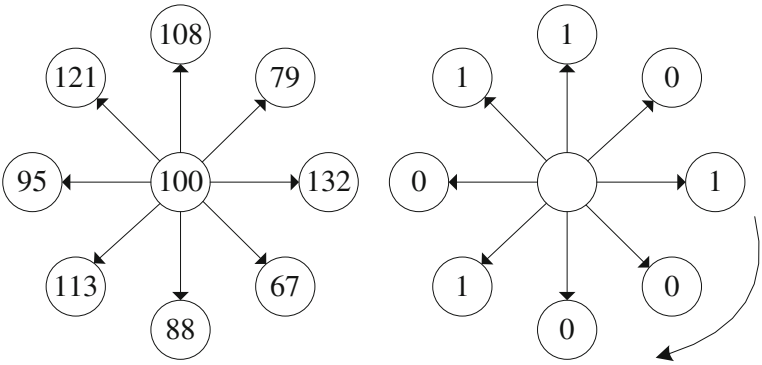


Fig. 7 Example of *Local Binary Pattern* (LBP) computation

For each candidate block, we firstly compute the *Number Of different LBPs* (*NOL*) and normalize it into [0, 1] by the maximum of *NOL* [6]. The NOL reflects intensity variation of candidate text block, and usually larger intensity variation will generate larger NOL value otherwise smaller one. Since text blocks usually have distinct texture features and intensity variations, their corresponding NOL values are usually larger than those of non-text blocks, which can be used for text verification. Then, for the i^{th} candidate block, we calculate the edge density λ and use it as a weight.

$$\lambda(i) = \frac{1}{M \times N} \sum_{(m,n) \in block(i)} \mathrm{SM}(m,n) \tag{8}$$

where SM(m, n) is the intensity value of point (m, n) on the SM, $M \times N$ is the size of the i^{th} block. Next, the *factor of LBPs* (FOL) is calculated according to the *NOL* and λ by Eq. (9), which reflects the combination of edge density and intensity variation of a candidate text block.

$$FOL(i) = \lambda(i) \times NOL(i) \tag{9}$$

Text blocks usually have larger FOL values while non-text blocks smaller ones. Thus, FOL can be used to distinguish text from non-text blocks. Kim et al. [6] and Wei et al. [16] identified the candidate text blocks by comparing the *FOL* with an empirical threshold. In this paper, we set *FOL* as a feature of the candidate text block and put it into the BP-Adaboost classifier, which will be more robust than setting a threshold as shown in the experiments.

(2) Haar wavelet transform features

For each candidate text blocks, Haar wavelet transform is used to obtain four sub-band images. We calculate the features of each sub-band image including mean μ, standard deviation σ and entropy E as follows:

$$\mu = \frac{1}{\mathrm{M} \times N} \sum_{i=1}^{\mathrm{M}} \sum_{j=1}^{\mathrm{N}} f(i,j) \tag{10}$$

$$\sigma = \left(\frac{1}{\mathrm{M} \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} [f(i,j) - \mu] \right)^{1/2} \tag{11}$$

$$E = -\sum_{i=1}^{\mathrm{M}} \sum_{j=1}^{\mathrm{N}} f(i,j) \log_2 f^2(i,j) \tag{12}$$

where $f(i,j)$ is the intensity value of pixel (i,j) in the sub-band images. Therefore, we obtain 12 dimension features for each candidate text block by Haar wavelet transform.

(3) GLCM features

GLCM is a texture analysis method used in many image processing applications [2–4, 16] which describes intensity differences of pixels in two-dimension spatial relationships. The obtained GLCM is a square array which has the same dimension as the number of intensity levels in the image to be analyzed. The value of each element in GLCM means frequency, denoted as $\mathrm{P}(i,j|d,\theta)$ and computed by Eq. (13), which indicates the intensity value relationship between two pixels (x_1,y_1) and $(x_1+\Delta x, y_1+\Delta y)$, apart from each other at a distance of d in the direction θ.

$$\mathrm{P}(i,j|d,\theta) = \sum_{x_1=1}^{m} \sum_{y_1=1}^{n} \begin{cases} 1, & if\ G(x_1,y_1) = i\ \ and\ \ G(x_1+\Delta x, y_1+\Delta y) = j \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where, G is the original grayscale image with size of $m \times n$ pixels, i and j the gray levels ranging in [0, L-1], and L the pre-set number of gray levels.

An example of calculating 4×4 GLCM at a distance of 1 in the direction of 0° is illustrated in Fig. 8.

In this paper, we compute GLCMs on four grayscale sub-band images obtained from wavelet transform, and for each of them, we first transform it from 256 intensity levels into 8 intensity levels and then compute the GLCMs at the distance of 1 in the directions of θ=0°, 45°, 90° and 135°, respectively, as suggested in [3]. Thus, 16 GLCMs are obtained, and for

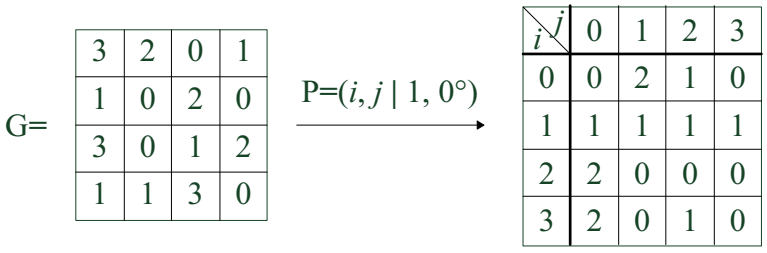


Fig. 8 Example of *Gray Level Co-occurrence Matrix* (GLCM) computation. (**a**) Image G with 4 intensity levels, (**b**) 4×4 GLCM of G at d=1, and θ=0

each GLCM, we compute four features such as Energy (Eg), Contrast (Con), Homogeneity (Hom) and Correlation (Cor) as follows:

$$\mathrm{Eg}(d,\theta) = \sum_i \sum_j P^2(i,j|d,\theta) \tag{14}$$

$$\mathrm{Con}(d,\theta) = \sum_i \sum_j (i\text{-}j)^2 P(i,j|d,\theta) \tag{15}$$

$$\mathrm{Hom}(d,\theta) = \sum_i \sum_j \frac{P^2(i,j|d,\theta)}{1+(i-j)^2} \tag{16}$$

$$\mathrm{Cor}(d,\theta) = \sum_i \sum_j \frac{(i-\mu_x)\left(j-\mu_y\right)P(i,j|d,\theta)}{\sigma_x\sigma_y} \tag{17}$$

where μ and σ are the mean and variance of the GLCM, respectively. In this way, 64 dimension GLCM features are obtained for each candidate text block.

By cascading the vectors obtained from the feature maps of LBP, wavelet transform, and GLCM, a 77-dimension feature vector is obtained for each candidate text block, which will be used to train the BP-Adaboost classifier. We apply the classifier to further verify candidate text blocks shown in Fig. 6 and obtain the corresponding results as illustrated in Fig. 9, from which we can find that the non-text blocks in Fig. 6 are removed and the text detection results are satisfactory.

4 Experimental results and discussion

In this section, we firstly describe the performance measures and datasets used in this paper. Then we conduct a series of experiments for parameters optimization and texture features performance evaluation, and compare our method with some existing methods. Finally, the limitations of the proposed method are presented.

Fig. 9 Results of BP-Adaboost identification on Fig. 6

4.1 Performance measures

To evaluate the text detection results quantitatively, we make the following definitions as done in method [13].

1) *Actual Text Blocks* (*ATB*): the number of text blocks counted manually.
2) *Truly Detected Blocks* (*TDB*): the number of detected blocks that contain at least one true character, i.e., a *TDB* may or may not fully enclose a text block.
3) *Falsely Detected Blocks* (*FDB*): the number of detected blocks that do not contain text.
4) *Text Blocks with Missing Data* (*MDB*): the number of detected blocks that miss more than 20% of the characters in a text line (*MDB* is a subset of *TDB*).

The performance measures are defined as follows:

$$\text{Recall}(R) = TDB/ATB \tag{18}$$

$$\text{Precision}(P) = TDB/(TDB + FDB) \tag{19}$$

$$\text{F-measure}(F) = 2 \times P \times R/(P + R) \tag{20}$$

$$\text{Misdetection Rate}(MDR) = MDB/TDB \tag{21}$$

where Recall (R) denotes the ratio between the number of truly detected blocks and that of the ground truth blocks, Precision (P) indicates the ratio between the number of truly detected blocks and that of all detected blocks, and the bigger the more accurate the detected text blocks. F-measure (F) takes both R and R into account, and the bigger the better the detected results. Misdetection Rate (*MDR*) is used to assess the rate of the detected blocks that miss some characters. The detected blocks are more integrated and the detection results are better when *MDR* is smaller.

4.2 Datasets

To illustrate the effectiveness of the proposed method on video text detection, we conduct some experiments on two datasets: our manually selected dataset and the publicly available dataset.

Our manually selected dataset contains 630 images with each size of 720× 540 pixels taken from 7 different programs including news, advertisements, entertainments, TV programs, cartoons, sports and the others, each of which includes 90 images. Table 2 shows the program and language distributions of text blocks in our manually selected dataset. For example, there are 533 text blocks altogether in the 90 TV program images, of which 200 in Chinese, 312 in English, and 21 in Korea. We select 210 images of the manually selected dataset to construct training set containing 1380 text blocks and 2206 non-text blocks. The training set is used for BP-Adaboost and the remaining 420 images are used for testing.

Table 2 Program and language distributions of text blocks in our manually selected dataset.

Languages	News	Advertisement	Entertainment	TV program	Cartoon	Sports	Others	Total
Chinese	460	237	222	200	18	266	258	1661
English	0	110	54	312	18	272	82	848
Japanese	0	7	0	0	612	0	0	619
Korea	0	39	123	21	0	0	6	189
Total	460	393	399	533	648	538	346	3317

The publicly available dataset is the Microsoft Asia dataset [5] containing 45 video images with each size of 352×288 pixels and all of them are used for testing.

All the ground truth text blocks of the aforementioned datasets are labeled manually.

4.3 Parameters optimization and features evaluation

In this section, we will describe how we set values of parameters used in this paper, and then perform texture features performance evaluation.

In Section 3.1, we compute MGD by setting parameter N, the preferred value of which is slightly larger than the stroke width of the largest character to be detected according to [17]. Some related works [11, 13, 16, 17] set N as 5, 11 and 21, and we conduct experiment to obtain the best N value. Figure 10a is the original video image, and Figs. 10b-d respectively illustrate binary images of MGD with N set as 5, 11 and 21, from which we can see that when N is small (N=5 for example), text can be extracted from the background and characters will not connect to the neighbor ones seriously, while with larger N, characters will connect to each other more seriously. In this paper, we combine binary image of f_h and those of MGDs (Figs. 10b-d) to extract text pixels and eliminate some noise at the same time, followed by morphological operations to obtain text localization results as shown in Figs. 10e-g, from which we can see that text localization results will not be influenced violently with different N values, and it is unnecessary to set N with a larger value which will provide much more redundant information. In addition, method in [17] set N as 21 to extract characters with large stroke width as mentioned above, which can be realized by our morphological operations. In our experiment, N is set as 5.

As mentioned in Section 3.2, a threshold T_1 is needed for boundary refinement. Figure 11 shows the comparison of text boundary refinement results with different T_1 values. If T_1 is too small (T_1=0 for example), text blocks will be under-segmented as the green ellipse-marked regions shown in Fig. 11a, while if T_1 is too large (T_1=0.1 for example), text blocks will be over-segmented as the green ellipse-marked region shown in Fig. 11c. Thus, we experientially set T_1 as 0.05 and obtain more accurate text boundary refinement results as shown in Figs. 11b and d.

As we known, text in video image with too small size is hard to read for both human and OCR engines [19]. In this paper, we assume the minimal width (T_2) and height (T_3) of recognizable text are 5 and 10 pixels, respectively. Since we mainly aim at detecting text in horizontal direction as mentioned in Section 1, candidate text block located at vertical direction will be treated as non-text block to be removed, and therefore T_4 is set as 1 in our method.

In Section 3.3, a lot of texture features are extracted for classifier training. In order to evaluate the performance of BP-Adaboost classifiers trained by these different texture features and obtain the most robust one, we perform a feature evaluation experiment on a sub-set

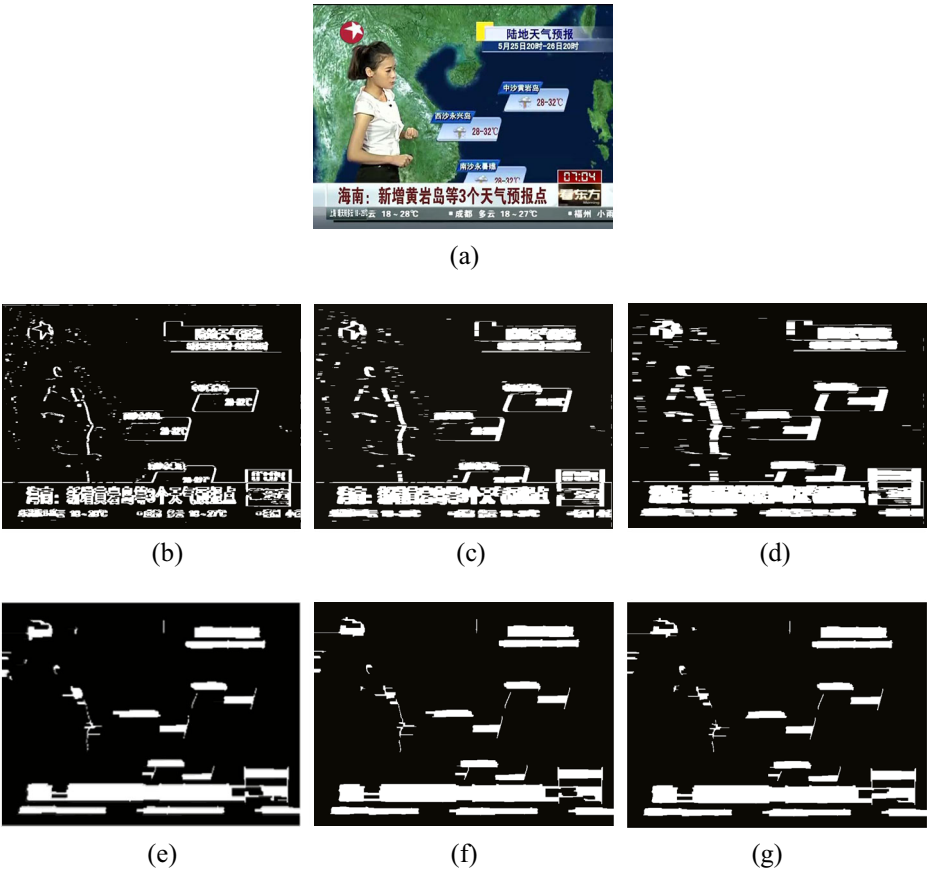


Fig. 10 Comparison of text localization results with different N values. (**a**) Original video image, (**b-d**) binary images of MGD with N set as 5, 11 and 21, respectively, and (**e-g**) text localization results by combining binary image of f_h with (**b**), (**c**), and (**d**), respectively.

containing 90 images selected from 420 testing images in our manually selected dataset as mentioned in Section 4.2.

Figure 12 gives the performance curves of BP-Adaboost classifiers trained by different features, where hit rate represents the ratio of correctly classified samples in the text block set, and false positive rate is the ratio of incorrectly classified samples in the non-text block set.

Fig. 11 Comparison of text boundary refinement results with different T_1 values. (**a-b**) T_1 set as 0 and 0.05, respectively, and (**c-d**) T_1 set as 0.1 and 0.05, respectively.

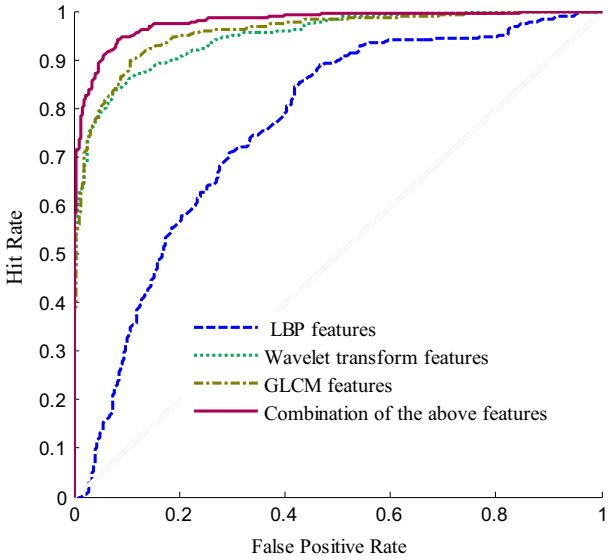


Fig. 12 Performance curves of BP-Adaboost classifiers trained by different features

From Fig. 12 we can find the BP-Adaboost classifier trained by GLCM features is more robust than that trained by LBP or wavelet transform features, which is mainly because the GLCM features are obtained from the wavelet-transformed images as mentioned in Section 3.3.2 and therefore they describe both the values and the locations of wavelet-transformed images. Thus, in this paper, we use BP-Adaboost identification with all the aforementioned 77 dimensional features combined together.

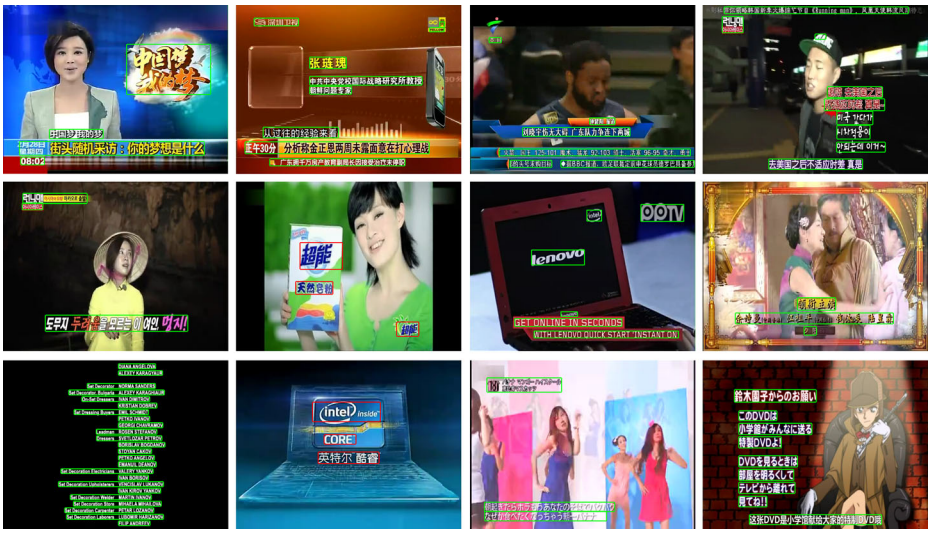


Fig. 13 Text detection results of the proposed method on randomly chosen video images from our manually selected dataset

4.4 Text detection results on different datasets

To show the effectiveness of the proposed method, we have conduct some experiments on the aforementioned two datasets and compare the text detection results with some existing methods.

Firstly, we perform the proposed method on our manually selected dataset. Figure 13 gives the text detection results of the randomly chosen video images in this dataset, from which it

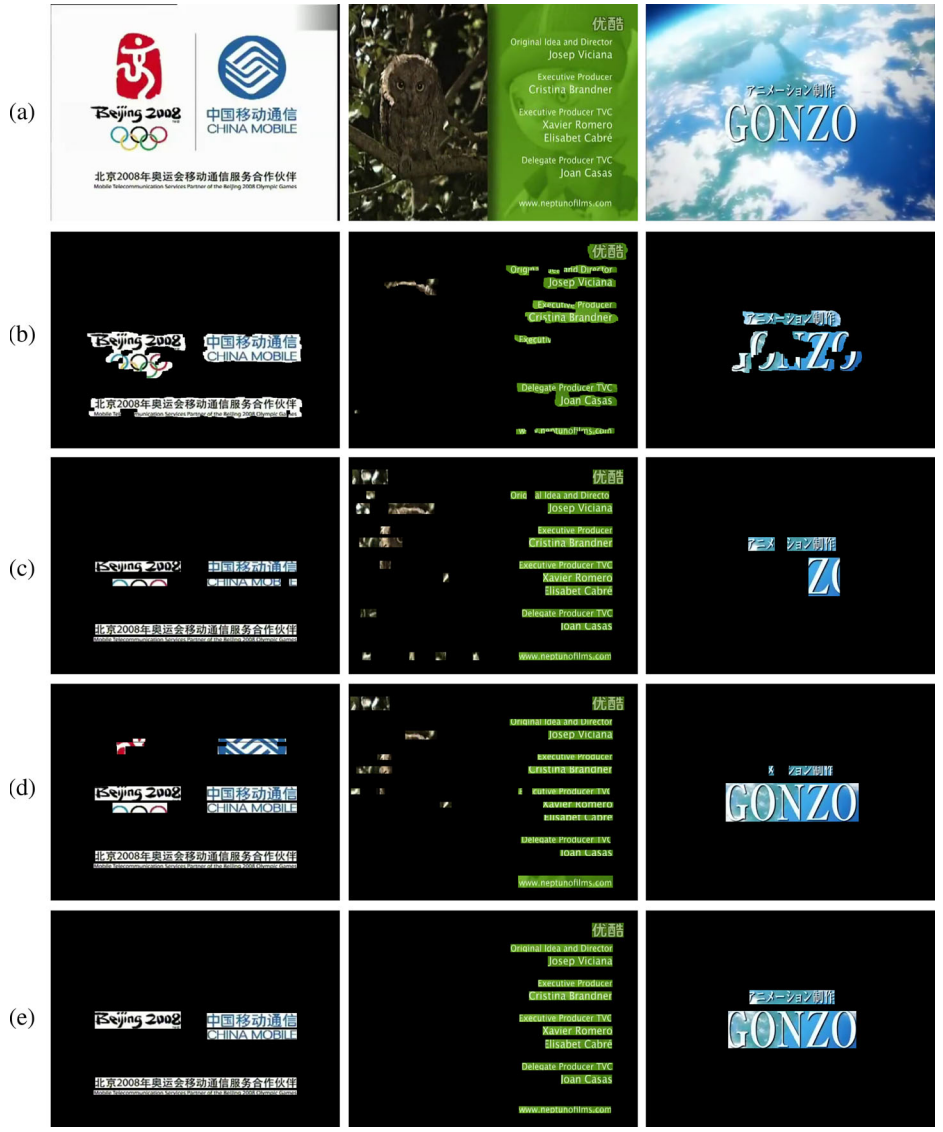


Fig. 14 Text detection result comparison of the proposed method with three existing methods. (**a**) Original images, (**b-d**) detection results of the methods [11, 13, 16], respectively, and (**e**) detection results of the proposed method

Table 3 Text detection performance comparison of different methods for video images from different programs on manually selected dataset

Methods	Performance	News	Advertisement	Entertainment	TV program	Cartoon	Sports	Others	Average
Shivakumara et al. [13]	*R*	0.58	0.51	0.63	0.56	0.58	0.51	0.49	0.51
	P	0.85	0.80	0.87	0.78	0.88	0.81	0.83	0.83
	F	0.69	0.62	0.73	0.65	0.70	0.63	0.62	0.63
	MDR	0.19	0.13	0.24	0.02	0.07	0.10	0.15	0.11
Phan et al. [11]	*R*	0.78	0.71	0.86	0.92	0.95	0.64	0.78	0.78
	P	0.64	0.68	0.58	0.72	0.84	0.67	0.66	0.69
	F	0.70	0.69	0.69	0.81	0.89	0.65	0.72	0.73
	MDR	0.10	0.20	0.11	0.03	0.03	0.06	0.10	0.08
Wei et al. [16]	*R*	0.90	0.86	0.88	0.80	0.95	0.85	0.84	0.87
	P	0.67	0.83	0.69	0.87	0.88	0.77	0.73	0.78
	F	0.77	0.84	0.77	0.83	0.91	0.81	0.78	0.82
	MDR	0.07	0.07	0.15	0.02	0.03	0.05	0.07	0.06
Ours	***R***	**0.91**	**0.77**	**0.88**	**0.93**	**0.95**	**0.81**	**0.89**	**0.91**
	P	**0.89**	**0.96**	**0.94**	**0.98**	**0.99**	**0.97**	**0.99**	**0.96**
	F	**0.90**	**0.85**	**0.91**	**0.95**	**0.97**	**0.88**	**0.94**	**0.93**
	MDR	**0.02**	**0.03**	**0.05**	**0.01**	**0.02**	**0.03**	**0**	**0.03**

Results are labelled in bold

Table 4 Text detection performance comparison of different methods for video images in different languages on manually selected dataset

Methods	Performance	Chinese	English	Japanese	Korea
Shivakumara et al. [13]	*R*	0.53	0.29	0.57	0.60
	P	0.80	0.86	0.89	0.90
	F	0.64	0.43	0.69	0.72
	MDR	0.12	0.12	0.06	0.21
Phan et al. [11]	*R*	0.76	0.81	0.95	0.28
	P	0.60	0.77	0.86	0.61
	F	0.67	0.79	0.90	0.38
	MDR	0.11	0.06	0.04	0.19
Wei et al. [16]	*R*	0.88	0.80	0.95	0.92
	P	0.72	0.81	0.90	0.82
	F	0.79	0.81	0.92	0.87
Ours	***R***	**0.91**	**0.88**	**0.94**	**0.96**
	P	**0.94**	**0.99**	**0.99**	**0.97**
	F	**0.92**	**0.93**	**0.96**	**0.97**
	MDR	**0.03**	**0.01**	**0.03**	**0.02**

Results are labelled in bold

can be observed that the proposed method is effective for text detection on video images from different programs and in different languages.

To show the advantages of the proposed method, we compare it with three existing methods [11, 13, 16]. The method in [13] introduces a novel skeleton based strategy detecting text in arbitrary direction and achieves good results. The method in [11] applies both edge and gradient features, and the method in [16] uses gradient features and *Support Vector Machine* (SVM) classifier for text detection. We implement these three existing methods and test them on the manually selected dataset. The corresponding parameters are set according to their recommended values in the papers. Text detection results of these methods on randomly chosen images on our manually selected dataset are shown in Fig. 14, from which it can be clearly observed that the methods in [11, 13, 16] get some false text blocks and sometimes over-segment the text blocks because they cannot remove the non-text blocks effectively when the characteristics between the text blocks and non-text blocks are similar. It can also be easily found that the proposed method is robust to video text detection in complex background and low contrast images.

We compare the proposed method with three existing methods [11, 13, 16] on our manually selected dataset by the performance measures. Tables 3 and 4 show the text

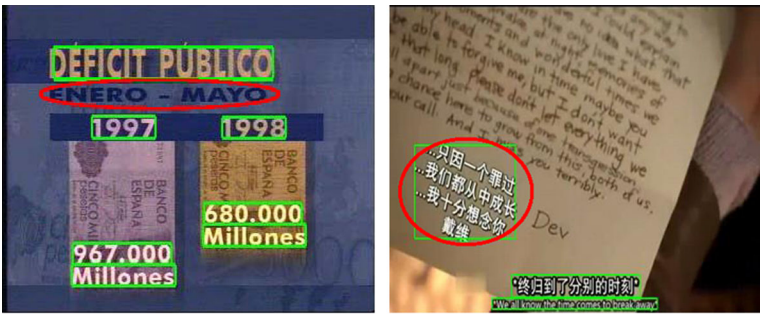
Fig. 15 Text detection results of the proposed method on some randomly chosen video images from Microsoft Asia dataset

Table 5 Text detection performance comparison of different methods on Microsoft Asia dataset

Methods	*R*	*P*	*F*	*MDR*
Shivakumara et al. [13]	0.93	0.81	0.87	0.07
Liu et al. [9]	0.75	0.54	0.63	0.16
Cai et al. [1]	0.69	0.43	0.53	0.13
Wei et al. [16]	0.90	0.83	0.86	0.01
Phan et al. [11]	0.90	0.72	0.80	0.02
Wong et al. [17]	0.51	0.75	0.61	0.13
Mariano et al. [10]	0.47	0.44	0.45	0.44
Ours	**0.90**	**0.91**	**0.91**	**0**

Results are labelled in bold

detection performance comparison of different methods for video images from different programs and in different languages, respectively, and results of our methods are labelled in bold. From Table 3 we can find that, the proposed method achieves high *R* values on News, TV programs and Cartoon images as 0.91, 0.93 and 0.95, respectively, while relatively low *R* values on advertisement and sports images as 0.77 and 0.81, respectively, which is mainly because most of text in News, TV programs and Cartoon images is graphics, relatively regular and rich in distinct texture features in contrast to advertisement and sport images. We can also observe that our method achieves low *P* value on News images mainly because many text-like blocks exist in these images such as human collars, faces, windows and so on. Table 3 also illustrates the cartoon images achieve the highest *F* value, because most of the text blocks in cartoon images are graphics text blocks with high contrast to the background, and have large gradient information and therefore can be easily detected by the proposed method. However, the proposed method achieves the lowest *F* value on the advertisement images because of complex background and multiple scene text blocks in them and therefore difficult to detect. It is also clear from Table 3 that the average *R*, *P* and *F* values of the proposed method are higher than those of the methods in [11, 13, 16], especially for indicator *P*, because we firstly combine f_h and MGD for text localization, which can eliminate some text-like horizontal noise at pixel level, and



(a) (b)

Fig. 16 Limitations of the proposed method. (**a**) Scene text failed to be detected, and (**b**) inaccurately detected boundary of non-horizontal text.

then use geometrical identification and classifier based strategy for candidate text verification, which can further remove non-text blocks at block level. In addition, the average *MDR* of the proposed method is the lowest, which indicates the proposed method detects text blocks more accurately than the other three methods.

From Table 4, we can see that the proposed method can detect text in different languages and receives relatively high *R* values for text in Japanese and Korea as 0.94 and 0.96 respectively, because these two kinds of images are mainly from Cartoon and TV programs which are in high contrast and simple background and can therefore be easily detected. It also can be observed that text in Chinese and Japanese receives relatively high MDR values by our method as both 0.03, because unlike text in English that one character usually contains one CC, character in these two languages usually contains more than one CC, which may therefore result in over segmentation and high MDR value.

In addition to our selected dataset, we also perform the proposed method on the Microsoft Asia dataset. Figure 15 shows some randomly chosen experimental results, from which it can also be easily observed that the proposed method is effective on the Microsoft Asia dataset.

The quantitatively comparison of the proposed method with some other methods on the Microsoft Asia dataset are shown in Table 5, in which the results of the methods in [1, 3, 4, 6, 20] are obtained from the method in [13] and those of the methods in [11, 16] are obtained from our own codes. To clearly show results of our method, we label them in bold as shown in Table 5. Obviously, the proposed method outperforms the existing methods mentioned above on the Microsoft Asia dataset. From Table 5, it is easy to find that the proposed method has the highest *P* and *F* values as both 0.91, the lowest MDR value and the second highest *R* value, which are mainly because most of text blocks in this dataset are high contrast graphics text and we perform text boundary refinement for each labeled block, and therefore, fewer characters will be missed and text boundary will not be much influenced by each other.

4.5 Limitations of the proposed method

The proposed method can detect most video text, especially for graphics text, but there are still some limitations:

(1) The method uses K-means to classify f_h and MGD into two clusters, so when there exist graphics text with larger gradient values and scene text with smaller ones in the image at the same time, the scene text will be misclassified into non-text cluster, as shown in Fig. 16a, where the red ellipse-marked text region fails to be detected.
(2) The method uses projection profile to refine text boundary, so when text aligns in non-horizontal direction, its boundary cannot be detect accurately, as the red ellipse-marked text region in Fig. 16b.

5 Conclusions and future work

In this paper, we propose an automatic video text detection method based on BP-Adaboost classifier. It is primarily composed of three steps. Firstly, K-means is applied to classify pixels into two clusters, text regions and non-text regions, according to the gradients of grayscale images, and morphological operations are then used to connect the scattered pixels in text

regions to form candidate text blocks. Subsequently, each candidate text region is refined according to the projection profile on the corresponding region in SM. Finally, the false positives are eliminated by the geometry and BP-Adaboost identifications. The proposed method is performed on two different datasets, our manually selected dataset and the publicly available Microsoft Asia dataset. The experimental results show that the proposed method can detect video text in different font sizes, styles and colors, and outperforms many existing methods. The main disadvantages of the proposed method include failure in scene text detection when there exist both graphics and scene text with low contrast in image at the same time, and in obtaining inaccurate boundary of non-horizontal text. Our future work will focus on the detection of multi-oriented text and make the method more adaptive to those specific and challenging situations.

Acknowledgments This work is partly supported by the National Natural Science Foundation of China (Grant Nos. 61172184, 61173122, 61174210, 61379107, and 61402539), Key Project of Hunan Provincial Natural Science Foundation of China (Grant No. 12JJ2038), Program for New Century Excellent Talents in University of Education Ministry in China (Grant No. NCET-13-0603), Specialized Research Fund for the Doctoral Program of Higher Education in China (Grant No. 20130162110016), Program for Hunan Province Science and Technology Basic Construction (Grant No. 20131199), and China Postdoctoral Science Foundation (Grant No. 2012 M521554), the Fundamental Research Funds for the Central Universities of Central South University (Grant No. 2015zzts052).

References

1. Cai M, Song J, Lyu MR (2002) A new approach for video text detection. In: Proceedings of IEEE International Conference on Image Processing, pp I-117
2. Gui W, Liu J, Yang C, Chen N, Liao X (2013) Color co-occurrence matrix based froth image texture extraction for mineral flotation. Miner Eng 46:60–67
3. Haralick RM, Shanmugam K, Dinstein IH (1973) Textural features for image classification. IEEE Trans Syst Man Cybern 6:610–621
4. He M, Yang C, Wang X, Gui W, Wei L (2013) Nonparametric density estimation of froth colour texture distribution for monitoring sulphur flotation process. Miner Eng 53:203–212
5. Hua XS, Wenyin L, Zhang HJ (2004) An automatic performance evaluation protocol for video text detection algorithms. IEEE Trans Circ Syst Vid 14(4):498–507
6. Kim W, Kim C (2009) A new approach for overlay text detection and extraction from complex video scene. IEEE Trans Image Process 18(2):401–411
7. Li Z, Liu G, Qian X, Guo D, Jiang H (2011) Effective and efficient video text extraction using key text points. IET Image Process 5(8):671–683
8. Liu X, Wang W (2012) Robustly extracting captions in videos based on stroke-like edges and spatio-temporal analysis. IEEE Trans Multimed 14(2):482–489
9. Liu C, Wang C, Dai R (2005) Text detection in images based on unsupervised classification of edge-based features. In: Proceedings of IEEE International Conference on Document Analysis and Recognition, pp 610–614
10. Mariano VY, Kasturi R (2000) Locating uniform-colored text in video frames. In: Proceedings of IEEE International Conference on Pattern Recognition, pp 539–542
11. Phan TQ, Shivakumara P, Tan CL (2009) A Laplacian method for video text detection. In: Proceedings of IEEE International Conference on Document Analysis and Recognition, pp 66–70
12. Qian X, Wang H, Hou X (2014) Video text detection and localization in intra-frames of H. 264/AVC compressed video[J]. Multimed Tools Appl 70(3):1487–1502
13. Shivakumara P, Phan TQ, Tan CL (2011) A Laplacian approach to multi-oriented text detection in video. IEEE Trans Pattern Anal Mach Intell 33(2):412–419
14. Shivakumara P, Sreedhar RP, Phan TQ, Lu S, Tan CL (2012) Multioriented video scene text detection through Bayesian classification and boundary growing. IEEE Trans Circ Syst Vid 22(8):1227–1235
15. Suzuki K, Horiba I, Sugie N (2003) Linear-time connected-component labeling based on sequential local operations. Comput Vis Image Und 89(1):1–23

16. Wei YC, Lin CH (2012) A robust video text detection approach using SVM. Expert Syst Appl 39(12): 10832–10840
17. Wong EK, Chen M (2003) A new robust algorithm for video text extraction. Pattern Recognit 36(6):1397–1406
18. Wu Y, Shivakumara P, Wei W, et al. A new ring radius transform-based thinning method for multi-oriented video characters [J]. Int J Doc Anal Recog (IJDAR), 2015: 1–15
19. Yang H, Quehl B, Sack H (2014) A framework for improved video text detection and recognition[J]. Multimed Tools Appl 69(1):217–245
20. Zhao M, Li S, Kwok J (2010) Text detection in images using sparse representation with discriminative dictionaries. Image Vision Comput 28(12):1590–1599

Hui Wu received the B.S. degree from Northeastern University, Shenyang, China, in July 2011, the M.S. degree from Central South University, Changsha, China, in June 2014. She is currently a Ph.D. candidate at Central South University. Her research interests include machine learning, pattern recognition and image processing.

Bei-ji Zou received the B.S. degree from Zhejiang University, Hangzhou, China, in June 1982, the M.S. and Ph. D. degrees in computer science and technology from Tsinghua University in 1984 and Hunan University in 2001, respectively. He joined the school of computer and communication at Hunan University in 1984, where he became Associate Professor in 1997, Professor in 2001. He served as the vice dean there since 1997. He is currently a Professor and also served as the dean at the school of Information Science and Engineering in Central South University. He is the founder of the Computer Vision and Virtual Reality Technology Lab. at Central South University. His research interesting is focus on computer graphics, image processing and virtual reality technology. Until now he has published more than 100 papers in journals.

Yu-qian Zhao received Ph.D. degree from Central South University, Changsha, China, in 2006. He engaged in postdoctoral research at Xiangya School of Medicine, Changsha, China, from May 2007 to June 2009, and at New Jersey Institute of Technology, Newark, New Jersey, USA, from July 2009 to July 2010. He is currently a Professor in the School of Geosciences and Info-Physics, Central South University. His research interests include image processing, pattern recognition, computer vision, video text detection, image forensics, and computer-aided diagnosis.

Hong-pu FU received the B.S. degree in mechanical engineering from Xiangtang University, China, in 1996, M.S. degree in computer application from Hunan University, China, in 2006. And He is now a Ph.D. candidate in computer application technology at Central South University, China. His research interests include object detection, recognition and machine learning.