Multimed Tools Appl
DOI 10.1007/s11042-017-4619-8

Automatic video superimposed text detection based on Nonsubsampled Contourlet Transform

Xiaodong Huang [1]

Received: 8 September 2016 / Revised: 14 March 2017 / Accepted: 16 March 2017

Abstract Compared with other video semantic clues, such as gestures, motions etc., video text generally provides highly useful and fairly precise semantic information, the analysis of which can to a great extent facilitate video and scene understanding. It can be observed that the video texts show stronger edges. The Nonsubsampled Contourlet Transform (NSCT) is a fully shift-invariant, multi-scale, and multi-direction expansion, which can preserve the edge/ silhouette of the text characters well. Therefore, in this paper, a new approach has been proposed to detect video text based on NSCT. First of all, the 8 directional coefficients of NSCT are combined to build the directional edge map (DEM), which can keep the horizontal, vertical and diagonal edge features and suppress other directional edge features. Then various directional pixels of DEM are integrated into a whole binary image (BE). Based on the BE, text frame classification is carried out to determine whether the video frames contain the text lines. Finally, text detection based on the BE is performed on consecutive frames to discriminate the video text from non-text regions. Experimental evaluations based on our collected TV videos data set demonstrate that our method significantly outperforms the other 3 video text detection algorithms in both detection speed and accuracy, especially when there are challenges such as video text with various sizes, languages, colors, fonts, short or long text lines.

Keywords Video text · Nonsubsampled Contourlet Transform · Text detection · Text frame classification

1 Introduction

With the rapid growth of digital images and videos, the researchers pay more attention to solve the basic problem of video understanding, search and retrieval in recent years. For the content-based image analysis, text, video and audio clues are most frequently utilized. Among those

✉ Xiaodong Huang
hxd@cnu.edu.cn

[1] Capital Normal University, Beijing 100048, China

Published online: 25 March 2017

clues, text generally provides the most useful facts and can be used to infer the video semantic information, which is beneficial to the further video understanding, search and retrieval. Therefore, video text recognition definitively play a key role in these fields.

There are two kinds of textual information in the video: the superimposed text and the scene text. In videos, the superimposed texts (e.g., captions in broadcast news programs) are those added by video editors and normally can be used to infer the semantic content of videos. On the contrary, the scene text is inherent text in the video captured by the video camera.

According to our analysis, the superimposed texts keep stable motion characteristics in video, it will not move no matter how the background changes. However, scene text is not stationary on consecutive frames. Due to the illumination changes, the same scent text line will show different colors. The superimposed text alignment is either horizontal or vertical. However, the scene text alignment is various due to shooting angles.

Therefore, the detection methods for scene text may be not suitable for superimposed text. In this paper, the author mainly focuses on the superimposed text detection.

According to the general definition [9], the video text recognition consists of four primary steps: detection, localization, extraction, and recognition.

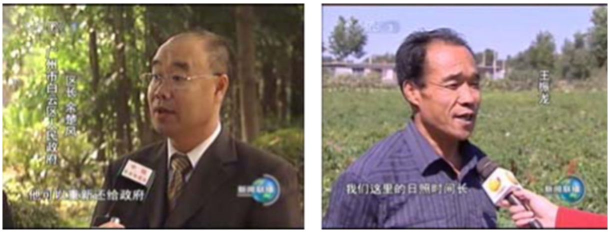
Since text detection is the premise for later stages and is critical to the overall system performance, a large number of approaches [4–11, 13–21] have been proposed to address this issue. For the video text detection, text frame classification is also a key stage, which is carried out before the text detection to determine whether the video frames contain the text lines. Because the text frame classification can ensure that the text detection will not be performed on the non-text frame, it can accelerate the text detection speed to a great degree. In this paper, the author concentrates on elaborating text frame classification and text detection.

In general, the video background is complicated. Some objects have similar appearance with video text, such as foliage [Fig.1(a)] and some objects own similar texture characteristics with video text, such as stripes of clothes [Fig.1(b)]. Although many methods have been proposed for text detection in the last decade [4–11, 13–21], due to complicated background, few of them can achieve great accuracy in any situations.

Meanwhile, the detection speed is also quite important for text detection. Thus, striking a balance of video text detection accuracy and speed is of vital significance.

Therefore, this paper proposes a novel method based on NSCT, which can to a great extent suppress background noises and improve the detection accuracy. Meanwhile, we perform the text frame classification in multi frames to accelerate the text detection speed to a great degree.

The remaining of this paper has the following structures. In Section II, the related works are reviewed. In Section III, the process of Nonsubsampled Contourlet Transform [3] is reviewed and discussed to retrieve the directional edge map. In terms of the directional edge map, the video text frame classification and text detection are performed, which is presented in


(a) foliage

(b) stripes of clothes

Fig. 1 Examples with Complicated Background

Section IV. Experimental results are described and elaborated in Section V. In the final part, the conclusion is drawn.

2 Related work

At present, the approaches to detect video text fall into two categories. The first category [8, 15, 17, 19] is frequency-based methods. Palaiahnakote Shivakumara et al. [17] proposed a video text detection method based on wavelet transform, statistical features and central moments for both superimposed and scene text. The method utilized wavelet single level decomposition LH, HL and HH subbands to retrieve features, which were fed to k-means clustering to discriminate the text from the background. Then the text regions were located using the projection profiles. Then, in 2010, they [19] further presented a new Fourier-statistical feature (FSF) in RGB space for detecting video text. And they also performed K-means clustering on the FSF features and discriminated text from the background. However, the method cannot be used to detect the non-horizontal text.

Ahsen Raza et al. [15] proposed a superimposed text detection and extraction method from images. Because the method did not depend on the specific features of a particular script or alphabet, the text detection/extraction method can process multilingual characters. The text detection utilized stepwise methodologies, which included Window based Discrete Stationary Wavelet Transform (D-SWT), Gabor Filters based Processing, Fast Fourier Transform (FFT) based thresholding and Fractal Dimension based Filtering.

The second category [4–7, 9–11, 13, 14, 16, 18, 20, 21] is spatial-based methods, which utilize the low level features to detect text. Most researchers have used edge, corners and stroke features for text detection in video/images.

Edge-based methods [5, 7, 9, 13, 14, 16, 18, 21]. Lyu et al. [9] presented an edge-based text detection approach, which focused on handling multilingual texts. The text detection is performed by edge detection, local thresholding, and hysteresis edge recovery. However, in the “text-like” texture, the text detection method will cause the high false alarm rate. Jing Zhang et al. [21] utilized Histogram of Oriented Gradient to retrieve text edges and locate the candidate character blocks, and then Graph Spectrum was used to get global relationship among candidate blocks and cluster candidate blocks were used to locate the boundaries of text regions. Kim et al. [7] proposed a novel method to detect the overlay text based on the transition map, which was introduced based on logarithmical change of intensity and modified saturation. Overlay text region update between frames was used to reduce the processing time.

Corner-based methods [4, 10, 20]. Mohieddin Moradi et al. [10] proposed a new Farsi/Arabic text detection and localization approach. First, with the help of edge extraction, artificial corners were obtained and font size estimation was performed. Second, by combining DCT coefficients, texture intensity picture was created, and a new Local Binary Pattern (LBP) picture was introduced to describe the obtained texture pattern. The input image was then divided into macro blocks and some features were extracted from them and fed into SVM to categorize them into text and non-text groups. Finally, the candidate text blocks undergo project profile analysis and empirical rules for text localization. However, the method need to adjust its parameters and thresholds for English and Chinese languages.

Li Sun et al. [20] presented a text detection approach based on corner responses. In terms of different responses among the text regions and non-text regions, text candidates were retrieved based on the block of corner responses. Finally, text line was located accurately via the projection of corner responses. However, beside the video text produces the corner responses, the similar backgrounds also produce corner responses which will pose the interference for the text detection.

Stroke-based methods [6, 11]. Cheolkon Jung et al. [6] presented a stroke filter to detect texts. They utilized the stroke filter to remove non-text candidates which owned strong edges. Then they used spatial-similarity CCA to retrieve the text candidates.

Ali Mosleh et al. [11] proposed a video text detection method. First, text locations in each frame were found via an unsupervised clustering performed on the connected components produced by the stroke width transform (SWT). Next, the motion patterns of the text objects of each frame are analyzed to localize video texts.

In [11], considering that the video text appears in a consequence of frames with specific motion properties compared to the rest of the video, they employed CAMSHIFT algorithm to perform the tracking and motion analysis scheme in order to specify the video text regions. Meanwhile, in [4], they performed a detailed analysis of motion patterns of video text, and showed that the superimposed and scene text exhibit different motion patterns on consecutive frames. Then they defined Motion Perception Field (MPF) to represent the text motion patterns.

The video text shares similarities with motion objects. As a result, the motion analysis [1, 2, 12] is of vital importance for video text detection.

Olga Barinova et al. [1] proposed a framework for detecting multiple object instances in images. Their framework shared the simplicity and wide applicability of the Hough transform but bypasses the problem of multiple peak identification in Hough images and permits detection of multiple objects without invoking nonmaximum suppression heuristics. They also presented the line and pedestrian detection based on greedy inference within their framework. However, compared to the traditional Hough transform, the method increased computation cost, which is not suitable for fast object detection, especially for the video text detection.

Duc-Son Pham et al. [12] addressed that dynamic background can be detected reliably and efficiently using simple motion features and in the presence of similar but meaningful events, such as loitering. Inspired by the tree aerodynamics theory, they proposed a novel method named local variation persistence (LVP), that captured the key characteristics of swaying motions. They derived a computationally efficient algorithm to solve the optimization problem, the solution of which was then used to form a powerful detection statistic. However, the suggested background detection method is suitable for the static camera. When the background of video changes dramatically, it does not work well. Due to the fast changed background of video, [12] cannot be used to extract the background of video text frame.

Zijing Chen et al. [2] presented a novel robust object tracking technique depended on subspace learning-based appearance model. In this model, mask templates were introduced, helping greatly reduce the complexity of the system and with a theoretical guarantee of the efficiency of the solution. They also exploited the dynamic information of the tracking target which significantly improved the tracking accuracy and coverage of target. However, the method cannot handle the multi object tracking tasks. When the video frame contains multi video texts, the suggested method cannot process the video text tracking.

Based on above analysis, we found that most of previous methods were easily interfered by background noises. Meanwhile, the stable motion characteristics of video text need to be utilized to detect text in video.

Therefore, we propose a method to detect text using NSCT, which can to a great extent suppress background noises. First of all, the 8 directional coefficients of NSCT are combined to build the directional edge map (DEM), which can keep the horizontal, vertical and diagonal edge features and suppress other directional edge features. For the DEM, we retrieve the binary directional edge map (BDEM). The pixels in the BDEM can be regarded as the candidate text characters pixels. Then various directional pixels of BDEM are integrated into a whole binary image (BE). Because of the stable motion patterns of video text, text frame classification based on the BE is carried out to determine whether the video frames contain the text lines. Finally, text detection based on the BE is performed on consecutive frames to discriminate the video text from non-text regions. The framework of the proposed method is listed in Fig.2.

3 Directional edge map

Arthur L. Cunha et al. [3] proposed an overcomplete transform that was defined as the Nonsubsampled Contourlet Transform (NSCT). The NSCT was a fully shift-invariant, multi-scale, and multi-direction expansion that own better directional frequency localization and faster implementation.

According to the analysis of [3], NSCT could retrieve the geometrical information pixel by pixel from the coefficients. In [3], Arthur L. Cunha et al. divided the pixels into three categories: strong edges, weak edges, and noise. They proposed that the strong edges represented those pixels with large magnitude coefficients in all subbands; the weak edges represented those pixels with large magnitude coefficients in some directional subbands but small magnitude coefficients in other directional subbands within the same scale; the noise represented those pixels with small magnitude coefficients in all subbands. Therefore, NSCT is suitable for retrieving the edge features in some directional subbands.

Based on the observation of the NSCT coefficients in NSCT multi-scale decomposition, we find that the NSCT coefficients in some directional subbands can keep the video texts characters pixels as strong edge, as well as the background pixels as weak edge or noise. As a result, NSCT coefficient is beneficial to text detection in the text image.

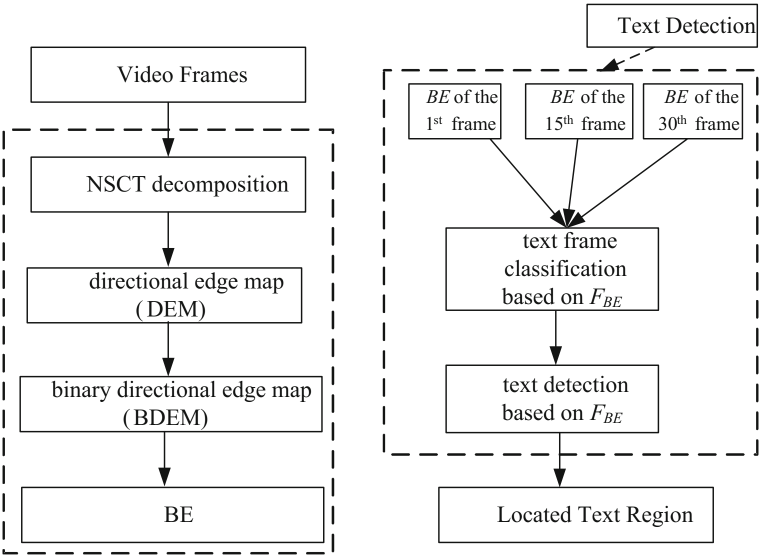

Video Frames
NSCT decomposition
directional edge map (DEM)
binary directional edge map (BDEM)
BE
Text Detection
BE of the 1st frame
BE of the 15th frame
BE of the 30th frame
text frame classification based on F_{BE}
text detection based on F_{BE}
Located Text Region

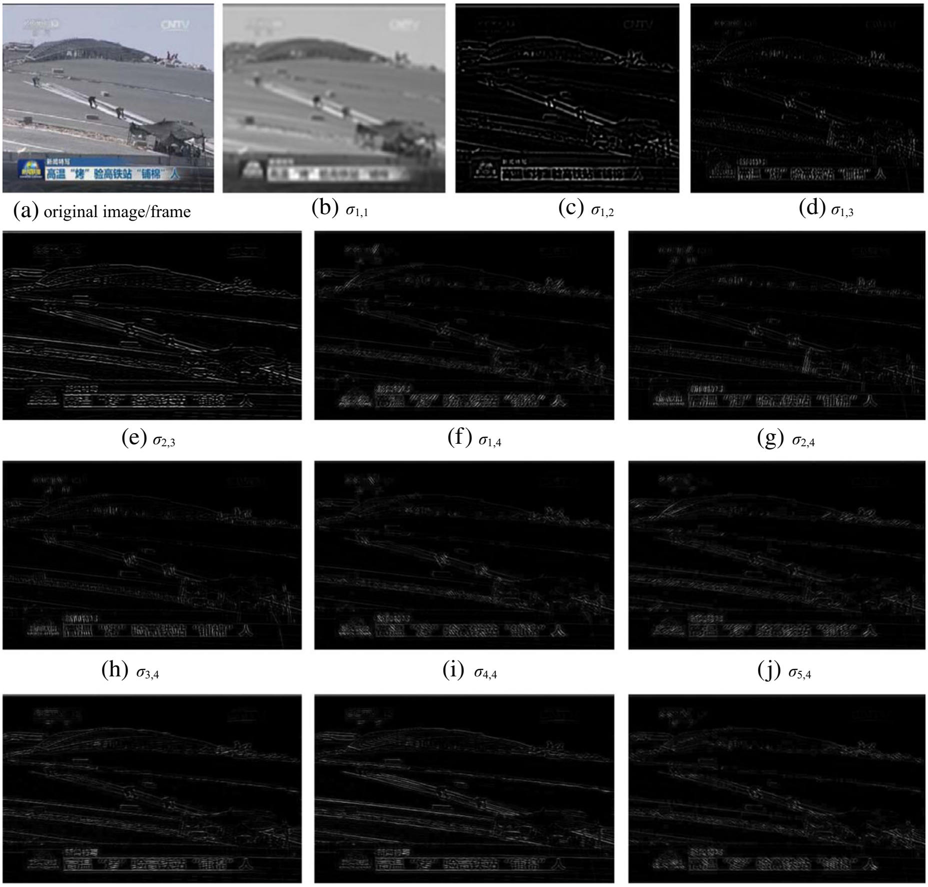
Fig. 2 The framework of the proposed method

According to the discussion of [3], the NSCT can be used for the multiresolution analysis, meanwhile it can also be utilized for the geometrical and directional representation. Therefore, based on NSCT we implement the image multiresolution decomposition and retrieve the NSCT coefficients.

We perform 4 scales of decomposition for NSCT. We retrieve 1,1,2,8 directions in the scales. We use $\sigma_{i,j}$ to denote the variance of the coefficient at the *i*-th directional subband of the *j*-th scale.

According to our experiments which is shown in Fig.3, we find that the NSCT coefficients in level 1 $\sigma_{1,1}$ [Fig.3(b)] is blurred. The NSCT coefficients in level 2 $\sigma_{1,2}$ [Fig.3(c)] preserve the edge features but keep many background noises as well. Meanwhile, the NSCT coefficients in level 3 $\sigma_{1,3}$ [Fig.3(d)] and $\sigma_{2,3}$ [Fig.3(e)] retrieve the text edge, however there are too many backgrounds noises.

Compared with other level coefficients, the NSCT coefficients in level 4 keep the text edge features in some directions completely, which can also suppress the edge features of background noises effectively.



(a) original image/frame (b) $\sigma_{1,1}$ (c) $\sigma_{1,2}$ (d) $\sigma_{1,3}$

(e) $\sigma_{2,3}$ (f) $\sigma_{1,4}$ (g) $\sigma_{2,4}$

(h) $\sigma_{3,4}$ (i) $\sigma_{4,4}$ (j) $\sigma_{5,4}$

(k) $\sigma_{6,4}$ (l) $\sigma_{7,4}$ (m) $\sigma_{8,4}$

Fig. 3 NSCT Decomposition Coefficients

For the 4th scale, we retrieve 8 directional coefficients $\{\sigma^c_{1,4}, \sigma^c_{2,4}, \sigma^c_{3,4}, \sigma^c_{4,4}, \sigma^c_{5,4}, \sigma^c_{6,4}, \sigma^c_{7,4}, \sigma^c_{8,4}\}$ on RGB channels respectively, $c \in \{R,G,B\}$. Then we can compute the $\sigma_{i,j} \in A = \{\sigma_{1,4}, \sigma_{2,4}, \sigma_{3,4}, \sigma_{4,4}, \sigma_{5,4}, \sigma_{6,4}, \sigma_{7,4}, \sigma_{8,4}\}$ as follows:

$$\sigma_{i,4} = \sqrt{\left(\sigma^R_{i,4}\right)^2 + \left(\sigma^G_{i,4}\right)^2 + \left(\sigma^B_{i,4}\right)^2} \tag{1}$$

where $i \in \{1,2,3,4,5,6,7,8\}$. Fig. 3 (f)-(m) show that the coefficients which we retrieve from the 4th scale at the 8 directional subbands.

From the Fig. 3 (f)-(m), we can find that most of the characters edge information can be kept and the background noises will be removed to some degree, which proves that the NSTC multi-resolution decomposition can remove the background noise interference effectively. As a result, it is beneficial to further text detection.

In terms of the text character, its edge features are generally the key clues for text detection. In this paper, we also want to utilize the character edge to find the text in the video. Based on our observation, we found that human beings can recognize the text character due to the strong character edge features in some directions, such as horizontal, vertical and diagonal directions. Therefore, we want to retrieve the edge features in these directions, which is important for us to perform the text detection.

According to the analysis of [3], the NSCT coefficients include multi-directional information. Therefore, we integrate the 8 directional coefficients in different ways, which can retrieve the horizontal, vertical and diagonal edge features.

In this paper, we can define them as directional edge map (DEM). We define $DEM = E_\theta$, $\theta \in \{0^0, 45^0, 90^0, 135^0\}$. Then we can get the *DEM* via formula (2),(3),(4),(5).

$$E_{\theta=0^0} = \sqrt{\left(\sigma^2_{64} + \sigma^2_{74}\right)} \tag{2}$$

$$E_{\theta=45^0} = \sqrt{\left(\sigma^2_{14} + \sigma^2_{54}\right)} \tag{3}$$

$$E_{\theta=90^0} = \sqrt{\left(\sigma^2_{24} + \sigma^2_{34}\right)} \tag{4}$$

$$E_{\theta=135^0} = \sqrt{\left(\sigma^2_{44} + \sigma^2_{84}\right)} \tag{5}$$

$E_{\theta=45^0}$ can represent the 45^0 edge features. $E_{\theta=135^0}$ can display the 135^0 edge features. $E_{\theta=90^0}$ shows the vertical edge features. $E_{\theta=0^0}$ demonstrates the horizontal edge features.

The *DEM* was shown in Fig.4, which displays the different directional edge features and removes some background noises. Compared with the 8 directional coefficients $\sigma_{i,j} \in A = \{\sigma_{1,4}, \sigma_{2,4}, \sigma_{3,4}, \sigma_{4,4}, \sigma_{5,4}, \sigma_{6,4}, \sigma_{7,4}, \sigma_{8,4}\}$ in Fig.3, we conclude that the *DEM* aggregates the character features in 4 directions ($\theta = \{0^0, 45^0, 90^0, 135^0\}$), which can preserve more character edge features.

Furthermore, as to text detection, based on *DEM* we pinpoint the characters candidate pixels in the frames. Then based on these candidate pixels, we can locate the text characters regions easily. Therefore, for the *DEM*, formula (6) was applied to retrieve the binary

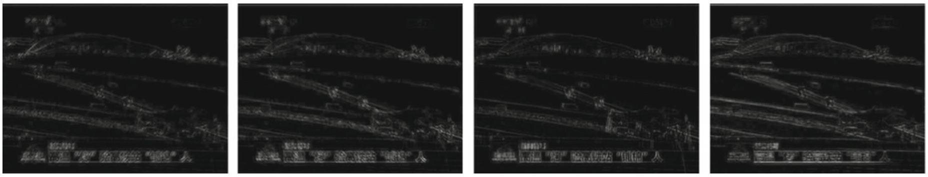

(a) $\theta=45^0$ (b) $\theta=135^0$ (c) $\theta=90^0$ (d) $\theta=0^0$

Fig. 4 directional edge map (*DEM*)

directional edge map (*BDEM*), BDEM $= BE_{\theta}\in\{BE_{\theta=45^0}, BE_{\theta=135^0}, BE_{\theta=90^0}, BE_{\theta=0^0}\}$. The pixels in the *BDEM* can be regarded as the candidate text characters pixels.

$$BE_{\theta}=^{\theta}\begin{cases} 1 \text{ if } E_{\theta} > \alpha T \\ 0 \text{ otherwise} \end{cases} \tag{6}$$

In formula (6), α was defined as the binary factor to retrieve the binary map. T is the mean value of E_{θ}. The refined binary factor algorithm is listed in algorithm 1.

Algorithm 1 Refined binary factor *α*

```
Procedure ReBinaryF(α)
/* a is the initialized binary factor */
/* initialized α value*/
α =12;
/* compute the mean value $m_e$ in DEM*/
  $m_1$=mean($E_{\theta=45^0}$);
  $m_2$=mean($E_{\theta=135^0}$);
  $m_3$=mean($E_{\theta=90^0}$);
  $m_4$=mean($E_{\theta=0^0}$);
  $m_e$=max($m_1$, $m_2$, $m_3$, $m_4$);
/* compute the standard deviation value $S_e$ in DEM*/
  $S_{d1}$=std($E_{\theta=45^0}$);
  $S_{d2}$=std($E_{\theta=135^0}$);
  $S_v$=std($E_{\theta=90^0}$);
  $S_h$=std($E_{\theta=0^0}$);
  $S_e$=max($S_{d1}$, $S_{d2}$, $S_v$, $S_h$);
  $M_d = \sqrt{\frac{1}{4}\sum_{i=1}^{4}(m_i - m_e)^2}$
if $M_d$ >0.01* $S_e$
    α= $M_d$/ $S_e$;
else
    α=α+$M_d$/(0.5*$S_e$);
endif
return α;
```

Fig.5 displays the *BDEM*. The *BDEM* contains some directional text characters pixels. In order to detect the text characters, the various directional pixels are integrated into a whole binary image (*BE*) as formula (7).

$$BE(x,y) = \begin{cases} 1 \text{ if } BE_{\theta=45^0}(x,y)=0 \,\|\, BE_{\theta=135^0}(x,y)=0 \\ \quad \|\, BE_{\theta=90^0}(x,y)=0 \,\|\, BE_{\theta=0^0}(x,y)=0 \\ 0 \quad \text{otherwise} \end{cases} \tag{7}$$

For the *BE*, all white pixels will be regarded as the candidate text characters pixels.

The NSCT could retrieve the geometrical information pixel by pixel from the coefficients, therefore, this edge detection method reveals reliable edge pixels. Moreover, NSCT is suitable for retrieving strong edges, weak edges and noise. The strong edges, weak edges and noise regions are with various pixel intensities, which can be used to eliminate noises of the frame. Therefore, the NSCT edges are quite appropriate to locate text-edges in the video frame.

Fig. 6 shows the results of five different edge detection methods including Sobel, Prewitt, Canny, wavelet and the proposed NSCT method.

The input frame includes video texts and noisy pixels. Our proposed approach shows considerably better results compared to the other methods.

4 Text detection

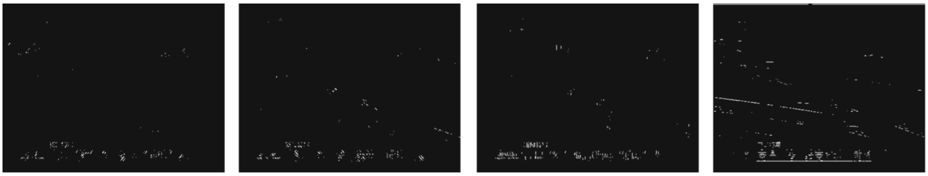
In Section III, this paper elaborated that the NSCT can be utilized to obtain the candidate pixels of text character on *BE*.

As a result, the text regions in the image/frames can be detected via *BE*. According to the definition of [19], text frame classification was performed to determine a video frame as text or non-text before text detection. Therefore, we utilize the *BE* in consecutive frames to discriminate the video text frames from non-text frames.

4.1 Text frame classification

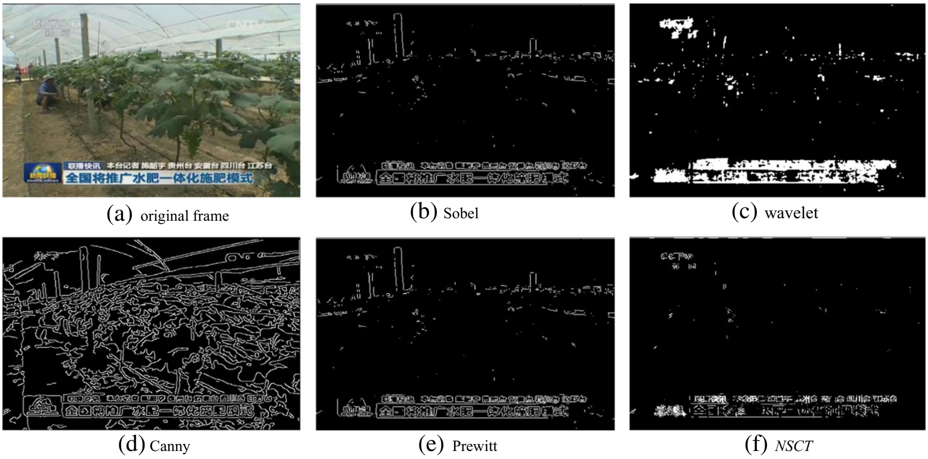
Before text detection, text frame classification should be carried out to determine whether the video frames contained the text lines. It can help to save the computation cost in terms of detecting non-text video frames.

In general, according to the theory of persistence of vision, visual perception of an object does not cease for some time after the rays of light proceeding from it have ceased to enter the eye. As a result, if human beings want to read the video texts clearly, the texts must remain



(a) $\theta=45^0$ (b) $\theta=135^0$ (c) $\theta=90^0$ (d) $\theta=0^0$

Fig. 5 binary directional edge map (*BDEM*)



(a) original frame (b) Sobel (c) wavelet

(d) Canny (e) Prewitt (f) *NSCT*

Fig. 6 Edges by different methods

stable during 1–2 s and keep the same position at least from 30 to 60 consecutive frames. Thus, we select the minimum 30 consecutive frames to perform the same video text detection in our research, which is applicable to any video television signals.

We select the first frame, 15th frame and 30th frame respectively at first. These three frames are processed to retrieve the final *BE*, which can represent the 30 frames text features and remove the most of the background noises.

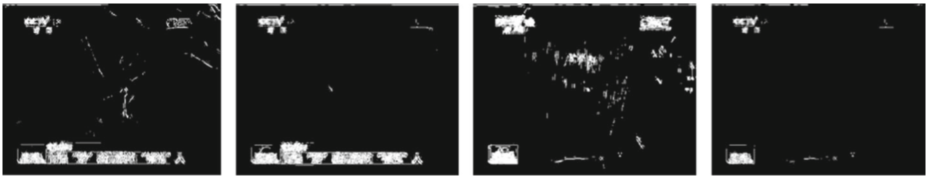
We assume that during the 30 consecutive frames, if the first frame, the 15th frame and the 30th frame all contain the same texts, the *BE* of these frames will include same text characters pixels.

Therefore, we integrate the *BE* of these frames into the F_{BE} as formula (8). F_{BE} represents the same text characters pixels in the 30 consecutive frames.

$$F_{BE}(x,y) = \begin{cases} 1 \text{ if } BE_1(x,y) = 0\&\&BE_{15}(x,y) = 0 \\ \quad \&\&BE_{30}(x,y) = 0 & \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

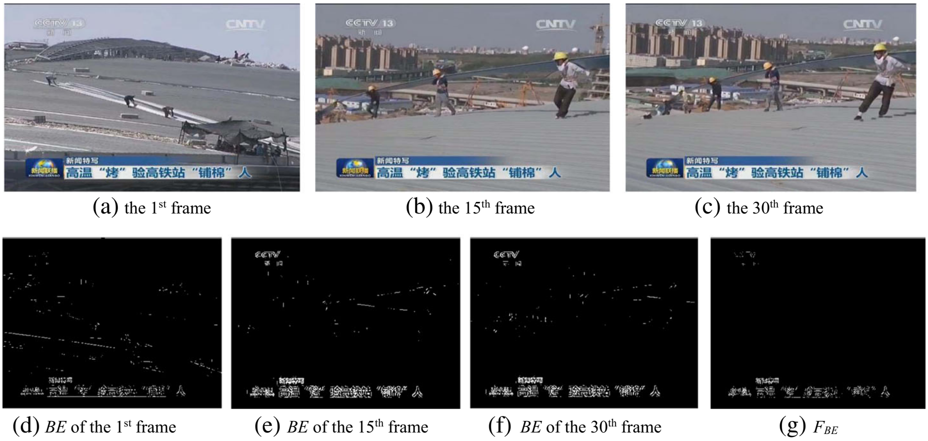
In this paper, F_{BE} is utilized to quickly determine whether the 30 consecutive frames include the same video texts.

Based on our observation, during the 30 consecutive frames if these consecutive frames do not include same video texts, they have relatively few same pixels. These consecutive frames without same video texts generally include same channel logo and a few other pixels as well, as is shown in Fig.7(d).



(a) *BE* of the 1st frame (b) *BE* of the 15th frame (c) *BE* of the 30th frame (d) F_{BE}

Fig. 7 *BE* does not include same texts during 30 consecutive frames



(a) the 1st frame (b) the 15th frame (c) the 30th frame

(d) *BE* of the 1st frame (e) *BE* of the 15th frame (f) *BE* of the 30th frame (g) F_{BE}

Fig. 8 binary image (*BE*) of the 1st, 15th and 30th frames in the consecutive video sequences

As a result, if Num(F_{BE}) < τ, we can regard that there are no same video texts during the 30 consecutive frames. The Num(F_{BE}) is the number of F_{BE} pixels. As the logo only covers a tiny fraction of the whole video frame and the video frame resolution is 640 × 480, we define $\tau = (W*H)/10{,}000$. *W* and *H* represent the video frame width and height, respectively.

As is shown in Fig.7(d), the F_{BE} kept only few candidate text characters pixels besides the logo information of news video.

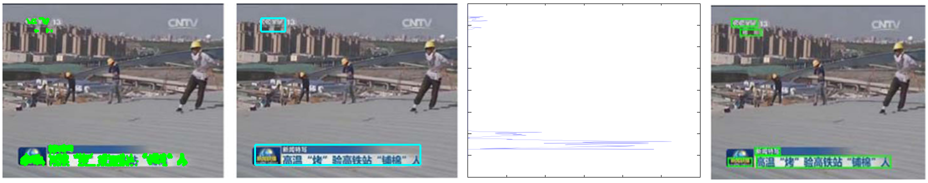
Therefore, the 30 consecutive frames do not contain the same video texts. In the following text detection, we can remove these video frames, which can significantly increase the text detection speed and accuracy.

4.2 Text detection

After the text frame classification, we can get the 30 consecutive frames which include the same video texts. For the 30 consecutive frames, we can still retrieve the *BE* of the first frame, the 15th frame and the 30th frame, which is shown in Fig.8(d)(e)(f), respectively.

For the single frame *BE* which is shown in the Fig.8(d) (e) (f), we find that although the *BE* contains text characters pixels, it also includes some background noises, which will definitely pose some difficulties for further text detection. The F_{BE} integrates the *BE* of the first frame, the 15th frame and the 30th frame, which is shown in Fig.8 (g). We can find that the F_{BE} preserves the most of the text characters pixels and removes the background noises efficiently.

Therefore, we can use F_{BE} to locate the text regions. However, there are still some background noises in the F_{BE}. So, we filter every candidate text pixels to pinpoint the key



(a) key text pixels (b) text regions (c) text lines (d) final detected text lines

Fig. 9 text detection in the *FBE*

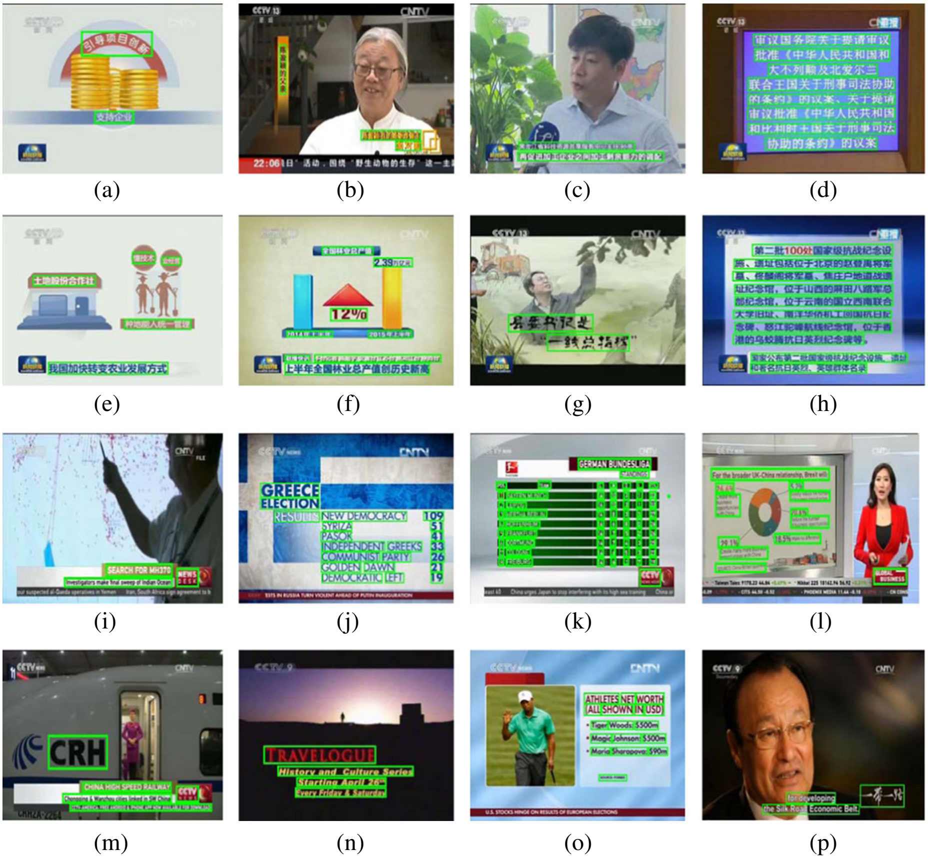
text pixels. We find that in the neighborhood of the key text pixels, there still exist some other text pixels. Otherwise, the isolated candidate pixels may be determined as the background noises.

We use the sliding windows with height and width as 20 and 20 pixels to process the F_{BE}. For the pixel in the middle of the sliding window, if the candidate pixels number of the sliding windows is above 10, we can conclude that the pixel in the middle of the sliding window is key text pixels. The green pixels of Fig.9(a) display the key text pixels. Then we use the key text pixels to locate the text regions, which is shown in Fig.9(b). Based on the text pixels projection in the text regions, we can segment the text regions into final text lines. The text pixels projection is shown in Fig.9(c). The final detected text lines are shown in Fig.9(d).

5 Experiments and discussion

The data set used in this work comprises two sets of TV video from two TV channels (CCTV-1 and CCTV-News). The two data sets are captured from the broadcast news video.

China Central Television, or CCTV, is a national television station of the People's Republic of China. CCTV-1 is available in Mandarin and CCTV-News is available in English. Each set



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

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

Fig. 10 Text Detection Experiments Results

is a collection of news report in Chinese or English. CCTV is constantly updated with top news from China and around the world, offering news reports, live and on-demand video content. The video content ranges from culture, sports, economy to politics. As a result, the texture of the videos in the data set is rather complicated and video text detection in this dataset is a kind of challenging task.

We used 210-min news video to evaluate the proposed method, FPS (Frames Per Second) is 25 frames per second. As a result, a total of 351,000 video frames are performed to evaluate the method. We normalize the news video resolution as 640 × 480.

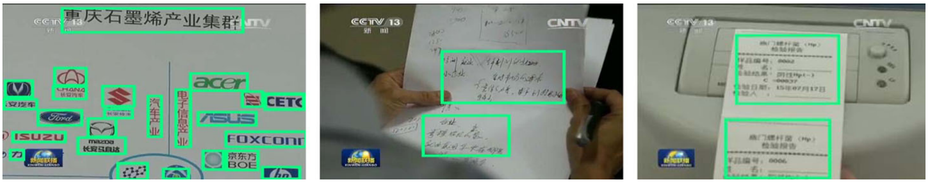
The experimental results of video text detection are shown in Fig.10. The first and second row of Fig.10 [(a)-(h)] are the detection results of Chinese texts. The third and fourth row of Fig.10 [(i)-(p)] are the detection results of English texts. The missed text lines at the bottom in Fig.10 are the scrolling texts. Our method cannot detect the scrolling text. Curved alignment text is successfully detected in Fig.10(a). Fig.10(b) demonstrates that the proposed approach is robust to detect horizontal/vertical text. Based on the detected results of Fig.10(a) and Fig.10(b), we can find that our method can detect not only horizontal/vertical alignment text but also curved alignment text.

Fig.10(c) shows that our method can discriminate the text accurately from the backgrounds which own similar texture regions, such as leaves and shirt stripes. Fig.10(d) shows the proposed method is robust to detect the text line with big characters. The text rows which have complex layout [Fig.10(e) (f)] have been accurately detected, which shows that the complex layout can be accurately detected. Fig.10(f) shows that the proposed approach can detect the text with complex layout and different colors. Fig.10(g) demonstrates that our method can distinguish the calligraphy text from the background. The multi text lines can also be detected accurately in Fig.10(h).

In Fig.10(i), the background displays many dots and points, which in general will interfere the text detection accuracy due to similar textures. Fig. 10(i) demonstrates that the English texts embedded in complicated background are correctly detected. English texts with complex layout can be located by our approach in Fig.10(j)(k)(l). Fig.10(m)(n)(o) show that our approach is robust to various font-sizes and font-colors. In Fig. 10(p), both Chinese characters and English characters are successfully detected.

These experimental results in Fig.10 demonstrate that our method can detect multilingual (Chinese/English) horizontal/vertical/curved texts with complex layout, multi-colors and multi-sizes.

Fig.11 displays some experimental results with misses or false detections. Fig.11 (a) shows that some trademarks were wrongly classified as text. It can be seen that our method cannot detect the handwritten texts in Fig.11 (b) because the handwritten texts show low contrast and handwritten characters are so blurred that even human eyes can hardly recognize them in video. In Fig.11(c), the small characters cannot be detected



(a) (b) (c)

Fig. 11 Examples with Missed and False Detections

Table 1 Video Text Detection Performance Comparison

	Total textboxes	Total missed textboxes	Total false alarm	Detection rate	False alarm rate	Detection speed (second/frame)
Lyu's Method	217,963	17,219	20,706	92.1%	9.49%	0.26
Palaiahnakote's Method	217,963	14,821	20,074	93.2%	9.2%	0.24
Mosleh's Method	217,963	16,129	19,834	92.6%	9.1%	0.25
Kim's method	217,963	16,784	19,399	92.3%	8.9%	0.23
Our Method	217,963	13,949	18,810	93.6%	8.63%	0.22

correctly. Because the small characters edge features are weak, our method cannot locate the boundaries of the text lines. Therefore, conclusion can be drawn that the proposed approach will bring about some misses when the characters are too small or handwritten, and cause some false alarms in some symbols that show similar edge features with the text characters.

We conduct the comparison of our method with Lyu's approach [9], Palaiahnakote's approach [19], Mosleh's approach [11] and Kim's approach [7].

The reason why we choose them is that Lyu's approach is a classical solution which owns the multilingual processing capability, while Palaiahnakote's method is a frequency-based method, which detects text in video frames of unconstrained background, different fonts, different scripts, and different font sizes. The Mosleh's approach is a kind of stroke-based text detection method, which has been widely utilized in video text detection in recent years. The text locations in each frame are found via an unsupervised clustering performed on the connected components produced by the stroke width transform (SWT). Next, the motion patterns of the text objects of each frame are analyzed to localize video texts. Kim et al. [7] propose a novel method to detect the overlay text based on the transition map, which is introduced based on logarithmical change of intensity and modified saturation. Overlay text region update between frames is used to reduce the processing time. Table 1 shows the comparison results.

The results of comparison display that our method acquires the better detection rate and precision than that of Lyu's method, Palaiahnakote's approach, Mosleh's approach and Kim's approach. Our method is the fastest among the five methods. The detection speed indicates the average processing time per frame of our method evaluated on a Personal Computer with Intel(R) Core(TM)2 Quad CPU Q4900 2.66GHz. The comparison results in Table 1 indicates that the proposed approach can achieve comparatively higher detection rate and lower false alarm rate. The original truth data are calculated manually.

Table 2 Video Text Detection Performance Comparison on Long Text Lines

	Total textboxes	Total missed textboxes	Total false alarm	Detection rate	False alarm rate	Detection speed (second/frame)
Lyu's Method	154,876	11,474	10,615	92.6%	6.85%	0.26
Palaiahnakote's Method	154,876	9885	10,221	93.6%	6.6%	0.24
Mosleh's Method	154,876	10,531	10,376	93.2%	6.7%	0.25
Kim's method	154,876	10,687	10,082	93.1%	6.51%	0.23
Our Method	154,876	9137	9524	94.1%	6.14%	0.22

Table 3 Video Text Detection Performance Comparison on Short Text Lines

	Total textboxes	Total missed textboxes	Total false alarm	Detection rate	False alarm rate	Detection speed (second/frame)
Lyu's Method	63,087	5745	10,091	90.9%	15.9%	0.26
Palaiahnakote's Method	63,087	4936	9853	92.1%	15.6%	0.24
Mosleh's Method	63,087	5598	9458	91.1%	14.99%	0.25
Kim's method	63,087	6097	9317	90.3%	14.8%	0.23
Our Method	63,087	4812	9286	92.3%	14.7%	0.21

Text lines on the video frames generally have different number of characters. In general, the characters in the short text lines are so few that they cannot produce enough apparent profiles regions. Some methods have difficulty in detecting the short text lines with few text characters.

On the contrary, the long text lines with more text characters will display similar textures with the background so it is also not easy to detect the text.

For the text lines, we define the text lines which have more than 5 characters as long text line, on the contrary, the text lines which have 5 characters or fewer is short text line.

For a quantitative evaluation of detecting the texts with long text line and short text line respectively, we divide the comparison results of Table 1 into two Sub-Tables, as shown in Tables 2–3.

Tables 2 and 3 show the text detection comparison results with multi text lines. As can be seen in Tables 2 and 3, our method achieves the highest detection rate and accuracy for both types of text lines. The comparison results show that our method is more suitable for the long text line and short text line detection than the other four methods.

For a quantitative evaluation of detecting the texts with different languages (English and Chinese), we divide the comparison results of Table 1 into two Sub-Tables, as shown in Tables 4-5.

As can be seen in Tables 4 and 5, our method achieves the highest detection rate and accuracy for both English and Chinese text boxes. The comparison results show that our method is more suitable for multiple languages detection than the other four methods.

6 Conclusion

In this paper, the author proposes a new text detection approach based on NSCT. Our approach is robust to detect multilingual horizontal/vertical texts. Due to the fully shift-invariant, multi-scale, and multi-direction features of NSCT, the author integrates the directional coefficients of

Table 4 Performance comparison for text detection with English

	Total textboxes	Total missed textboxes	Total false alarm	Detection rate	False alarm rate	Detection speed (second/frame)
Lyu's Method	103,691	7154	7206	93.1%	6.95%	0.26
Palaiahnakote's Method	103,691	6739	6978	93.5%	6.73%	0.24
Mosleh's Method	103,691	6636	6646	93.6%	6.41%	0.25
Kim's method	103,691	7757	6715	92.5%	6.47%	0.23
Our Method	103,691	6428	6439	93.8%	6.21%	0.22

Table 5 Performance comparison for text detection with Chinese

	Total textboxes	Total missed textboxes	Total false alarm	Detection rate	False alarm rate	Detection speed (second/frame)
Lyu's Method	114,272	10,065	13,500	91.19%	11.81%	0.26
Palaiahnakote's Method	114,272	8082	13,096	92.92%	11.46%	0.24
Mosleh's Method	114,272	9493	13,188	91.69%	11.54%	0.25
Kim's method	114,272	9027	12,684	92.1%	11.1%	0.23
Our Method	114,272	7521	12,371	93.41%	10.82%	0.21

NSCT into directional edge map (DEM). Based on the DEM, text frame classification and text detection are carried out. We also report the experimental results and the comparison results in detail, which shows that our approach can efficiently detect video text with multilingual horizontal/vertical/curved texts with complex layout and multi-color.

Acknowledgements This work reported in this paper is supported by Beijing Natural Science Foundation(4173073); the Surface Project of Beijing Committee of Education under Grant No. KM201710028021.

References

1. Barinova O, Lempitsky V, Kholi P (2012) On detection of multiple object instances using Hough transforms. IEEE Trans Pattern Anal Mach Intell 34(9):1773–1784
2. Chen Z, You X, Zhong B, Li J, Tao D (2016) Dynamically modulated mask sparse tracking. IEEE Trans Cybern. doi:10.1109/TCYB.2016.2577718
3. Cunha AL, Zhou J, Do MN (2006) Nonsubsampled contourlet transform: theory, design, and applications. IEEE Trans Image Process 15(10):3089–3101
4. Huang X, Ma H, Lin CX, Gao G (2014) Detecting both superimposed and scene text with multiple languages and multiple alignments in video. Multimedia Tools and Applications 70(3):1703–1727
5. Jung C, Liu Q, Kim J (2009) Accurate text localization in images based on SVM output scores. Image Vis Comput 27(9):1295–1301
6. Jung C, Liu Q, Kim J (2009) A stroke filter and its application to text localization. Pattern Recogn Lett 30(2):114–122
7. 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
8. Lu S, Barner KE, (2008) Weighted DCT Coefficient based text detection, in Proceeding of 2008 I.E. International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp: 1341–1344
9. Lyu MR, Song J, Cai M (2005) A comprehensive method for multilingual video text detection, localization, and extraction. IEEE Trans Circuits Syst Video Technol 15(2):243–255
10. Moradi M, Mozaffari S (2013) Hybrid approach for Farsi/Arabic text detection and localisation in video frames. IET Image Process 7(2):154–164
11. Mosleh A, Bouguila N (2013) Abdessamad ben Hamza, automatic Inpainting scheme for video text detection and removal. IEEE Trans Image Process 22(11):4460–4472

12. Pham D-S (2015) Ognjen Arandjelovi'c, Svetha Venkatesh, detection of dynamic background due to swaying movements from motion features. IEEE Trans Image Process 24(1):332–344
13. Phan TQ, Shivakumara P, Tan CL, (2009) A Laplacian Method for Video Text Detection", In Proceeding of 2009 International Conference on Document Analysis and Recognition (ICDAR), pp: 66–70
14. Phan TQ, Shivakumara P, Tan CL, (2009) A Gradient Difference based Technique for Video Text Detection", in Proceeding of 2009 International Conference on Document Analysis and Recognition (ICDAR), pp: 156–160
15. Raza A, Siddiqi I, Djeddi C, Ennaji A (2013) Multilingual artificial text detection Using a Cascade of Transforms", In Proceeding of 2013 12th International Conference on Document Analysis and Recognition (ICDAR), pp: 309–313
16. Shivakumara P, Huang W (2010) Trung Quy Phan, Chew Lim tan, accurate video text detection through classification of low and high contrast images. Pattern Recogn 43(6):2165–2185
17. Shivakumara P, Phan TQ, Tan CL (2009) A Robust Wavelet Transform Based Technique for Video Text Detection", In Proceeding of 2009 International Conference on Document Analysis and Recognition (ICDAR), pp: 1285–1289
18. Shivakumara P, Phan TQ, Tan CL, (2009) Video Text Detection Based on Filters and Edge Features", In Proceeding of 2009 I.E. International Conference on Multimedia and Expo (ICME), pp:514–517
19. Shivakumara P, Phan TQ, Tan CL (2010) New Fourier-statistical features in RGB space for video text detection. IEEE Trans Circuits Syst Video Technol 20(11):1520–1532
20. Sun L, Liu G, Qian X, Guo D, (2009) A Novel Text Detection and Localization Metheod Based on Corner Response", In Proceeding of 2009 I.E. International Conference on Multimedia and Expo (ICME), pp: 390–393
21. Zhang J, Kasturi R (2010) Text Detection Using Edge Gradient and Graph Spectrum", In Proceeding of 2010 International Conference on Pattern Recognition (ICPR), pp: 3979–3982

Xiaodong Huang is an associate professor of capital normal university, China. He received his Ph.D. degree in Computer Science from the Beijing University of Posts and Telecommunications in 2010, M.S. degree in computer science from the Beijing University of Posts and Telecommunications in 2006 and B.S. degree in computer science from Wuhan University of Technology in 1995. His research interests include pattern recognition and computer vision.